

Direct-a-Video: Customized Video Generation with User-Directed Camera Movement and Object Motion

Shiyuan Yang

City University of Hong Kong¹
Tianjin University²
¹Hong Kong, ²Tianjin, China
s.y.yang@my.cityu.edu.hk

Liang Hou

Kuaishou Technology
Beijing, China
houliang06@kuaishou.com

Haibin Huang

Kuaishou Technology
Beijing, China
jackiehuanghaibin@gmail.com

Chongyang Ma

Kuaishou Technology
Beijing, China
chongyangm@gmail.com

Pengfei Wan

Kuaishou Technology
Beijing, China
wanpengfei@kuaishou.com

Di Zhang

Kuaishou Technology
Beijing, China
zhangdi08@kuaishou.com

Xiaodong Chen

Tianjin University
Tianjin, China
xdchen@tju.edu.cn

Jing Liao*

City University of Hong Kong
Hong Kong, China
jingliao@cityu.edu.hk

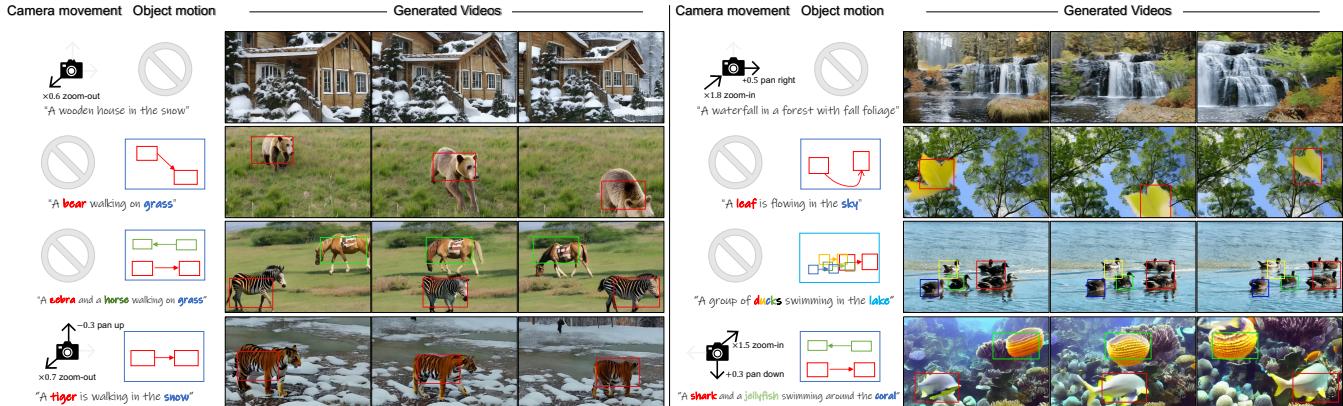


Figure 1: Direct-a-Video is a text-to-video generation framework that allows users to individually or jointly control the camera movement and/or object motion.

ABSTRACT

Recent text-to-video diffusion models have achieved impressive progress. In practice, users often desire the ability to control object motion and camera movement independently for customized video creation. However, current methods lack the focus on separately controlling object motion and camera movement in a decoupled manner, which limits the controllability and flexibility of text-to-video models. In this paper, we introduce Direct-a-Video, a system that allows users to independently specify motions for multiple objects as well as camera's pan and zoom movements, as if directing a video. We propose a simple yet effective strategy for the decoupled control of object motion and camera movement. Object motion is controlled through *spatial cross-attention* modulation using the model's inherent priors, requiring no additional optimization. For camera movement, we introduce new *temporal cross-attention* layers

to interpret quantitative camera movement parameters. We further employ an augmentation-based approach to train these layers in a self-supervised manner on a small-scale dataset, eliminating the need for explicit motion annotation. Both components operate independently, allowing individual or combined control, and can generalize to open-domain scenarios. Extensive experiments demonstrate the superiority and effectiveness of our method. Project page and code are available at <https://direct-a-video.github.io/>.

CCS CONCEPTS

- Computing methodologies → Motion processing.

KEYWORDS

Text-to-video generation, motion control, diffusion model.

*Corresponding author.

1 INTRODUCTION

Text-to-image (T2I) diffusion models have already demonstrated astonishingly high quality and diversity in image generation and editing [Ho et al. 2020; Ramesh et al. 2022; Rombach et al. 2022; Saharia et al. 2022; Wang et al. 2023a; Yang et al. 2020, 2023a; Yuan et al. 2024]. The rapid development of T2I diffusion models has also spurred the recent emergence of text-to-video (T2V) diffusion models [Blattmann et al. 2023a,b; Ho et al. 2022a; Singer et al. 2022; Wang et al. 2023d], which are normally extended from pre-trained T2I models for video generation and editing. On the other hand, the advent of controllable techniques in T2I models, such as ControlNet [Zhang et al. 2023], T2I-adapter [Mou et al. 2023] and GLIGEN [Li et al. 2023], has allowed users to specify the spatial layout of generated images through conditions like sketch maps, depth maps, or bounding boxes etc., significantly enhancing the spatial controllability of T2I models. Such spatial controllable techniques have also been successfully extended to spatial-temporal control for video generation. One of the representative works in this area is VideoComposer [Wang et al. 2023f], which can synthesize a video given a sequence of sketch or motion vector maps.

Despite the success of video synthesis, current T2V methods often lack support for user-defined and disentangled control over camera movement and object motion, which limits the flexibility in video motion control. In a video, both objects and the camera exhibit their respective motions. Object motion originates from the subject’s activity, while camera movement influences the transition between frames. The overall video motion becomes well-defined only when both camera movement and object motion are determined. For example, focusing solely on object motion, such as generating a video clip where an object moves to the right within the frame, can lead to multiple scenarios. The camera may remain stationary while the object itself moves right, or the object may be stationary while the camera moves left, or both the object and the camera may be moving at different speeds. This ambiguity in the overall video motion can arise. Therefore, the decoupling and independent control of camera movement and object motion not only provide more flexibility but also reduce ambiguity in the video generation process. However, this aspect has received limited research attention thus far.

To control camera movement and object motion in T2V generation, a straightforward approach would be to follow the supervised training route similar to works like VideoComposer [Wang et al. 2023f]. Following such kind of scheme involves training a conditional T2V model using videos annotated with both camera and object motion information. However, this would bring the following challenges: (1) In many video clips, object motion is often coupled with camera movements due to their inherent correlation. For example, when a foreground object moves to some direction, the camera typically pans in the same direction due to the preference to keep the main subject at the center of the frame. Training on such coupled camera and object motion data makes it difficult for the model to distinguish between camera movements and object motion. (2) Obtaining large-scale video datasets with complete camera movement and object motion annotations is challenging due to the laborious and costly nature of performing frame-by-frame

object tracking and camera pose estimation. Additionally, training a video model on a large-scale dataset can be computationally expensive.

In this work, we introduce Direct-a-Video, a text-to-video framework that enables users to independently specify the camera’s pan and zoom movements and the motions of scene objects, allowing them to create their desired motion pattern as if they were directing a video (Figure 1). To achieve this, we propose a strategy for decoupling camera and object control by employing two orthogonal controlling mechanisms. In essence, we learn the camera movement through a self-supervised and lightweight training approach. Conversely, during inference, we adopt a training-free method to control object motion. Our strategy avoids the need for intensive collection of motion annotations and video grounding datasets.

In camera movement control, we train an additional module to learn the frame transitions. Specifically, we introduce new temporal cross-attention layers, known as the camera module, which functions similarly to spatial cross-attention in interpreting textual language. This camera module interprets “camera language”, specifically camera panning and zooming parameters, enabling precise control over camera movement. However, acquiring datasets with camera movement annotations can pose a challenge. To overcome this laborious task, we employ a self-supervised training strategy that relies on camera movement augmentation. This approach eliminates the need for explicit motion annotations. Importantly, we train these new layers while preserving the original model weights, ensuring that the extensive prior knowledge embedded within the T2V model remains intact. Although the model is initially trained on a small-scale video dataset, it acquires the capability to quantitatively control camera movement in diverse, open-domain scenarios.

In object motion control, a significant challenge arises from the availability of well-annotated grounding datasets for videos, curating such datasets is often a labor-intensive process. To bypass these issues, we draw inspiration from previous attention-based image-layout control techniques in T2I models [Hertz et al. 2022; Kim et al. 2023]. We utilize the internal priors of the T2V model through spatial cross-attention modulation, which is a training-free approach, thereby eliminating the need for collecting grounding datasets and annotations for object motion. To facilitate user interaction, we enable users to specify the spatial-temporal trajectories of objects by drawing bounding boxes at the first and last frames, as well as the intermediate path. Such interaction is simpler and more user-friendly compared to previous pixel-wise control methods [Wang et al. 2023f].

Given that our approach independently controls camera movement and object motion, thereby effectively decouples the two, offering users enhanced flexibility to individually or simultaneously manipulate these aspects in video creation.

In summary, our contributions are as follows:

- We propose a unified framework for controllable video generation that decouples camera movement and object motion, allowing users to independently or jointly control both aspects.
- For camera movement, we introduce a novel temporal cross-attention module dedicated to camera movement conditioning. This camera module is trained through self-supervision, enabling

- users to quantitatively specify the camera’s horizontal and vertical panning speeds, as well as its zooming ratio.
- For object motion, we utilize a training-free spatial cross-attention modulation, enabling users to easily define the motion trajectories for one or more objects by drawing bounding boxes.

2 RELATED WORK

2.1 Text-to-Video Generation

The success of text-to-image (T2I) models has revealed their potential for text-to-video (T2V) generation. T2V models are often evolved from T2I models by incorporating temporal layers. Early T2V models [Ho et al. 2022a,b; Singer et al. 2022] perform the diffusion process in pixel space, which requires multiple cascaded models to generate high-resolution or longer videos, resulting in high computational complexity. Recent T2V models draw inspiration from latent diffusion [Rombach et al. 2022] and operate in a lower-dimensional and more compact latent space [Blattmann et al. 2023b; Esser et al. 2023; Guo et al. 2023; Wang et al. 2023d; Zhou et al. 2022]. The most recent Stable Video Diffusion [Blattmann et al. 2023a] utilizes curated training data and is capable of generating high-quality videos.

On the other hand, the development of T2I editing techniques [Gal et al. 2022; Hertz et al. 2022; Kumari et al. 2023; Mokady et al. 2023; Ruiz et al. 2023] has facilitated zero/few-shot video editing tasks. These techniques convert a given source video to a target video through approaches such as weight fine-tuning [Wu et al. 2023b], dense map conditioning [Esser et al. 2023; Geyer et al. 2023; Yang et al. 2023b; Zhao et al. 2023b], sparse point conditioning [Gu et al. 2023; Tang et al. 2023], attention feature editing [Ceylan et al. 2023; Liu et al. 2023b; Qi et al. 2023; Wang et al. 2023c], and canonical space processing [Chai et al. 2023; Kasten et al. 2021; Ouyang et al. 2023]. Some works specifically focus on synthesizing human dance videos using source skeleton sequences and reference portraits [Chang et al. 2023; Feng et al. 2023; Hu et al. 2023; Wang et al. 2023b; Xu et al. 2023], which have yielded impressive results.

2.2 Video Generation with Controllable Motion

As motion is an important factor in video, research on video generation with motion control has garnered increasing attention. We can categorize the works in this field into three groups based on the type of input media: image-to-video, video-to-video, and text-to-video.

Image-to-video. Some methods focus on transforming static images into videos, and a popular approach for motion control is through key point dragging [Chen et al. 2023a; Deng et al. 2023; Yin et al. 2023]. While this interaction method is intuitive and user-friendly, it has limitations due to the local and sparse nature of the key points. Consequently, its capacity for controlling motion at a large granularity is significantly restricted.

Video-to-video. These works primarily focus on motion transfer, which involves learning a specific subject action from source videos and applying it to target videos using various techniques, including fine-tuning the model on a set of reference videos with similar motion patterns [Jeong et al. 2023; Wei et al. 2023; Wu et al. 2023a; Zhao et al. 2023a], or borrowing spatial features (e.g., sketch, depth maps) [Chen et al. 2023b; Wang et al. 2023f] or sparse features (e.g.,

DIFT point embedding) [Gu et al. 2023] from source videos. These methods highly rely on the motion priors from the source videos, which, however, are not always practically available.

Text-to-video. In the case where the source video is unavailable, generating videos from text with controllable motion is a meaningful but relatively less explored task. Our work focuses on this category. Existing approaches in this category include AnimateDiff [Guo et al. 2023], which utilizes ad-hoc motion LoRA modules [Hu et al. 2021] to enable specific camera movements. However, it lacks quantitative camera control and also does not support object motion control. VideoComposer [Wang et al. 2023f] provides global motion guidance by conditioning on pixel-wise motion vectors. However, the dense control manner offered by VideoComposer is inefficient to use and does not explicitly separate camera and object motion, resulting in inconvenient user interaction. A concurrent work, Peekaboo [Jain et al. 2023], also uses bounding boxes to control the trajectory of the object through attention masking. However, their method originally does not consider multi-object scenarios and also does not support control over camera movement, unlike our approach. MotionCtrl [Wang et al. 2023e], another concurrent work, allows for separate 2D point-driven object motion control and 3D trajectory-driven camera control by training camera and object control modules. However, its training preparation is labor-intensive, requiring the extraction of motion trajectories on large-scale video dataset. Moreover, it struggles to control multiple different objects with varied motion directions, as it lacks the explicit binding between objects and their motion trajectories during the training. In contrast, our self-supervised training scheme does not require any motion annotations and can achieve motion control over camera and multiple objects, bringing more flexibility for video synthesis.

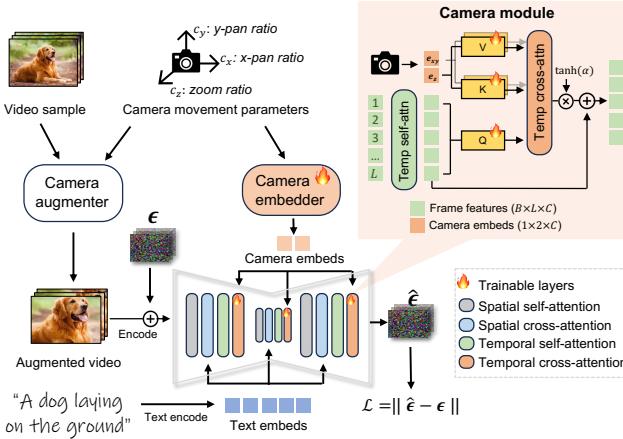
3 METHOD

3.1 Overview

Task formulation. In this paper, we focus on text-to-video generation with user-directed camera movement and/or object motion. First of all, user should provide a text prompt which may optionally contain one or more object words O_1, O_2, \dots, O_N . To determine the camera movement, user can specify an x-pan ratio c_x , a y-pan ratio c_y , and a zoom ratio c_z . To determine the motion of n -th object O_n , user needs to specify a starting box B_n^1 , an ending box B_n^L (L is the video length), and an intermediate track ζ_n connecting B_n^1 and B_n^L , our system then generates a sequence of boxes $[B_n^1, \dots, B_n^L]$ centered along the track ζ_n via interpolation to define the spatial-temporal journey of the object. Consequently, our model synthesizes a video that adheres to the prescribed camera movement and/or object motion, creating customized and dynamic visual narrative.

Overall pipeline. Our overall pipeline is illustrated in Figure 2. The camera movement is learned in the training stage and the object motion is implemented in the inference stage. During the training, we use video samples captured by a stationary camera, which are then augmented to simulate camera movement according to $[c_x, c_y, c_z]$. The augmented videos are subsequently used as input to the U-Net. Additionally, the camera parameters are also encoded and injected into a newly introduced trainable temporal

Training



Inference

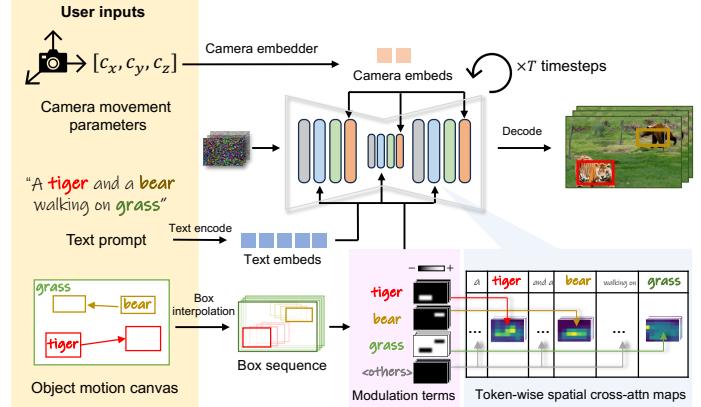


Figure 2: The overall pipeline of Direct-a-Video. The camera movement is learned in the training stage and the object motion is implemented in the inference stage. Left: During training, we apply augmentation to video samples to simulate camera movement using panning and zooming parameters. These parameters are embedded and injected into newly introduced temporal cross-attention layers as the camera movement conditioning, eliminating the need for camera movement annotation. Right: During inference, along with camera movement, user inputs a text prompt containing object words and associated box trajectories. We use spatial cross-attention modulation to guide the spatial-temporal placement of objects, all without additional optimization. Note that our approach, by independently controlling camera movement and object motion, effectively decouples the two, thereby enabling both individual and joint control.

cross-attention layer to condition the camera movement (detailed in Section 3.2). During the inference, with trained camera embedder and module, users can specify the camera parameters to control its movement. Concurrently, we incorporate the object motion control in a training-free manner: given the object words from the user’s prompt and the corresponding boxes, we modulate the frame-wise and object-wise spatial cross-attention maps to redirect the object spatial-temporal size and location (detailed in Section 3.3). It is noteworthy that the modulation in inference stage does not involve additional optimization, thus the incremental time and memory cost is negligible.

3.2 Camera Movement Control

We choose three types of camera movement: horizontal pan, vertical pan, and zoom, parameterized as a triplet $\mathbf{c}_{\text{cam}} = [c_x, c_y, c_z]$ to serve as the camera control signal. This allows for quantitative control, a feature not available in previous work [Guo et al. 2023], and is simple to use and sufficiently expressive for our needs.

Data construction and augmentation. Extracting camera movement information from existing video can be computationally expensive since the object motion needs to be identified and filtered out. As such, we propose a self-supervised training approach using camera augmentation driven by \mathbf{c}_{cam} , thereby bypassing the need for intensive movement annotation.

We first formally define the camera movement parameters. c_x represents the x-pan ratio, and is defined as the total x-shift of the frame center from the first to the last frame relative to the frame width, $c_x > 0$ for panning rightward (e.g., $c_x = 0.5$ for a half-width

right shift). Similarly, c_y is the y-pan ratio, representing the total y-shift of the frame center over the frame height, $c_y > 0$ for panning downward. c_z denotes the zoom ratio, defined as the scaling ratio of the last frame relative to the first frame, $c_z > 1$ for zooming-in. We set the range of c_x, c_y to $[-1, 1]$ and c_z to $[0.5, 2]$, which are generally sufficient for covering regular camera movement range.

In practice, for given \mathbf{c}_{cam} , we simulate camera movement by applying shifting and scaling to the cropping window on videos captured with a stationary camera. This data augmentation exploits readily available datasets like MovieShot [Rao et al. 2020]. Further details of this process, including pseudo code and sampling scheme of \mathbf{c}_{cam} are provided in the supplemental material.

Camera embedding. To encode \mathbf{c}_{cam} into a camera embedding, we use a camera embedder that includes a Fourier embedder, which is widely used for encoding coordinate-like data [Mildenhall et al. 2021], and two MLPs. One MLP jointly encodes the panning movement c_x, c_y , while the other encodes the zooming movement c_z . We empirically found that separately encoding panning and zooming helps the model distinguish between these two distinct types of camera movements effectively, and we validate this design in Section 4.5. The embedding process can be formulated as $\mathbf{e}_{xy} = \text{MLP}_{xy}(\mathcal{F}([\mathbf{c}_x, \mathbf{c}_y]))$, $\mathbf{e}_z = \text{MLP}_z(\mathcal{F}(c_z))$, where \mathcal{F} denotes Fourier embedder. Both \mathbf{e}_{xy} and \mathbf{e}_z have the same feature dimensions. By concatenating them, we obtain the camera embedding $\mathbf{e}_{\text{cam}} = [\mathbf{e}_{xy}, \mathbf{e}_z]$, which has a sequence length of two.

Camera module. We now consider where to inject the camera embedding. Previous studies have highlighted the role of temporal layers in managing temporal transitions [Guo et al. 2023; Zhao

et al. 2023a]. As such, we inject camera control signals via temporal layers. Inspired by the way spatial cross-attention interprets textual information, we introduce new trainable temporal cross-attention layers specifically for interpreting camera information, dubbed as camera modules, which are appended after the existing temporal self-attention layers within each U-Net block of the T2V model, as depicted by the orange box in Figure 2. Similar to textual cross-attention, in this module, the queries are mapped from visual frame features F , we separately map the keys and values from panning embedding e_{xy} and zooming embedding e_z for the same reason stated in the previous section. Through temporal cross-attention, the camera movement is infused into the visual features, which is then added back as a gated residual. We formulate this process as follows:

$$F = F + \tanh(\alpha) \cdot \text{TempCrossAttn}(F, e_{\text{cam}}) \quad (1)$$

$$\text{TempCrossAttn}(F, e_{\text{cam}}) = \text{Softmax} \left(\frac{Q[K_{xy}, K_z]^T}{\sqrt{d}} \right) [V_{xy}, V_z], \quad (2)$$

where $[,]$ denotes concatenation in sequence dimension, K_{xy}, K_z are key vectors, V_{xy}, V_z are value vectors mapped from the e_{xy}, e_z respectively, d is the feature dimension of Q , and α is a learnable scalar initialized as 0, ensuring that the camera movement is gradually learned from the pretrained state.

To learn camera movement while preserving the model’s prior knowledge, we freeze the original weights and train only the newly added camera embedder and camera module. These are conditioned on camera movement c_{cam} , and video caption c_{txt} . The training employs the diffusion noise-prediction loss:

$$\mathcal{L} = \mathbb{E}_{x_0, c_{\text{cam}}, c_{\text{txt}}, t, \epsilon \sim \mathcal{N}(0, I)} [\|\epsilon - \epsilon_\theta(x_t, c_{\text{cam}}, c_{\text{txt}}, t)\|_2^2], \quad (3)$$

where x_0 is the augmented input sample, t denotes the diffusion timestep, $x_t = \alpha_t x_0 + \sigma_t \epsilon$ is the noised sample at t , α_t and σ_t are time-dependent DDPM hyper-parameters [Ho et al. 2020], ϵ_θ is the diffusion model parameterized by θ .

3.3 Object Motion Control

We choose the bounding box as the control signal for object motion as it aligns best with our method, *i.e.*, modulating attention values within regions defined by boxes. Additionally, boxes are more efficient than dense conditions (*e.g.*, sketch maps require drawing skills) and are more expressive than sparse conditions (*e.g.*, key points lack the specification for object’s size).

While it is theoretically possible to train a box-conditioned T2V model similar to GLIGEN [Li et al. 2023]. However, unlike images, well-annotated video grounding datasets are less accessible, curating and training on large-scale dataset can be labor-intensive and computationally expensive. To bypass this issue, we opt to fully leverage the inherent priors of pretrained T2V models by steering the diffusion process to our desired result. Previous T2I works have demonstrated the ability to control an object’s spatial position by editing cross-attention maps [Balaji et al. 2022; Chen et al. 2024; Hertz et al. 2022; Kim et al. 2023; Ma et al. 2023; Sarukkai et al. 2024]. Similarly, we employ the spatial cross-attention modulation in T2V model for object motion crafting.

In cross-attention layers, the query features Q are derived from visual tokens, the key K and value features V are mapped from textual tokens. QK^\top constitutes an attention map, where the value at index $[i, j]$ reflects the response of the i -th image token feature to the j -th textual token feature. We modulate the attention map QK^\top as follows:

$$\text{CrossAttnModulate}(Q, K, V) = \text{Softmax} \left(\frac{QK^\top + \lambda S}{\sqrt{d}} \right) V, \quad (4)$$

where λ represents modulation strength, d is the feature dimension of Q , and S is the modulation term of the same size as QK^\top . It comprises two types of modulation: amplification and suppression.

Attention amplification. Considering the n -th object in the k -th frame, enclosed by the bounding box B_n^k , since we aim to increase the probability of the object’s presence in this region, we could amplify the attention values for the corresponding object words (indexed as T_n in the prompt) within the area B_n^k . Note that if there exists a background word, we treat it in the same way, and its region is the complement of the union of all the objects’ regions. Following the conclusion from DenseDiff [Kim et al. 2023], the scale of this amplification should be inversely related to the area of B_n^k , *i.e.*, smaller box area are subject to a larger increase in attention. Since our attention amplification is performed on box-shaped regions, which does not align with the object’s natural contours, we confine the amplification to the early stages (for timesteps $t \geq \tau$, τ is the cut-off timestep), as the early stage mainly focuses on generating coarse layouts. For $t < \tau$, we relax this control to enable the diffusion process to gradually refine the shape and appearance details.

Attention suppression. To mitigate the influence of irrelevant words on the specified region and prevent the unintended dispersion of object features to other areas, we suppress attention values for unmatched query-key token pairs (except start token `<sos>` and end token `<eos>` otherwise the video quality would be compromised). Different from attention amplification, attention suppression is applied throughout the entire sampling process to prevent mutual semantic interference, an potential issue in multi-object generation scenarios where the semantics of one object might inadvertently bleed into another. We will present the results and analysis in the ablation studies (Section 4.5).

Formally, the attention modulation term for the n -th object in the k -th frame $S_n^k[i, j]$ is formulated as:

$$S_n^k[i, j] = \begin{cases} 1 - \frac{|B_n^k|}{|QK^\top|}, & \text{if } i \in B_n^k \text{ and } j \in T_n \text{ and } t \geq \tau \\ 0, & \text{if } i \in B_n^k \text{ and } j \in T_n \text{ and } t < \tau \\ -\infty, & \text{otherwise} \end{cases} \quad (5)$$

where $|X|$ denotes the number of elements in matrix X . We perform such modulation for each object in every frame so that the complete spatial-temporal object trajectory can be determined. Note that although this modulation is independently performed in each frame, we observe that the generated videos remain continuous, thanks to the pretrained temporal layers which maintains temporal continuity.

4 EXPERIMENT

4.1 Experimental Setup

Implementation details. We adopt pretrained Zeroscope T2V model [Wang et al. 2023d] as our base model, integrating our proposed trainable camera embedder and module to facilitate camera movement learning, please refer to supplementary materials for training details. During the inference, we use DDIM sampler [Song et al. 2020] with $T = 50$ sampling steps and a classifier-free guidance scale of 9 [Ho and Salimans 2022]. The default attention control weight λ and cut-off timestep τ are 25 and $0.95T$ respectively. The output video size is $320 \times 512 \times 24$.

Datasets. For camera movement training, we use a subset from MovieShot [Rao et al. 2020], which contains 22k static-shot movie trailers, *i.e.*, the camera is fixed but the subject is flexible to move, ensuring that the training samples are devoid of original camera movement. Despite the limited number and category of the training data, our trained camera module is still able to adapt to general scenes. For camera control evaluation, we collected 200 scene prompts from the prompt set provided by [Chivileva et al. 2023]. For object control evaluation, we curated a benchmark of 200 box-prompt pairs, comprising varied box sizes, locations, and trajectories, with prompts primarily focusing on natural animals and objects.

Metrics. (1) To assess video generation quality, we employ FID-vid [Heusel et al. 2017] and FVD [Unterthiner et al. 2018]. The reference set consist of 2048 videos from MSRVTT [Xu et al. 2016] for the camera control task and 800 videos from AnimalKingdom [Ng et al. 2022] for the object control task. (2) To evaluate camera movement control, we introduce the flow error metric. We utilize VideoFlow [Shi et al. 2023], a state-of-the-art optical flow model, to extract flow maps from the generated videos. These are then compared against the ground truth flow maps, which are derived from the given camera movement parameters. (3) To measure the object-prompt alignment in object control task, we uniformly extract 8 frames per video sample and calculate the CLIP image-text similarity (CLIP-sim) [Hessel et al. 2021] within the box area, with a templated prompt “a photo of <obj>”, where <obj> corresponds to the object phrase. (4) To measure the object-box alignment, we employ Grounding DINO [Liu et al. 2023a] to detect object boxes in generated videos. We then calculate the mean Intersection over Union (mIoU) against the input boxes and compute the average precision score at the 0.5 IoU threshold (AP50).

Baselines. We compare our method with recent diffusion-based T2V models with the camera movement or object motion controllability, including AnimateDiff [Guo et al. 2023] (for camera movement), Peekaboo [Jain et al. 2023] (for object motion), and VideoComposer [Wang et al. 2023f] (for both).

4.2 Camera Movement Control

For camera movement control, we conduct comparisons with AnimateDiff and VideoComposer. For AnimateDiff, we use official pretrained LoRA motion modules, each dedicated to a specific type of camera movement but lacking support for precise control. For VideoComposer, we hand-craft a motion vector map based on the camera movement parameters, as demonstrated in its paper.

Table 1: Quantitative comparison for camera movement control evaluation.

	FVD ↓	FID-vid↓	Flow error↓
AnimateDiff	1685.40	82.57	-
VideoComposer	1230.57	82.14	0.74
Direct-a-Video (ours)	888.91	48.96	0.46

Qualitative comparison. We present side-by-side visual comparison with baselines in Figure 3. As can be seen, all the methods are capable of generating videos with the single type of camera movement, but AnimateDiff does not support hybrid camera movement (*e.g.*, pan+zoom) since its loaded motion module is dedicated to one type of camera movement only, while our method and VideoComposer can combine or switch the camera movement by altering the motion input, without the need for re-loading extra modules. In terms of precise control, both our method and VideoComposer can quantitatively control the camera speed. Specifically, VideoComposer [Wang et al. 2023f] requires a sequence of motion maps as input, while ours only requires three camera parameters. Moreover, in terms of disentanglement, our method’s camera control does not impact foreground objects, as we do not impose any motion constraints on them. In contrast, VideoComposer employs a global motion vector map, which often binds objects together with background movement. As shown in the 3rd column of Figure 3, the zebra in our results exhibits its independent motion from the camera, whereas in VideoComposer’s results (the 2nd column), the zebra is tied to the camera movement, so does the fish in the last 2nd column. Finally, our results also exhibit higher visual quality, a testament to the superiority of our base model.

Quantitative comparison. We report FVD, FID-vid, and Flow error in Table 1. Note that AnimateDiff is excluded from the flow error comparison due to its lack of quantitative control. Our results achieve the best FVD and FID-vid scores, indicating superior visual quality compared to baselines, and show more precise camera control, evidenced by a lower flow error.

4.3 Object Motion Control

For object motion control, we compare with VideoComposer [Wang et al. 2023f] and Peekaboo [Jain et al. 2023]. To enable VideoComposer generating object motion via boxes, we first convert object box sequences into dense flow maps, which are then processed into motion vector maps compatible with its input format. Peekaboo’s visual results are taken from their official website.

Qualitative comparison. We present visual comparison with related baselines in Figure 4. For static object generation, VideoComposer fails to generate the object in desired location (see the panda in the first column), without any motion hint, it works like a vanilla T2V model. While all methods are capable of generating a single moving object, challenges arise in multiple moving objects scenarios. Peekaboo is excluded from this comparison as its code is not implemented for multiple objects. VideoComposer does not support specifying individual motion for each object unlike our method (see the shark and jellyfish examples in the 7th and 8th

Direct-a-Video

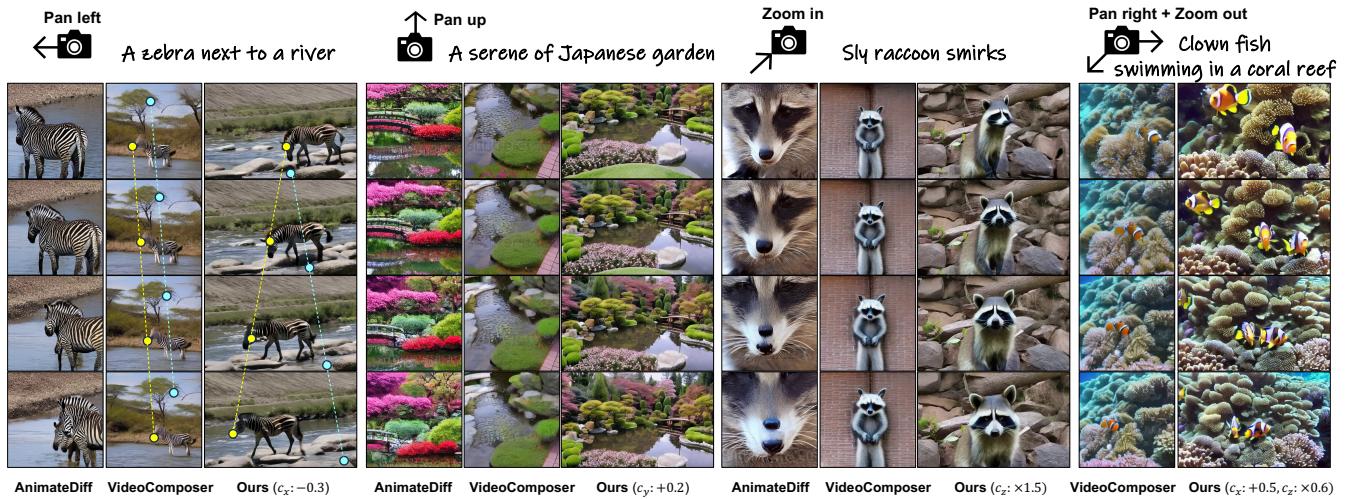


Figure 3: Qualitative comparison on camera movement control with related baselines. Our results in the third column show that the object motion (yellow lines) can be independent from the camera movement (cyan lines) unlike results by VideoComposer [Wang et al. 2023f] in the second column.

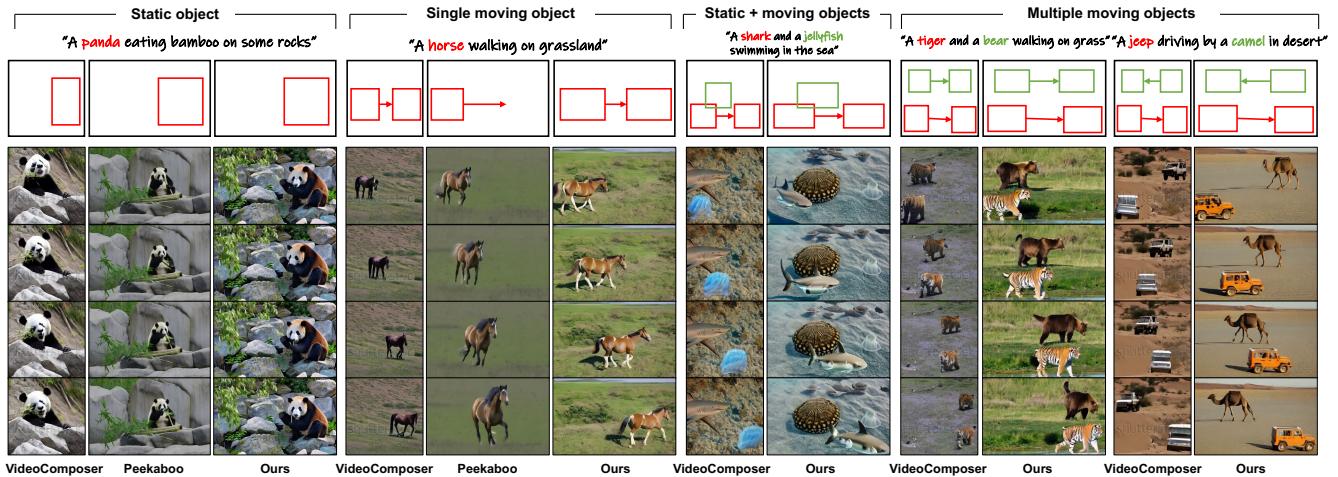


Figure 4: Qualitative comparison on object motion control with related baselines. Our method excels in handling cases involving more than one object.

columns). Moreover, its lack of explicit binding between objects and motion leads to two extra issues: semantic mixing and absence. Semantic mixing refers to the blending of one object’s semantics with another. This is exemplified in the 9th column, where tiger’s texture leaks into bear. Semantic absence occurs when an object does not appear as anticipated, a known issue in T2I/T2V models [Chefer et al. 2023]. For instance, in the 11th column, the expected camel is missing, replaced instead by a jeep. In contrast, our method effectively addresses these issues through ad-hoc attention modulation for each object, facilitating easier control over multiple objects’ motion.

Quantitative comparison. We report quality metrics (FVD, FID-vid) and grounding metrics (CLIP-sim, mIoU, AP50) in Table 2. In terms of quality, our method is comparable to Peekaboo, as both utilize the same superior model that outperforms VideoComposer’s. For object control, our method slightly surpasses VideoComposer and significantly exceeds Peekaboo by additionally incorporating attention amplification, in contrast to Peekaboo’s reliance on attention masking alone. We believe the use of amplification plays important role in improving grounding ability, as demonstrated in our ablation study (Section 4.5).

Table 2: Quantitative comparison for object motion control evaluation.

	FVD ↓	FID-vid↓	CLIP-sim↑	mIoU (%)↑	AP50 (%)↑
VideoComposer	1620.83	90.57	27.35	45.24	31.01
Peekaboo	1384.62	44.49	27.03	36.55	18.77
Direct-a-Video	1300.86	43.55	27.63	47.83	31.33

4.4 Joint Control of Camera Movement and Object Motion

Direct-a-Video features in jointly supporting the control of both camera movement and object motion, we demonstrate such capability in Figure 5. Given the same box sequence, our method can generate videos with varied combination of foreground-background motions. For example, Figure 5(a) illustrates that a static box does not always imply a static object, by setting different camera movements, our system can generate videos of a zebra standing still (2nd column), walking right (3rd column), or walking left (4th column). Similarly, Figure 5(b) suggests that a moving box does not necessarily indicate that the object itself is in motion, it could be stationary in its position while the camera is moving (last column). Existing works focused only on object often fail to differentiate between the object’s inherent motion and apparent motion induced by camera movement. In contrast, our method enables users to distinctly specify both camera movement and object motion, offering enhanced flexibility in defining overall motion patterns. More examples are provided in Figure 11 and our project page.

4.5 Ablation Study

We conduct ablation studies to evaluate several key components of our work.

Attention amplification. This is crucial for object localization, the absence of attention amplification results in a decrease of grounding ability, *i.e.*, the object is less likely to follow the boxes, as shown in the first row in Figure 6, and a decrease of metrics in Table 3.

Attention suppression. This is introduced to mitigate the unintended semantic mixing in multi-object scenarios, particularly when objects share similar characteristics. Since our attention amplification is applied only in the initial steps, and this constraint is subsequently relaxed. Without suppression, object A’s prompt feature can also attend to object B’s region, leading to semantic overlap. As shown in second row of Figure 6, where the tiger’s texture erroneously appears on the bear’s body. The third row shows that this issue can be resolved by enabling the attention suppression.

Camera embedding design. To assess the effectiveness of separately encoding panning (c_x, c_y) and zooming (c_z) movements in camera control as detailed in Section 3.2, we contrast this with a joint encoding approach. Here, $[c_x, c_y, c_z]$ are encoded into a single camera embedding vector using a shared MLP, followed by shared key-value projection matrix in the camera module. We train and evaluate the model with the same setting, we observed a reduced ability in camera movement control, with flow error increasing from

Table 3: Quantitative evaluation of attention amplification and suppression.

Attn amp.	Attn sup.	CLIP-sim ↑	mIoU (%)↑	AP50 (%)↑
✗	✓	25.82	15.35	3.46
✓	✗	27.49	38.87	10.25
✓	✓	27.63	47.83	31.33

0.46 to 1.68. This underscores the advantages of separate encoding for distinct types of camera movements.

5 LIMITATIONS

We consider several limitations of our method.

(1) For joint control, while our method provides disentangled control over object and camera motion, conflicts can sometimes arise in the inputs. For instance, in the top row of Figure 7, we attempt to maintain a static object (house) within a static box while simultaneously panning the camera to the left. Given these conflicting signals, our method ends up generating a moving house, which is unrealistic. This necessitates careful and reasonable user interaction.

(2) In camera control, due to the camera augmentation technique used in our method, which currently involves only 2D-panning and zooming, this limits the system’s ability to produce complex 3D camera movements that are out of this scope, *e.g.*, our method cannot generate camera movements like “panning around an object”. To overcome this constraint, we consider envisaging the adoption of more sophisticated augmentation algorithms, or curating a synthetic video dataset from a rendering engine given the camera movements, which we will leave in our future work.

(3) In object control, another issue arises when handling colliding boxes. In scenarios like the one depicted in the bottom row of Figure 7, where two boxes overlap, the semantics of one object (the bear) can interfere with another (the tiger). This issue can be mitigated by modulating attention on an adaptively auto-segmented region during the diffusion sampling process, rather than relying on the initial box region.

6 CONCLUSION

In this work, we propose Direct-a-Video, a text-to-video framework that addresses the previously unmet need for independent and user-directed control over camera movement and object motion. Our approach effectively decouples these two elements by integrating a self-supervised training scheme for temporal cross-attention layers tailored for camera movement control, with a training-free modulation for spatial cross-attention dedicated to object motion control. Experimental evaluations demonstrate the capability of our approach in separate and joint control of camera movement and object motion. This positions Direct-a-Video as an efficient and flexible tool for creative video synthesis with customized motion.

ACKNOWLEDGMENTS

This work was supported by a GRF grant (Project No. CityU 11208123) from the Research Grants Council (RGC) of Hong Kong, and research funding from Kuaishou Technology.



Figure 5: Joint control of object motion and camera movement. Given the same box sequence, by setting different camera movement parameters, our approach is capable of synthesizing videos that exhibit a diverse combination of foreground motion (yellow lines) and background motion (cyan lines). The user can create a well-defined overall video motion by distinctly specifying both object motion and camera movement using our method.

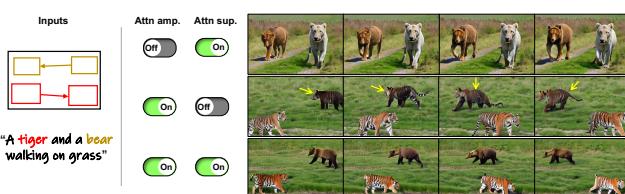


Figure 6: Effect of attention amplification and suppression. Without amplification (the first row), the objects do not adhere to boxes; Without suppression (the second row), tiger's texture mistakenly leaks into the bear's body. These issues are resolved with both enabled (the third row).

REFERENCES

- Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. 2022. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324* (2022).
- Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. 2023a.

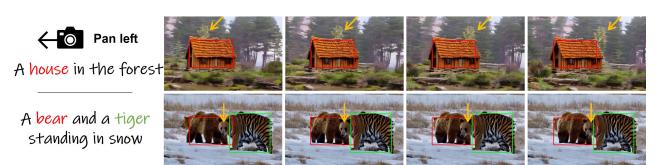


Figure 7: Limitations of our method. Top: conflicting inputs can lead to unreal results - a moving house. Bottom: Overlapping boxes may lead to object interfere - tiger with a bear head.

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

Duygu Ceylan, Chun-Hao P Huang, and Niloy J Mitra. 2023. Pix2video: Video editing using image diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 23206–23217.

- Wenhai Chai, Xun Guo, Gaoang Wang, and Yan Lu. 2023. Stablevideo: Text-driven consistency-aware diffusion video editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 23040–23050.
- Di Chang, Yichun Shi, Quankai Gao, Jessica Fu, Hongyi Xu, Guoxian Song, Qing Yan, Xiao Yang, and Mohammad Soleymani. 2023. MagicDance: Realistic Human Dance Video Generation with Motions & Facial Expressions Transfer. *arXiv preprint arXiv:2311.12052* (2023).
- Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. 2023. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–10.
- Minghao Chen, Ira Laina, and Andrea Vedaldi. 2024. Training-free layout control with cross-attention guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 5343–5353.
- 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).
- Weifeng Chen, Jie Wu, Pan Xie, Hefeng Wu, Jiashi Li, Xin Xia, Xuefeng Xiao, and Liang Lin. 2023b. Control-A-Video: Controllable Text-to-Video Generation with Diffusion Models. *arXiv preprint arXiv:2305.13840* (2023).
- Iya Chivileva, Philip Lynch, Tomas Ward, and Alan Smeaton. 2023. Text prompts and videos generated using 5 popular Text-to-Video models plus quality metrics including user quality assessments. (10 2023). <https://doi.org/10.6084/m9.figshare.24078045.v2>
- Yufan Deng, Ruida Wang, Yuhao Zhang, Yu-Wing Tai, and Chi-Keung Tang. 2023. DragVideo: Interactive Drag-style Video Editing. *arXiv preprint arXiv:2312.02216* (2023).
- Patrick Esser, Johnathan Chiu, Parmida Atighbehchian, Jonathan Granskog, and Anastasis Germanidis. 2023. Structure and content-guided video synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7346–7356.
- Mengyang Feng, Jinlin Liu, Kai Yu, Yuan Yao, Zheng Hui, Xiefan Guo, Xianhui Lin, Haolan Xue, Chen Shi, Xiaowen Li, et al. 2023. DreMoving: A Human Video Generation Framework based on Diffusion Models. *arXiv e-prints* (2023), arXiv-2312.
- Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. 2022. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618* (2022).
- Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. 2023. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arXiv:2307.10373* (2023).
- Yuchao Gu, Yipin Zhou, Bichen Wu, Licheng Yu, Jia-Wei Liu, Rui Zhao, Jay Zhangjie Wu, David Junhao Zhang, Mike Zheng Shou, and Kevin Tang. 2023. VideoSwap: Customized Video Subject Swapping with Interactive Semantic Point Correspondence. *arXiv preprint arXiv:2312.02087* (2023).
- 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).
- Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. 2022. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626* (2022).
- Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2021. ClipScore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718* (2021).
- Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems* 30 (2017).
- Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. 2022a. 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. *Advances in Neural Information Processing Systems* 33 (2020), 6840–6851.
- Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598* (2022).
- Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. 2022b. Video diffusion models. *arXiv:2204.03458* (2022).
- 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).
- Li Hu, Xin Gao, Peng Zhang, Ke Sun, Bang Zhang, and Liefeng Bo. 2023. Animate Anyone: Consistent and Controllable Image-to-Video Synthesis for Character Animation. *arXiv preprint arXiv:2311.17117* (2023).
- Yash Jain, Anshul Nasery, Vibhav Vineet, and Harkirat Behl. 2023. PEEKABOO: Interactive Video Generation via Masked-Diffusion. *arXiv preprint arXiv:2312.07509* (2023).
- Hyeonho Jeong, Geon Yeong Park, and Jong Chul Ye. 2023. VMC: Video Motion Customization using Temporal Attention Adaption for Text-to-Video Diffusion Models. *arXiv preprint arXiv:2312.00845* (2023).
- Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. 2021. Layered neural atlases for consistent video editing. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–12.
- Yunji Kim, Jiyoung Lee, Jin-Hwa Kim, Jung-Woo Ha, and Jun-Yan Zhu. 2023. Dense text-to-image generation with attention modulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7701–7711.
- Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. 2023. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1931–1941.
- Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. 2023. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 22511–22521.
- Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. 2023a. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499* (2023).
- Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. 2023b. Video-p2p: Video editing with cross-attention control. *arXiv preprint arXiv:2303.04761* (2023).
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).
- Wan-Duo Kurt Ma, JP Lewis, W Bastiaan Kleijn, and Thomas Leung. 2023. Directed diffusion: Direct control of object placement through attention guidance. *arXiv preprint arXiv:2302.13153* (2023).
- Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106.
- Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. 2023. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6038–6047.
- Chong Mou, Xiantao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. 2023. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453* (2023).
- Xun Long Ng, Kian Eng Ong, Qichen Zheng, Yun Ni, Si Yong Yeo, and Jun Liu. 2022. Animal Kingdom: A Large and Diverse Dataset for Animal Behavior Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 19023–19034.
- Hao Ouyang, Qiuyu Wang, Yuxi Xiao, Qingyan Bai, Juntao Zhang, Kecheng Zheng, Xiaowei Zhou, Qifeng Chen, and Yujun Shen. 2023. Codef: Content deformation fields for temporally consistent video processing. *arXiv preprint arXiv:2308.07926* (2023).
- Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xiantao Wang, Ying Shan, and Qifeng Chen. 2023. Fatezero: Fusing attentions for zero-shot text-based video editing. *arXiv preprint arXiv:2303.09535* (2023).
- 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).
- Anyi Rao, Jiaze Wang, Lining Xu, Xuekun Jiang, Qingqiu Huang, Bolei Zhou, and Dahua Lin. 2020. A unified framework for shot type classification based on subject centric lens. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI* 16. Springer, 17–34.
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10684–10695.
- Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. 2023. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 22500–22510.
- 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. *Advances in Neural Information Processing Systems* 35 (2022), 36479–36494.
- Vishnu Sarukkai, Linden Li, Arden Ma, Christopher Ré, and Kayvon Fatahalian. 2024. Collage diffusion. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 4208–4217.
- Xiaoyu Shi, Zhaoyang Huang, Weikang Bian, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. 2023. Videoflow: Exploiting temporal cues for multi-frame optical flow estimation. *arXiv preprint arXiv:2303.08340* (2023).
- 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).
- Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502* (2020).
- Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. 2023. Emergent Correspondence from Image Diffusion. *arXiv preprint arXiv:2306.03881* (2023).

- 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).
- Jun Wang, Bohan Lei, Liya Ding, Xiaoyin Xu, Xianfeng Gu, and Min Zhang. 2023a. Autoencoder-based conditional optimal transport generative adversarial network for medical image generation. *Visual Informatics* (2023).
- Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shwei Zhang. 2023d. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571* (2023).
- Tan Wang, Linjie Li, Kevin Lin, Yuanhao Zhai, Chung-Ching Lin, Zhengyuan Yang, Hanwang Zhang, Zicheng Liu, and Lijuan Wang. 2023b. DisCo: Disentangled Control for Realistic Human Dance Generation. *arXiv preprint arXiv:2307.00040* (2023).
- Wen Wang, Kangyang Xie, Zide Liu, Hao Chen, Yue Cao, Xinlong Wang, and Chunhua Shen. 2023c. Zero-shot video editing using off-the-shelf image diffusion models. *arXiv preprint arXiv:2303.17599* (2023).
- Xiang Wang, Hangjie Yuan, Shwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. 2023f. VideoComposer: Compositional Video Synthesis with Motion Controllability. In *Advances in Neural Information Processing Systems*. 7594–7611.
- Zhouxia Wang, Ziyang Yuan, Xiantao Wang, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. 2023e. MotionCtrl: A Unified and Flexible Motion Controller for Video Generation. *arXiv preprint arXiv:2312.03641* (2023).
- Yujie Wei, Shwei Zhang, Zhiwu Qing, Hangjie Yuan, Zhiheng Liu, Yu Liu, Yingya Zhang, Jingren Zhou, and Hongming Shan. 2023. Dreamvideo: Composing your dream videos with customized subject and motion. *arXiv preprint arXiv:2312.04433* (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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7623–7633.
- 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).
- Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5288–5296.
- Zhongcong Xu, Jianfeng Zhang, Jun Hao Liew, Hanshu Yan, Jia-Wei Liu, Chenxu Zhang, Jiashi Feng, and Mike Zheng Shou. 2023. MagicAnimate: Temporally Consistent Human Image Animation using Diffusion Model. *arXiv preprint arXiv:2311.16498* (2023).
- Han Yang, Ruimao Zhang, Xiaobao Guo, Wei Liu, Wangmeng Zuo, and Ping Luo. 2020. Towards photo-realistic virtual try-on by adaptively generating-preserving image content. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 7850–7859.
- Shiyuan Yang, Xiaodong Chen, and Jing Liao. 2023a. Uni-paint: A unified framework for multimodal image inpainting with pretrained diffusion model. In *Proceedings of the 31st ACM International Conference on Multimedia*. 3190–3199.
- Shuai Yang, Yifan Zhou, Ziwei Liu, and Chen Change Loy. 2023b. Rerender A Video: Zero-Shot Text-Guided Video-to-Video Translation. *arXiv preprint arXiv:2306.07954* (2023).
- Shengming Yin, Chenfei Wu, Jian Liang, Jie Shi, Houqiang Li, Gong Ming, and Nan Duan. 2023. Dragnuwa: Fine-grained control in video generation by integrating text, image, and trajectory. *arXiv preprint arXiv:2308.08089* (2023).
- Liang Yuan, Dingkun Yan, Suguru Saito, and Issei Fujishiro. 2024. DiffMat: Latent diffusion models for image-guided material generation. *Visual Informatics* (2024).
- Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3836–3847.
- Min Zhao, Rongzhen Wang, Fan Bao, Chongxuan Li, and Jun Zhu. 2023b. ControlVideo: Adding Conditional Control for One Shot Text-to-Video Editing. *arXiv preprint arXiv:2305.17098* (2023).
- Rui Zhao, Yuchao Gu, Jay Zhangjie Wu, David Junhao Zhang, Jiawei Liu, Weijia Wu, Jussi Keppo, and Mike Zheng Shou. 2023a. Motondirector: Motion customization of text-to-video diffusion models. *arXiv preprint arXiv:2310.08465* (2023).
- 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).

Appendix

A ADDITIONAL IMPLEMENTATION DETAILS

A.1 Camera Augmentation Details

Extracting camera movement parameters from real-world videos are computationally intensive, often requiring the cumbersome process of separating object motion from camera movement. To bypass these challenges, we propose a method of camera augmentation that simulates camera movement by algorithmically manipulating a stationary camera’s footage. In brief, the camera augmentation is implemented by altering the calculated cropping window across the video sequence captured by a stationary camera, thereby simulating the effect of camera movement in a computationally efficient manner. The detailed pseudo-code of this process is illustrated in Figure 8.

A.2 Training Details for Camera Control

Camera movement parameters sampling. During the training, we adopt the following sampling scheme for camera movement parameters $\mathbf{c}_{\text{cam}} = [c_x, c_y, c_z]$:

$$\begin{aligned} c_x &\sim \begin{cases} 0, & \text{with probability } \frac{1}{3}, \\ \text{Uniform}(-1, 1), & \text{with probability } \frac{2}{3}, \end{cases} \\ c_y &\sim \begin{cases} 0, & \text{with probability } \frac{1}{3}, \\ \text{Uniform}(-1, 1), & \text{with probability } \frac{2}{3}, \end{cases} \\ c_z &\sim \begin{cases} 1, & \text{with probability } \frac{1}{3}, \\ 2^\omega, & \text{with probability } \frac{2}{3}, \end{cases} \text{ where } \omega \sim \text{Uniform}(-1, 1). \end{aligned}$$

Note that each component is sampled independently.

Training scheme. We adopt pretrained Zeroscope T2V model [Wang et al. 2023d] as our base model. To facilitate camera movement learning while retain the pretrained state, only the newly added layers are trainable, which include camera embedder and camera module. To speed up the training, we use a coarse-to-fine strategy: we first train on videos of size $256 \times 256 \times 8$ (height \times width \times frames) for 100k iterations, then we resume training on videos of size $320 \times 512 \times 16$ and $320 \times 512 \times 24$ for 50k iterations each. The training is performed using a DDPM noise scheduler [Ho et al. 2020] with timestep t uniformly sampled from $[400, 1000]$, such preference to higher timesteps helps to prevents overfitting to low-level details, which are deemed non-essential for understanding temporal transitions. We employ an AdamW optimizer [Loshchilov and Hutter 2017] with a learning rate of 5e-5 and a batch size of 8 on 8 NVIDIA Tesla V100 GPUs.

A.3 Inference Details for Camera Control.

In the text-to-image sampling process, classifier-free guidance [Ho and Salimans 2022] is widely used to facilitate the text response in generated images, where the predicted noise is extrapolated from the unconditional branch (which uses null-text \emptyset_{txt}) towards the conditional branch (which uses normal prompt \mathbf{c}_{txt}). We additionally propose a similar technique to enhance the camera control capability of our model. Specifically, our conditional branch uses

Table 4: Assessment of attention amplification on different parts of UNet.

	E	M	D	E+M	M+D	E+D	E+M+D
CLIP-sim \uparrow	26.20	25.93	26.75	26.35	26.74	27.62	27.63
mIOU(%) \uparrow	30.90	14.28	29.25	31.71	28.98	49.06	47.83
AP50(%) \uparrow	5.50	0.27	5.75	6.61	6.13	30.04	31.33

desired camera parameters $\mathbf{c}_{\text{cam}} = [c_x, c_y, c_z]$, while the unconditional branch uses a static camera status $\emptyset_{\text{cam}} = [0, 0, 1]$ (*i.e.*, no panning or zooming). The predicted noise $\hat{\epsilon}_\theta$ at each sampling step is calculated as:

$$\begin{aligned} \hat{\epsilon}_\theta(\mathbf{z}_t, \mathbf{c}_{\text{cam}}, \mathbf{c}_{\text{txt}}, t) &= \epsilon_\theta(\mathbf{z}_t, \emptyset_{\text{cam}}, \emptyset_{\text{txt}}, t) \\ &+ s(\epsilon_\theta(\mathbf{z}_t, \mathbf{c}_{\text{cam}}, \mathbf{c}_{\text{txt}}, t) - \epsilon_\theta(\mathbf{z}_t, \emptyset_{\text{cam}}, \emptyset_{\text{txt}}, t)), \end{aligned} \quad (6)$$

where s is the guidance scale. On the other hand, unlike text conditioning, which is applied throughout the sampling process, we found that applying the camera conditioning in only a few initial steps is sufficient for controlling camera movement, as the general temporal transitions is already determined in early stages. Formally, during the inference, we bypass the camera module when t is less than a certain threshold, which we refer to as the camera control cut-off timestep, we empirically set this value to 0.85T.

B ADDITIONAL ABLATION STUDIES

We conduct additional ablation studies to validate the settings of our method.

Which layers for attention amplification? To determine which layers to apply the attention amplification, we divide the U-Net into three parts: encoder (E), middle layer (M), and decoder (D). We applied attention amplification to various combinations of these three and assessed their impact on the CLIP-sim, mIOU and AP50 scores. The results are presented in Table 4. We observed that applying attention amplification to either the encoder or the decoder significantly enhances object responsiveness, as evidenced by higher values across all metrics. Controllability is further strengthened when attention amplification is applied to both components. The middle layer has a comparatively smaller influence, incorporating middle layer does not bring noticeable statistic change. Consequently, we apply attention amplification across all layers.

Attention amplification hyper-parameters. In attention amplification, the strength λ and cut-off timestep τ are two hyper-parameters. Generally, lower τ and higher λ will increase the strength of attention amplification. To determine a proper choice of hyper-parameters, we conduct tests with different combinations of λ and τ . Visual examples are provided in Figure 10. We observed that object responses are more sensitive to the value of τ than to λ . As illustrated in the 1st and 2nd rows, over-responsiveness in box regions typically occurs for $\tau < 0.9$. This is because the early sampling stage in the diffusion model plays a significant role in determining the coarse layout of the output image or video; thus, applying amplification for an extended duration results in over-responsiveness in the box region. We also report CLIP-sim and mIOU metrics in Table 5, as can be seen, setting $\tau > 0.9$ results in better semantic quality, as evidenced

```

Function aug_with_cam_motion(src_video, cx, cy, cz, h, w):

    # Parameters:
    # src_video: Source video, a tensor with the size of [frames (f), 3, src_height, src_width]
    # cx: Horizontal translation ratio (-1 to 1)
    # cy: Vertical translation ratio (-1 to 1)
    # cz: Zoom ratio (0.5 to 2)
    # h: Height of the augmented video
    # w: Width of the augmented video
    # Returns: Augmented video, a tensor with the size of [f, 3, h, w]

    # Get source frame number, width and height from src_video
    f, src_h, src_w = src_video.shape[0], src_video.shape[2], src_video.shape[3]

    # Initialize camera boxes for frame cropping
    cam_boxes = zeros(f, 4) # f frames, 4: [x1,y1,x2,y2]

    # Calculate dynamic cropping relative coordinates for each frame
    # The first frame coordinates is the reference, which is always [0,0,1,1].
    cam_boxes[:, 0] = linspace(0, cx + (1 - 1/cz) / 2, f) # x1, top-left x
    cam_boxes[:, 1] = linspace(0, cy + (1 - 1/cz) / 2, f) # y1, top-left y
    cam_boxes[:, 2] = linspace(1, cx + (1 + 1/cz) / 2, f) # x2, bottom-right x
    cam_boxes[:, 3] = linspace(1, cy + (1 + 1/cz) / 2, f) # y2, bottom-right y

    # Compute the minimum and maximum relative coordinates
    min_x = min(cam_boxes[:, 0::2])
    max_x = max(cam_boxes[:, 0::2])
    min_y = min(cam_boxes[:, 1::2])
    max_y = max(cam_boxes[:, 1::2])

    # Normalize the camera boxes
    normalized_boxes = zeros_like(cam_boxes)
    normalized_boxes[:, 0::2] = (cam_boxes[:, 0::2] - min_x) / (max_x - min_x)
    normalized_boxes[:, 1::2] = (cam_boxes[:, 1::2] - min_y) / (max_y - min_y)

    # Initialize a tensor for the new frames
    augmented_frames = zeros(f, 3, h, w)

    # Process each frame
    for i in range(f):
        # Calculate the actual cropping coordinates
        x1, y1, x2, y2 = normalized_boxes[i] * tensor([src_w, src_h, src_w, src_h])

        # Crop the frame according to the coordinates
        crop = src_video[i][:, int(y1):int(y2), int(x1):int(x2)]

        # Resize the cropped frame and store it
        augmented_frames[i] = interpolate(crop, size=(h, w), mode='bilinear')

return augmented_frames

```

Figure 8: Pseudo-code for the camera augmentation function.

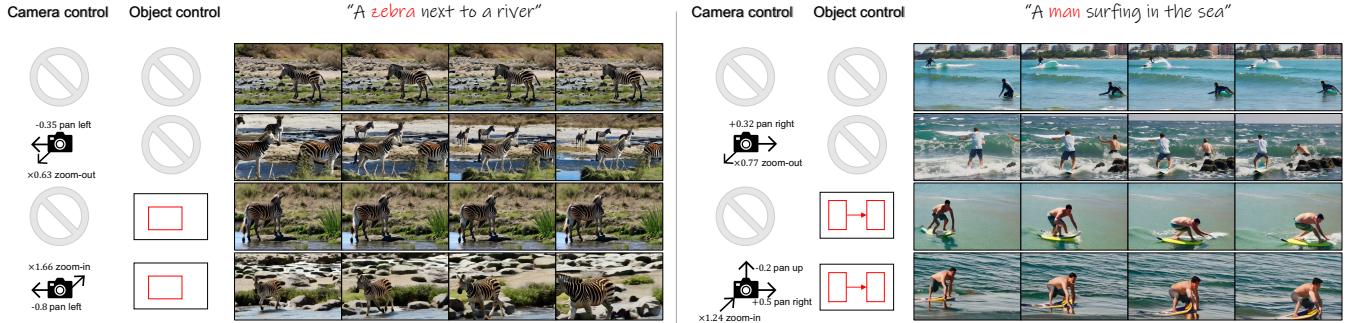


Figure 9: Qualitative comparison of generated videos using the same prompt but different controls. 1st row: base model, i.e., no control; 2nd row: camera control only; 3rd row: object control only; 4th row: camera + object control. Adding control introduces more dynamic content without noticeable quality degradation.

by higher CLIP-sim scores and the visual results. On the other hand, setting $\lambda \geq 10$ generally yields higher mIOU values. It is important to note that while a higher mIOU indicates better object grounding ability, it does not necessarily equate to better object quality. In summary, we empirically determine that $\tau \in [0.9T, 0.95T]$ and $\lambda \in [10, 25]$ are generally appropriate for most cases.

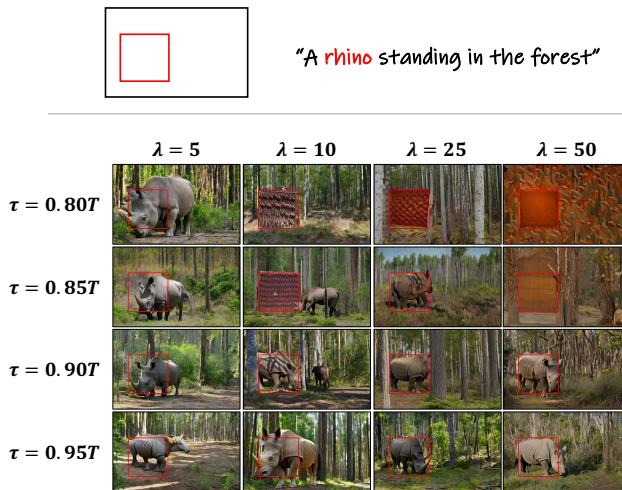


Figure 10: Effect of attention amplification strength λ and cut-off timestep τ (only the first frame is showed).

Table 5: CLIP-sim and mIOU metrics tested on different attention amplification hyper-parameters.

	$\lambda = 5$	$\lambda = 10$	$\lambda = 25$	$\lambda = 50$		
	CLIP-sim	mIOU	CLIP-sim	mIOU	CLIP-sim	mIOU
$\tau = 0.80T$	26.81	27.99	24.91	47.71	24.65	49.76
$\tau = 0.85T$	26.85	25.72	25.31	42.52	25.03	47.69
$\tau = 0.90T$	26.78	26.74	26.15	40.83	26.60	45.50
$\tau = 0.95T$	26.48	21.82	27.49	41.61	27.63	47.83
					27.32	43.92

Table 6: Quantitative evaluation for camera/object control on video quality.

	Base	Cam	Obj	Cam+Obj
FID-vid ↓	41.12	44.95	43.55	41.20
FVD ↓	1104.36	1204.55	1300.86	1280.88

Effect of adding control on quality. To evaluate the impact of incorporating camera/object control on the video quality, we calculate the FVD and FID-vid score under four different settings: (1) Base: no control involved, i.e., the vanilla model; (2) Cam: only camera movement control is involved (with random camera parameters); (3) Obj: only object control is involved; and (4) Cam+Obj: both camera and object control are enabled. The quality metrics are presented in Table 6. The statistic shows that adding control may have a minor influence but not so significant, as the metrics are approximately in the same level with minor fluctuations. We also present visual examples in Figure 9. As can be seen, adding control does not result in a noticeable degradation of quality; on the contrary, it introduces more dynamic content into the generated videos compared to the base model.

C ADDITIONAL RESULTS

We show additional results in Figure 11. Please refer to our project page for dynamic results.

Direct-a-Video

Prompt	Camera movement	Object motion	Generated Videos					
Camera Movement	"A waterfall in a beautiful forest with fall foliage"							
	"A waterfall in a beautiful forest with fall foliage"							
	"A villa in a garden"							
	"A villa in a garden"							
	"A UFO is flying in the sky "							
	"Firework is exploding in the night"							
	"A drone is landing on the grassland "							
	"A jeep is climbing over the rocks "							
	"A tiger walking by a campfire in the forest "							
	"A running dog and a flying UFO on the beach "							
Single Object Motion	"A tiger and a zebra walking on grass "							
	"Clown fish and jellyfish swimming in ocean with coral reef"							
	"A bear walking on snow "							
	"A leaf is flowing in the sky "							
	"A shark and a jellyfish swimming in the sea "							
Joint Control	"An elephant and a flying UFO "							

Figure 11: Additional results of camera movement control and object motion control.