# TrailBlazer: Trajectory Control for Diffusion-Based Video Generation

WAN-DUO KURT MA, Victoria University of Wellington, New Zealand
J. P. LEWIS, Victoria University of Wellington, New Zealand and NVIDIA Research, USA
W. BASTIAAN KLEIJN, Victoria University of Wellington, New Zealand

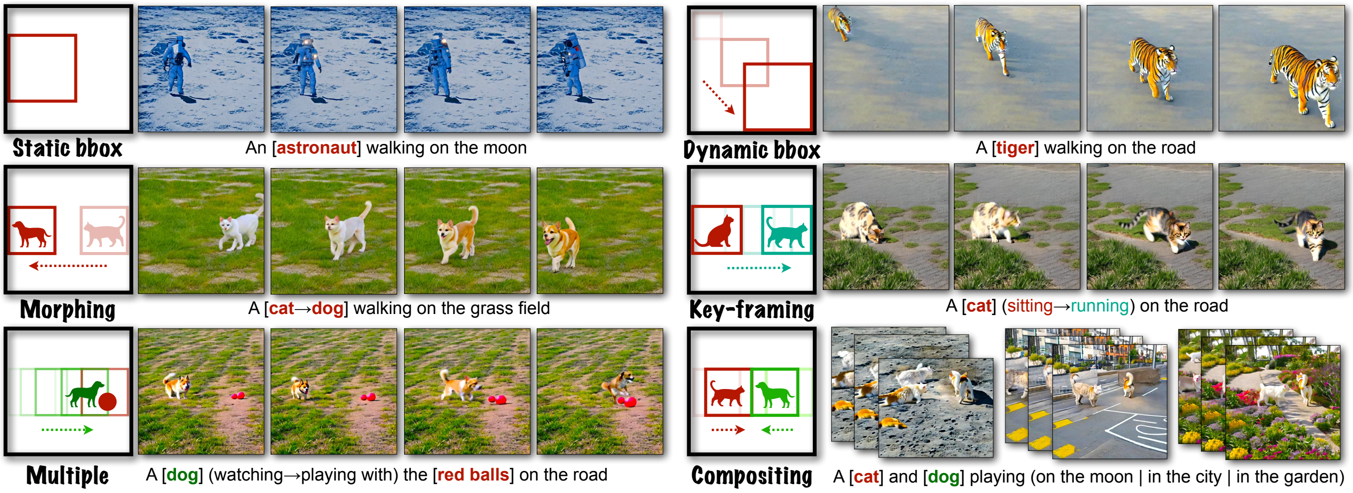
Static bbox
An [astronaut] walking on the moon
Dynamic bbox
A [tiger] walking on the road
Morphing
A [cat→dog] walking on the grass field
Key-framing
A [cat] (sitting→running) on the road
Multiple
A [dog] (watching→playing with) the [red balls] on the road
Compositing
A [cat] and [dog] playing (on the moon | in the city | in the garden)

Fig. 1. *TrailBlazer* extends a pre-trained video diffusion model to introduce trajectory control over one or multiple subjects. Its primary contribution lies in the ability to animate the synthesized subject using a bounding box *(bbox),* whether it remains static (Top-left) or dynamic in terms of location and bbox size (Top-right), morphing for subject interpolation (Middle-left), and varied movement speed (Middle-right). The moving subjects fit naturally within an environment specified by the overall prompt (Bottom-right). Additionally, the speed of the subjects can be controlled through keyframing (Bottom-left).

Large text-to-video (T2V) models such as Sora have the potential to revolutionize visual effects and the creation of some types of movies. Current T2V models require tedious trial-and-error experimentation to achieve desired results, however. This motivates the search for methods to directly control desired attributes. In this work, we take a step toward this goal, introducing a method for high-level, temporally-coherent control over the basic trajectories and appearance of objects. Our algorithm, *TrailBlazer,* allows the general positions and (optionally) appearance of objects to be controlled simply by keyframing approximate bounding boxes and (optionally) their corresponding prompts. Importantly, our method does not require a pre-existing control video signal that already contains an accurate outline of the desired motion, yet the synthesized motion is surprisingly natural with emergent effects including perspective and movement toward the virtual camera as the box size increases. The method is efficient, making use of a pre-trained T2V model and requiring no training or fine-tuning, with negligible additional computation. Specifically, the bounding box controls are used as soft masks to guide manipulation of the self-attention and cross-attention modules in the video diffusion model. While our visual results are limited by those of the underlying model, the algorithm may generalize to future models that use standard self- and cross-attention components.

CCS Concepts: • **Computing methodologies → Neural networks**; **Computer graphics**.

Additional Key Words and Phrases: Denoising diffusion, text-to-video generative models, artist guidance.

**ACM Reference Format:**
Wan-Duo Kurt Ma, J. P. Lewis, and W. Bastiaan Kleijn. 2024. TrailBlazer: Trajectory Control for Diffusion-Based Video Generation. In *SIGGRAPH Asia 2024 Conference Papers (SA Conference Papers '24), December 3–6, 2024, Tokyo, Japan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3680528.3687652

## 1 INTRODUCTION

Advancements in generative models for text-to-image (T2I) have been dramatic [2, 35–37]. Recently, text-to-video (T2V) systems such has Sora [31] have made remarkable strides, enabling the automatic generation of videos based on textual prompt descriptions [8, 13, 15, 31, 54]. These methods have the potential of revolutionizing visual effects and certain other aspects of movie making.

On the other hand, T2I methods do not provide full control over characteristics of the synthesized image, and repeated trial-and-error experimentation with different prompts and generally needed.

Authors' Contact Information: Wan-Duo Kurt Ma, Victoria University of Wellington, New Zealand, mawand@ecs.vuw.ac.nz; J. P. Lewis, Victoria University of Wellington, New Zealand and NVIDIA Research, Santa Clara, USA, noisebrain@gmail.com; W. Bastiaan Kleijn, Victoria University of Wellington, New Zealand, bastiaan.kleijn@vuw.ac.nz.

*SA Conference Papers '24, December 3–6, 2024, Tokyo, Japan*

ACM ISBN 979-8-4007-1131-2/24/12
https://doi.org/10.1145/3680528.3687652

Unfortunately, this issue is likely to be exacerbated for video synthesis. In one experimental short film made with Sora it was found that each several-second clip of the movie required about 300 synthesis attempts, as well as subsequent manual post-processing [40].

Control over *position and trajectory* of objects is a particular weak point of T2I and T2V models. Arguably this arises in large part from the fact that human language often omits descriptions of positions from the reference point of the viewer. For example, we typically might say “please put it down over there” [pointing to a location] rather than saying “please put it in the lower left quadrant of my field of view”. On the other hand, control over the spatial layout and trajectories of objects (as seen from the camera viewpoint) is *necessary* for understandable narration of a story [1].

Control over location and trajectory can be approached by providing *either* low/level or high/level guidance signals, and both approaches have their advantages. Visual effects artists typically require precise low-level control, and extensions [16] of the widely used ControlNet [61] are suitable for this purpose. These methods fine-tune the base T2I or T2V model to accept conditioning in the form of edge maps, depth maps, or other signals. On the other hand, producing such detailed signals for a video may be difficult. For example, if the prompt describes a *large dragon attacking a castle*, it is not possible to obtain the edge maps from an existing video, and sketching these edges in a temporally consistent way would be challenging and laborious even for artists.

The complementary approach of high-level control has its uses, both for casual users, and even for professional use – for example an actor might be directed to move from the door to a table while expressing some emotion, but the actor’s per-frame silhouette edges would not be specified by the director. To address this need, we introduce a high-level interface for the control of object trajectories in synthesized videos (Fig. 1). Users simply provide soft bounding boxes *(bboxes)* specifying the desired position of an object at several times (keyframes) in the video, together with optional text prompt(s) describing the desired object at the corresponding times. The provided bboxes are interpolated between the keyframes, resulting in smooth motion and size changes of the object, such as a cat in Fig. 1 (middle-right) that is initially sitting and then runs to the right. Note that the bboxes are implemented as *soft constraints* in our algorithm. *This imprecision is* necessary *for high-level control* – if the bbox exactly bounds the object it would require the user to precisely specify the aspect ratio of objects under perspective (a difficult task). If more than one different text prompt is provided, embeddings of these prompts are also interpolated, resulting in a “morphing” effect such as the cat → dog transformation in Fig. 1.

Our algorithm, *TrailBlazer*, involves editing *both spatial and temporal attention maps* for each specific object during the initial denoising diffusion steps to concentrate activation at the desired object location. Our method builds on previous works. We use the pre-trained ZeroScope model [6] (a fine-tuned version of [48]), as our underlying model. A body of previous and concurrent works have addressed guiding object position in image generation models, including [2, 3, 21, 25, 45, 55, 59, 62]. TrailBlazer most closely resembles the cross-attention injection used in [25], and we adopt some notation from that paper, however, our algorithm is both simpler and addresses a different problem. Specifically we target controlling trajectories and object attributes in *videos*, which requires a different approach to control temporal cross-frame attention. Our work also does not require any inference-time optimization as we found the results to be stable without need to optimize weights on individual cross-attention maps. While the reason our simpler method succeeds is unknown, we believe that satisfying the soft constraints over multiple frames may act as a regularizer.

Our contributions include:

- **Novelty.** We introduce a novel approach employing high-level bounding boxes to guide the subject in diffusion-based video synthesis. This approach is suitable for casual users, as it avoids the need to record or draw a frame-by-frame positioning control signal. In contrast, low-level guidance signals such as the widely used edge and depth maps in T2I models have disadvantages for video: it is difficult for non-artists to draw these signals in a temporally consistent way, while obtaining them by processing an existing video limits the synthesized motion to copies of existing videos.
- **Position, size, and prompt trajectory control.** Our approach enables users to position the subject by keyframing its bounding box. The size of the bbox can also be controlled, thereby producing perspective effects (Figs. 1, 6). The text prompt can be similarly keyframed to influence the behavior and identity of the subject in the synthesized video (see the *sitting* → *running* and *cat* → *dog* examples in Fig. 1).
- **Simplicity.** Our method operates by directly editing the spatial and temporal attention in the pre-trained denoising UNet. It requires no training or optimization, and the core algorithm can be implemented in less than 200 lines of code.

## 2 RELATED WORK

### 2.1 Text-to-Image (T2I)

Denoising diffusion models construct a stochastic [14, 42, 44] or deterministic [43] mapping between the data space and a corresponding-dimension multivariate Gaussian. Signals are synthesized by sampling from a normal distribution and performing a sequence of denoising steps. A number of works [29, 30, 35, 38] have performed T2I synthesis using images conditioned on the text embedding from a model such as CLIP [34]. Efficiency is significantly improved in the Latent Diffusion Model [36] (LDM) by performing the diffusion computation in the latent space of a carefully trained variational autoencoder. LDM was trained with a large scale dataset, resulting in the widely adopted Stable Diffusion (SD) system. We omit the basic diffusion derivation as tutorials are available, e.g., [52].

Despite the success of image generation using SD, it is widely acknowledged that SD lacks full controllability. Synthesizing multiple objects is particularly challenging and often results in missing objects or objects with incorrect attributes. Controllability has been greatly improved with methods such as ControlNet [61] and others [27] that introduce additional layers to fine-tune an existing model to accept various control inputs, as well as methods such as DragDiffusion [41] that provide control via inference-time optimization.

The methods of [3, 25, 45, 55, 59, 62] have addressed the layout-to-image (L2I) issue using few-shot learning. Directed Diffusion [25], BoxDiff [55], and MultiDiffusion [3] use coarse bboxes to control subject position, achieving good results by manipulating the spatial latent and text embeddings cross attention maps [11].

## 2.2 Text-to-Video (T2V)

Text-to-video (T2V) synthesis is generally more difficult than T2I due to the difficulty of ensuring temporal consistency and the requirement for a large paired text and video dataset. A number of T2V methods [9, 10, 15, 17, 19, 33, 47, 57] are extensions of T2I models such as SD. Some works [5, 24] also introduce 3D convolutional layers in the denoising UNet to learn temporal information. Imagen Video [13] achieves higher resolution by computing temporal and spatial super-resolution on initial low resolution videos. VideoLDM [5] and ModelScope [24] insert a temporal attention layer by reshaping the latent tensor. Text2Video-Zero [19] and FateZero [33] investigate how temporal coherence can be improved by cross frame attention manipulation with pre-trained T2I models. [9] addresses the same problem by introducing temporal correlation in the diffusion noise. The advent of Sora [31] demonstrated the remarkable potential of T2V models and is impacting thinking among professional movie creators [40].

Recently several works have been proposed to solve the controllability in video synthesis problem by using pre-trained models together with low-level conditioning information such as edge or depth maps. Control-A-Video [7] and MagicProp [56] use depth maps with ControlNet to train a temporal-aware network. Text2-Video-Zero [19] partially achieves controllability by initializing the latent frames conditioned on the first frame with applied linear translation. However, the control is indirect and requires two steps. First, the user first needs to find the subject's numerical location, and then adjust a translation offset. Distinct from the methods above, we use an attention injection method to guide the denoising path rather than optimization, and in general this is robust to different random seeds. Peekaboo [18] is a current state-of-the art method for providing pretrained video models with spatio-temporal location control. Both Peekaboo and TrailBlazer guide subjects by manipulating attention, however the formulations differ in many details. Peekaboo's use of an infinite negative attention injection in the background regions appears to often result in backgrounds with missing detail.

Additionally, there are training-based methods like DragNUWA [60], TrackDiffusion [20], MotionCtrl [51], and VideoComposer [50] that implement subject controllability with various approaches. Specifically, MotionCtrl and DragNUWA utilize trajectory paths to guide the synthesis process instead of bounding boxes.

In Sec. 4, we will provide both quantitative and visual evidence to demonstrate the better controllability and quality of our results. Other very recent preprints that address the layout-to-video problem in differing ways include [22, 49, 58]. We do not compare against these concurrent works because their source was not available at the time of writing.

# 3 METHOD

TrailBlazer is based on the open-source pre-trained model ZeroScope [6]. This is a fine-tuned version of ModelScope [24], known for its ability to generate reasonable quality videos without temporal flickering. It is noteworthy that TrailBlazer preserves this desirable temporal coherence. TrailBlazer does not require any training, optimization, or low-level control signals such ControlNet's edge or depth maps [61]. On the contrary, all that is required from the user is the prompt and an approximate bounding box (bbox) of the subject. Bboxes and corresponding prompts can be specified at several points in the video, and these are treated as *keyframes* and interpolated to smoothly control both the motion and prompt content.

We use the following notation conventions: Bold capital letters (e.g., $\mathbf{M}$) denote a matrix or a tensor depending on the context, vectors are represented with bold lowercase letters (e.g., $\mathbf{m}$), and scalars are denoted as lowercase letters (e.g., $m$). We use superscripts to denote an indexed tensor slice (e.g., $\mathbf{M}^{(i)}$). A synthesized video is composed of a number of images ordered in time. The individual images will be referred to as *frames*, and the collection of corresponding times is the *timeline*. Spatial or temporal attention will be informally referred to as *correlation*. Familiarity with modern diffusion models and common architecture components such as attention is assumed.

Our method draws significant inspiration from visual inspection of cross and self-attention maps in the underlying pre-trained video model [6, 48]. Consider the video in the upper left of Fig. 2, generated from the prompt "an [**astronaut**] walking on the moon". The prompt cross-attention for the word "astronaut" at the final denoising step, denoted as PA-Cross, is highlighted in the left of the second row and reflects the overall position of the subject. The right side of the first row displays "self-frame" temporal attention maps, denoted as TA-Self, which broadly align with PA-Cross.

The right side of the second row of Fig. 2 presents the visualization of "cross-frame" temporal attention maps, denoted as TA-Cross, illustrating the attention between the first frame and subsequent frames in the video. As the distance between frames increases, the attention becomes less correlated in the subject area but remains strongly correlated in the background area. This observation aligns with the video shown in the left of the first row, where the background remains nearly static while the astronaut's position varies frame by frame. TA-Self and TA-Cross are formally defined in Sec. 3.3, where we will consider temporal attention in more detail.

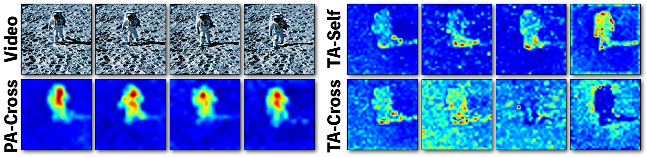

Fig. 2. **Inspiration for our method.** We draw inspiration from inspection of the prompt cross-attention (PA-Cross), and the attention maps of self-frame attention (TA-Self) and cross-frame attention (TA-Cross). Each subfigure shows frames 1, 4, 16, 24 of a 24-frame video. TA-Cross represents the "correlation" of these four frames with the first frame of video clip.

## 3.1 Pipeline

As mentioned above, keyframing [53] is a technique that defines properties of images at particular frames (keys) in a timeline and then automatically interpolates these values to achieve a smooth

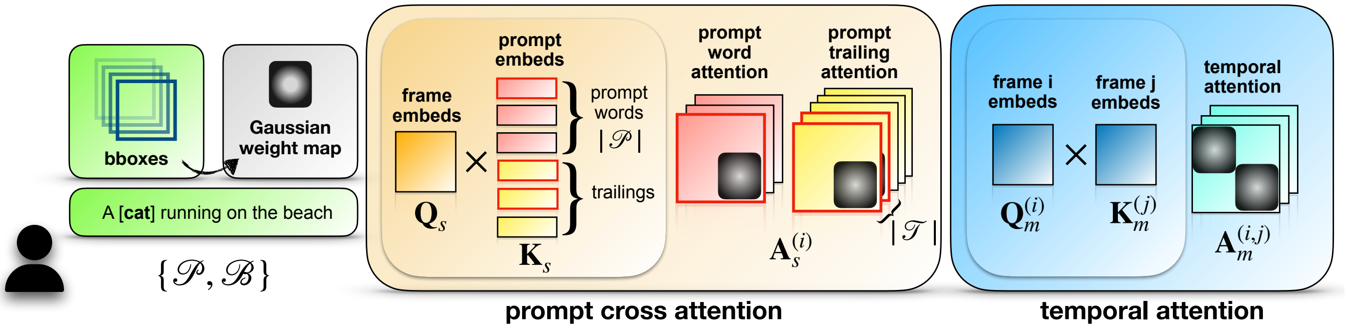

Fig. 3. **Pipeline Overview.** This figure highlights the central components of prompt cross-attention editing (middle, in the blanched almond-colored section) and temporal attention editing (right, in the blue section). This pipeline is exclusively applied in the initial denoising steps based on a user-provided bbox sequence for a user-selected subject word (left, green section). The objective is to alter the attention map (e.g., $\mathbf{A}_s, \mathbf{A}_m$) using a Gaussian weighting within a user-specified bbox. The example shows guidance from one prompt word attention map and two trailing attention maps as highlighted in red.

transition between the keys. Keyframing is commonly used in movie animation and visual effects because it eases the artist's workload and creates smooth motion that is difficult to achieve with direct image editing. Our system takes advantage of this principle, and asks the user to specify several keys, consisting of bboxes and the associated prompts, describing the subject location and appearance or behavior at the particular times. For instance, as shown in Fig. 1 (Middle-right), the video of the cat initially sitting on the left, then running to the right, is achieved simply by placing keys at only three frames. The sitting cat in the first part is created with two bboxes on the left at the start and middle of the timeline, both associated with "sitting." A third keyframe at the end places a bbox on the right with the prompt changing to "running." This results in the cat smoothly transitioning from sitting to running in the second part of the video.

We use the pre-trained ZeroScope model [6] in all our experiments with no neural network training, finetuning, or optimization at inference time. The prompt cross-attention and the temporal attention in our pipeline (Fig. 3) are discussed in detail in Sec. 3.2 and Sec. 3.3, respectively. All prompt and temporal attention editing is performed in the early steps $t \in \{T, ..., T - N_S\}$, and $t \in \{T, ..., T - N_M\}$ of the backward denoising process, where $T$ is the total number of denoising time steps, and $N_S$, and $N_M$ are hyperparameters specifying the number of steps of prompt and temporal attention editing. The parameter settings are detailed in our supplementary material.

In the subsequent sections we describe how our algorithm is implemented by modifying the prompt and temporal attention in a pre-trained diffusion model. Please refer to [14, 36, 43, 52] for background on overall diffusion model architectures.

Our system processes a set of keyframes, consisting of bbox regions $\mathcal{R}_f$ and associated prompts $\mathcal{P}_f$ at a frame $f$, where $f$ denotes the video frame index $f \in \{1, ..., N_F\}$. Users are required to specify a minimum of two keyframes: one at the start and one at the end of the video sequence. The information in these keyframes is linearly interpolated, resulting in the bbox $\mathcal{B}_f$ and the prompt text embedding $y(\mathcal{P}_f)$ at each frame, where $y(\cdot)$ denotes the text encoder. To enhance readability, we omit the subscript $f$ when discussing the core method.

A region $\mathcal{R}$ is characterized by a set of parameters $\mathcal{R} = \{\mathcal{B}, \mathcal{I}, \mathcal{T}\}$: a set of bbox positions, $\mathcal{B}$, the indices of the subject we would like to constrain, $\mathcal{I}$, and the indices of the *trailing maps* $\mathcal{T}$ to enhance controllability as described in the next paragraph. The subject indices $\mathcal{I} \subset \{i | i \in \mathbb{N}, 1 \leq i \leq |\mathcal{P}|\}$, are 1-indexed with the associated word in the prompt. For example, $\mathcal{I} = \{1, 2\}$ is associated with "a", "cat" in the prompt "a cat sitting on the car".

The trailing attention maps indices $\mathcal{T} \subset \{i | i \in \mathbb{N}, |\mathcal{P}| < i \leq N_P\}$ is the set of indices corresponding to the cross-attention maps generated without a prompt word association, where $N_P$ denotes the maximum prompt length that a tokenizer model can take. $N_P = 77$ in the case of CLIP [34]. The trailing attention maps serve as a means of controlling the spatial location of the synthesized subject and its attributes. A larger trailing indices set $|\mathcal{T}|$ provides greater controllability but comes with the risk of failed reconstruction [25].

A bbox $\mathcal{B} = \big\{(x, y) \,|\, b_{\text{left}} \times w \leq x \leq b_{\text{right}} \times w,\; b_{\text{top}} \times h \leq y \leq b_{\text{bottom}} \times h\big\}$, is a set of all pixel coordinates inside the bbox of resolution $w \times h$. $\mathcal{B}$ is represented as a tuple of the four scalars representing the boundary of the bbox $\mathbf{b} = (b_{\text{left}}, b_{\text{top}}, b_{\text{right}}, b_{\text{bottom}})$, where $b_{\text{left}}, b_{\text{top}}, b_{\text{right}}, b_{\text{bottom}} \in [0, 1]$ specify the bbox relative to the synthesis resolution. The height $h$ and width $w$, are defined by the resolution of the UNet intermediate representation [36].

### 3.2 Prompt cross-attention Guidance

The prompt cross-attention modules are implemented in the denoising UNet module of [36]. This module finds the cross-attention between the query $\mathbf{Q}_s \in \mathbb{R}^{N_F \times d_h \times d}$ obtained from the SD latent $\mathbf{z}_t$, and the key-value pair $\mathbf{K}_s, \mathbf{V}_s \in \mathbb{R}^{N_F \times N_P \times d}$ of the $N_P$ prompt words from the text model, where $d$ is the feature dimension of the keys and queries. The cross-attention map [11] is then defined as $\mathbf{A}_s = \text{Softmax}(\mathbf{Q}_s \mathbf{K}_s^T / \sqrt{d}) \in \mathbb{R}^{N_F \times d_h \times N_P}$,[1] where $d_h \equiv w \times h$, defined by the spatial height and width at the specific layer. We omit the batch size and the number of attention heads [46] for simplicity.

Given the set of indices of subject prompt words $\mathcal{I}$ and trailing maps $\mathcal{T}$, each cross-activation component at location $(x, y)$ in $\mathbf{A}_s$ is

[1] Note that this is a "batch" matrix multiplication (e.g., the method torch.bmm in PyTorch [32]), that is $\mathbf{C} = \mathbf{AB} \in \mathbb{R}^{b \times m \times n}$, where $\mathbf{A} \in \mathbb{R}^{b \times m \times p}$, and $\mathbf{B} \in \mathbb{R}^{b \times p \times n}$. Similarly, the transpose operation is $\mathbf{A}^\top \in \mathbb{R}^{b \times p \times m}$.

modified as follows,

$$\mathbf{A}_s^{(i)}(x,y) \coloneqq \mathbf{A}_s^{(i)}(x,y) \odot \mathbf{W}_s(x,y) + \mathbf{S}_s(x,y),\ \forall i \in \mathcal{I} \cup \mathcal{T}, \tag{1}$$

where $i$ is the index of the attention slice corresponding to a prompt word, and $\odot$ denotes the Hadamard (element-wise) product that scales the $x, y$ element of the cross-attention map $\mathbf{A}_s$ by the corresponding weight in $\mathbf{W}_s(\cdot)$, where $\mathbf{W}_s(\cdot)$ and $\mathbf{S}_s(\cdot)$ are,

$$\mathbf{S}_s(x,y) = \begin{cases} c_s\, g(x,y), & (x,y) \in \mathcal{B} \\ 0, & \text{otherwise}, \end{cases},\quad \mathbf{W}_s(x,y) = \begin{cases} c_w, & (x,y) \in \mathcal{B}' \\ 1, & \text{otherwise} \end{cases}, \tag{2}$$

and $x, y$ are are the spatial location indices of the attention map and $\mathcal{B}'$ is the complement of $\mathcal{B}$. $\mathbf{S}_s(\mathcal{B})$ uses a *soft* window function that "injects" attention inside $\mathcal{B}$, as illustrated in the gray box in Fig. 3 with $c_s > 0$. This is implemented as a Gaussian function $g(\cdot)$ of size $\sigma_x = b_w/2, \sigma_y = b_h/2$, where $b_w = \text{ceil}((b_{\text{right}} - b_{\text{left}}) \times w), b_h = \text{ceil}((b_{\text{top}} - b_{\text{bottom}}) \times h)$ are the width and the height of $\mathcal{B}$. In contrast, $\mathbf{W}_s(\cdot)$ with $c_w \leq 1$ attenuates the attention outside $\mathcal{B}$. The bbox $\mathcal{B}$ is extended across the entire video sequence through linear interpolation of the keyframes. For example, $\mathcal{B}_f = (1-a) \times \mathcal{B}_b + a \times \mathcal{B}_e$, where $a = f/N_F$, and $\mathcal{B}_b$, $\mathcal{B}_e$ denote the bboxes for the beginning and end keyframes of the time interval.

### 3.3 Temporal Cross-Frame Attention Guidance

To capture the temporal correlation in the video clip during training, a prevalent approach involves reshaping the latent tensor. This involves shifting the spatial information to the first dimension, a technique employed in VideoLDM [5]. The reshaping is done before passing the hidden activation into the temporal layers, allowing the model to learn about the "correlation" of spatial components through the convolutional layers. As shown in Fig. 3 (right), the temporal attention map is obtained by $\mathbf{A}_m = \text{Softmax}(\mathbf{Q}_m\mathbf{K}_m^T/\sqrt{d}) \in \mathbb{R}^{d_h \times N_F \times N_F}$, where $d_h$ is the spatial dimensions of this tensor, $\mathbf{Q}_m \in \mathbb{R}^{d_h \times N_F \times d}$, and $\mathbf{K}_m \in \mathbb{R}^{d_h \times N_F \times d}$.

What is different from the spatial counterpart is that now $\mathbf{A}_m$ learns about the relation between the correlated components across all frames. For instance, $\mathbf{A}_m^{(x,y,i,j)}$ denotes the correlation at location $(x, y)$ between frame $i$ and frame $j$. We denote such tensors as $\mathbf{A}_m^{(i,j)}(x,y)$ to keep the notation consistent. As seen in Fig. 2 (right), the background attention is higher when the cross frame attention (i.e., TA-Cross, when $i \neq j$) compares frames that are temporally far from each other, and the foreground attention is higher when the frames are temporally closer in the video sequence. The self frame attention (TA-Self, when $i = j$) generally aligns spatially with the prompt cross attention (PA-Cross), as seen in Fig. 2.

To achieve this pattern of activations under user control we design an approach similar to Eq. 1 but considering the normalized video timeline distance $d = \frac{|i-j|}{N_F}, i, j \in \{1, ..., N_F\}$. The temporal injection function is defined as,

$$\mathbf{S}_m(x,y) = \begin{cases} (1-d)\ g(x,y) - d\ g(x,y), & (x,y) \in \mathcal{B}, \\ 0, & \text{otherwise}. \end{cases}$$

Here the normalized video temporal distance $d$ determines the level of the weight injection as a triangular window in time. Values $d \approx 0$ increase the activation inside the bbox. In contrast, when $d \approx 1$, the activation inside the box is *reduced*, approximating the temporal "anti-correlation" effect seen in Fig. 2. The editing by $\mathbf{S}_m(\cdot)$ is performed during the initial $N_M$ steps of the denoising process. Similarly to Eq. 1, the temporal cross-frame attention map editing is defined as,

$$\mathbf{A}_m^{(i,j)}(x,y) \coloneqq \mathbf{A}_m^{(i,j)}(x,y) \odot \mathbf{W}_m(x,y) + \mathbf{S}_m(x,y), \tag{3}$$

where $\mathbf{W}_m(\cdot)$ is defined similarly to $\mathbf{W}_s(\cdot)$.

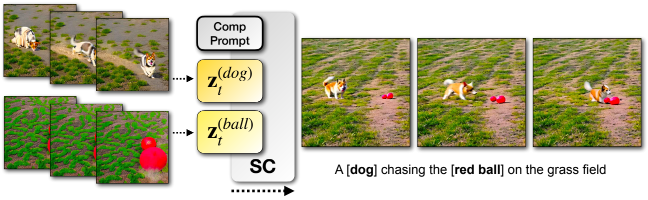

Fig. 4. **Scene Compositing.** Given the set of latents generated from our system using a single bbox denoted as $\mathbf{z}_t^{(\text{ball})}$ and $\mathbf{z}_t^{(\text{dog})}$ for the case of prompts related to ball and dog, the scene compositor (SC) produces a synthesis of multiple subjects with the complete prompt and the single subject latents. We refer reader to our supplementary video to view the implemented speed control of the dog.

### 3.4 Scene compositing

The problem space becomes more complicated for video synthesis with more than one moving subject. Although the parameters $c_s, c_w$ in Eq. 2 are specific to a particular subject, they indirectly affect the entire scene through the global denoising. Thus, the choice of these parameters for different subjects might interact and require a parameter search in the number of subjects to find the best synthesis.

Considering the reason above, we follow work such as [3, 25] that combines multiple subjects, each with their own prompt, during the latent denoising. The latents $\mathbf{z}_t^{(r)}$ for the $r$-th subject are then composited into an overall image latent $\mathbf{z}_t$ under the control of a "composed" prompt, as illustrated in Fig. 4 and formulated as,

$$\mathbf{z}_t(x,y) \coloneqq \frac{1}{R}\sum_{r=0}^{N_R}\Big(w\,\mathbf{z}_t(x,y) + (1-w)\,\mathbf{z}_t^{(r)}(x,y)\Big), \tag{4}$$

where $\forall t \in \{T, ..., T-N_C\}$, $(x, y) \in \mathcal{B}_r$, and $N_C$ is the number of editing steps at the beginning of denoising. The weight $w \in [0, 1]$ determines the weight of linear interpolation between the specific subject latent $\mathbf{z}_t^{(r)}$ and the composed latent $\mathbf{z}_t$. It is formulated by considering the ratio of the current denoising timestep between $N_C$ and $T$, such that $w = 1 - \big(N_C - (T-t)\big)/N_C$. At the beginning of the denoising process (so at $t = T$), the compositing fully prioritizes the subject latent $\mathbf{z}_t^{(r)}$ in each local region in the associated bbox $\mathcal{B}_r$. As $t$ decreases, $w$ gradually increases, giving higher priority to composed latent $\mathbf{z}_t$. This process concludes when $t = T - N_C$, resulting in $w = 1$, thus the remaining denoising steps are global and ignore the per-subject latents.

## 4 EXPERIMENTS

Here we briefly present experiments and quantitative evaluations. Please see our supplementary materials and the project video for full experiments, including implementation details, limitations, ablations, and finer details. Except for Fig. 6, the figures show an evenly spaced temporal sampling of frames from the videos.

### 4.1 Main results

Fig. 5 and Fig. 6 show our main result on trajectory control of a single subject. We use the same prompts to compare *TrailBlazer* to MotionCtrl [51], Videocomposer [50], and Peekaboo [18], which are current state-of-the-art approachs for bbox guidance of video generation. For VideoComposer, their handcrafted motion guidance was not available in the paper's repository at the time of writing.[2] Instead, we use TrailBlazer's output as the input for VideoComposer, with the synthesized video serving as motion guidance and the first frame as the initial input frame. For MotionCtrl, we calculated the mean of the bounding box to serve as the trajectory input.

Fig. 5 illustrates motion generated from linearly interpolated bboxes starting at the left of the image and moving to the right. The results from TrailBlazer demonstrate anatomically plausible motion of the subject and a more accurate fitting of the subject within the bbox. Further, all subjects (e.g., cat, bee, astronaut, and clown fish) face in the direction that they move. Although the synthesized subject's motion generally follows the bbox in Peekaboo, it does not fit the bbox well in our experience. Occasionally, artifacts may emerge such as a rectangular object following the astronaut. Moreover, our synthesized background exhibits better visual quality. The background often appears plain, blurry, or lacks detail behind the subject in Peekaboo.

Next, as depicted in Fig. 6, TrailBlazer excels in cases where the synthesized subject's location, size, and speed are strongly varied. For example, the tiger walks along the road in a perspective view induced by the changing bbox size. The whale gracefully descends into the ocean during the latter part of its jumping motion. The horse accurately follows a zigzag path, simulating a galloping motion. Remarkably, the dog seamlessly follows a large number of keyframes (8 keyframes) within a 24-frame video clip, covering the distance from one boundary to the opposite in approximately 2 frames. The clownfish fits into a tiny bounding box. These successes are generally not evident in the other approaches.

*TrailBlazer* also provides innovative functionality in T2V subject editing as shown in Fig. 8. *Subject morphing* (Fig. 8, left) involves smoothly interpolating the subject identity. Related concepts have earlier been shown for image generation in MagicMix [23] with, for example, the "corgi coffee machine". Morphing [4] has long been used in the entertainment and VFX industries, for example in the *Hulk* movies [39]. Conversely, when the prompt is held fixed *TrailBlazer* preserves the identity, while changing bbox locations and aspect ratios induce realistic perspective effects, as seen (Fig. 8, Right) where the simulated fish swims respects the bbox size by swimming toward and away from the virtual camera.

Multi-subject synthesis is generally challenging, particularly when the number of objects exceeds two. We delve into this issue in the supplementary materials. In Fig. 9, we present experiments with two subjects, a cat and a dog. The synthesis of the cat and dog in isolation is depicted in the first and second column, serving as a baseline for comparison. We also show eight results combining different environments ("... on the moon", "... in the park") after the composed prompt ("A [**white cat**] and a [**yellow dog**] running..."). This experiment demonstrates the flexibility of TrailBlazer in synthesizing subjects under varied environmental conditions. Notably, the interactions between the background and subjects appear plausible, as seen in the cast shadows, and the reflections and splashes in the swimming pool case. The results also show some artifacts such as extra limbs that are inherited from the underlying model.

| Method | FVD(↓) | FID(↓) | mIoU(↑) | CD(↓) | CS(↑) |
|---|---|---|---|---|---|
| MotionCtrl | **1586.45** | **150.93** | 0.15 | 13.68% | **32.28** |
| Videocomposer | 1691.86 | 151.57 | 0.09 | 13.69% | 31.21 |
| Peekaboo | 1788.06 | 170.33 | 0.23 | 8.58% | 31.55 |
| TrailBlazer | 1596.98 | 172.72 | **0.26** | **7.27%** | 30.60 |

Table 1. Quantitative results for static bbox.

### 4.2 Quantitative evaluation

The field of controllability in generative video lacks standardized quantitative metrics and visual evaluation is essential – please refer to our video. In particular there are no metrics that reflect depth and orientation, e.g. the ability to make an object appear to move in perspective toward the virtual camera and then turn away. Although mean intersection over union (mIoU) is an obvious measure for bounding box guidance (and our performance exceeds that of baseline methods), it is not clear if this is the *ideal* measure for our intended high-level soft guidance.

Despite these reservations, it is helpful to provide some form of quantitative evaluation. Here we follow the methodology in [5, 16, 18]. We report Fréchet Inception Distance [12] (FID), Fréchet Video Distance (FVD), mIoU, centroid distance (CD) [18], and CLIP similarity (CS) metrics on all image frames of 400 randomly selected videos from the AnimalKingdom dataset [28]. As described in the supplementary materials, we evaluate methods using the prompt set published in [18]. The mIoU evaluation utilizes the OWL-ViT-large open-vocabulary object detector [26] to obtain the bbox of the synthesized subject.

For a fair quantitative evaluation, we generated baseline results using Peekaboo, Videocomposer, and MotionCtrl without additional conditioning input (e.g., sketch, depth map). The comparisons used 24-frame video sequences. We conducted two experiments with random keyframing for our work: *Static bbox*, and *Dynamic bbox*.

The bboxes in the *Static bbox* experiments are constant across all keyframes, where the top left corner is randomly generated in the second quadrant, and the width and height is randomly selected between 25% to 50% of the image resolution. This experiments mainly evaluate the method without considering the bbox motion. The result is summarized in Table. 1. As observed, our performance is not far from other approaches across all metrics, while our mIoU, and CD are superior to that of other approaches.

[2] https://github.com/ali-vilab/videocomposer/issues/24

| Method | FVD(↓) | FID(↓) | mIoU(↑) | CD(↓) | CS(↑) |
|---|---|---|---|---|---|
| MotionCtrl | 1938.03 | 154.62 | 0.18 | 12.28% | **32.12** |
| Videocomposer | 2117.43 | **134.89** | 0.17 | 10.68% | 31.40 |
| Peekaboo | **1613.27** | 167.09 | 0.25 | 9.18% | 31.32 |
| TrailBlazer | 1679.94 | 147.29 | **0.36** | **6.83**% | 30.32 |

Table 2. Quantitative results for dynamic bbox.

Table. 2 presents the results of the *Dynamic bbox* experiments to assess the effectiveness of movement control in each method. The bboxes were generated by randomly specifying between two and six keyframes alternately located on the left and right, causing the subject to run from one side to the other and back as shown in Fig. 6. The location and size of all bboxes are randomly selected for each video, with the height and width of each bbox chosen between 10% and 50% of the image size.

In Table. 2, the notable improvement is our mIoU score compared to Peekaboo and other methods. This can be attributed to TrailBlazer's proficiency in following complex bbox movement including generating perspective/depth effects with dynamically changing bboxes (Fig. 6, 8). For instance, the running dog (Fig. 6) is a clear example where the dog generated from other methods failed to follow fast bbox motion. Additionally, TrailBlazer has the best mIoU/CD, while its FVD is second-best. This discrepancy might be explained by the nature of the AnimalKingdom [28] dataset, which mostly contains actions with relatively limited translation (walking, eating, grooming, etc.). Running motion such as the dog in Fig. 6 is generally absent in their dataset, possibly contributing to the lower FVD score in our case. Comparing to Peekaboo, our better FID score suggests that the individual frame quality in our video clip is better.

In summary, the objective scores in Tables 1, 2 do not give a clear ordering of methods. However, recall that our goal is *controlling movement*. TrailBlazer achieves this, showing significantly better mIoU/CD scores. Equally important, TrailBlazer shows improved subjective movement, with moving objects facing in plausible directions and having realistic motion (please refer to our video). Lastly, note that Videocomposer and MotionCtrl are training-based methods, while Peekaboo and TrailBlazer are zero-shot methods.

## 5 LIMITATIONS

Our method shares and inherits common failure cases of the underlying diffusion model. Notably, at the time of writing, models based on CLIP and Stable Diffusion sometimes generate deformed objects and struggle to generate multiple objects and correctly assign attributes (e.g. color) to objects. We show some failures in Fig. 7 (Right-Bottom). For instance, we requested a red jeep driving on the road but the synthesis shows it sinking into a mud road. The panda example shows the camera moving instead of the panda itself. The red car has implausible deformation, and Darth Vader's light saber turns into a surf board. The length of the resulting video clips is restricted to that produced by the pre-trained model, for instance, the 24 images in the case of ZeroScope. This is not a crucial limitation, as movies are commonly (with some exceptions!) composed of short "shots" of several seconds each. The bbox guides object placement without precisely constraining it. This is an advantage as well, however, since otherwise the user would have to specify the exact x-y aspect ratio for objects, a complicated task for non-artists.

## 6 ABLATIONS

We conducted ablation experiments on the number of trailing attention maps and the number of temporal steps.

**Trailing attention maps.** Fig. 7 (Left) shows an ablation varying the number of trailing attention maps used in our spatial cross attention process, from the top row without trailing attention maps ($|\mathcal{T}| = 0$) to the bottom row with $|\mathcal{T}| = 30$ trailing maps. The guiding bbox moving from left to right is annotated in green. It is observed that with no trailing maps the astronaut remains static at the image center. In contrast, the synthesis with a large number of trailing attentions can lead to failed results such as a flag rather than the intended astronaut. A good number of edited trailing attention maps is between $|\mathcal{T}| = 10$ and $|\mathcal{T}| = 20$.

**Temporal attention editing.** We further show an ablation test in Fig. 7 (Top-Right) with a varied number of temporal attention editing steps. We take the case of the astronaut experiment with $|\mathcal{T}| = 10$ mentioned above, and set $N_M = 0$ (no editing steps), and $N_M = 10$. The result with $N_M = 0$ shows a red blob moving from left to right. The value $N_M = 10$ gives a satisfactory result for the astronaut, but the background along the bbox path is missing. From these results we see that a reasonable balance between spatial and the temporal attention editing must be maintained, while extreme values of either produce poor results. An intermediate value such as $N_M = 5$ used in most of our experiments produces the desired result of an astronaut moving over a moon background.

## 7 CONCLUSION

We have addressed the problem of controlling the motion of objects in a diffusion-based text-to-video model. Specifically, we introduced a combined spatial and temporal attention guidance algorithm, *TrailBlazer*, operating in the pre-trained ZeroScope model. The spatial location of a subject can be guided through simple bounding boxes. Bounding boxes and prompts can be animated via keyframes, enabling users to alter the trajectory and coarse behavior of the subject along the timeline. The resulting subject(s) fit seamlessly in the specified environment, providing a viable approach to video storytelling by casual users. Our approach requires no model finetuning, training, or online optimization, ensuring computational efficiency and a good user experience. Lastly, the results are natural, with desirable emergent effects such as perspective, motion with the correct object orientation, and the interactions (shadows, dust, splashes) between object and environment arising automatically. While our visual results inherit the limitations of the underlying open source T2V model (limited resolution, multiple limbs, etc.), our combined spatial and temporal attention guidance algorithm may generalize to future models that use standard self- and cross-attention modules.

## 8 ACKNOWLEDGEMENT

We thank Ming-Yu Liu for his feedback.

## REFERENCES

[1] Daniel Arijon. 1976. *Grammar of the Film Language*. Focal Press.

[2] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. 2022. eDiff-I: Text-to-Image Diffusion Models with an Ensemble of Expert Denoisers. *CoRR* abs/2211.01324 (2022). https://doi.org/10.48550/arXiv.2211.01324 arXiv:2211.01324

[3] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. 2023. MultiDiffusion: Fusing Diffusion Paths for Controlled Image Generation. *CoRR* abs/2302.08113 (2023). arXiv:2302.08113

[4] Thaddeus Beier and Shawn Neely. 1998. *Feature-based image metamorphosis*. Association for Computing Machinery, New York, NY, USA, 373–380. https://doi.org/10.1145/280811.281029

[5] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. 2023. Align your Latents: High-Resolution Video Synthesis with Latent Diffusion Models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[6] cerspense. 2023. zeroscope-v2-576w. https://huggingface.co/cerspense/zeroscope-v2-576w Accessed: 2023-10-01.

[7] Weifeng Chen, Jie Wu, Pan Xie, Hefeng Wu, Jiashi Li, Xin Xia, Xuefeng Xiao, and Liang Lin. 2023. Control-A-Video: Controllable Text-to-Video Generation with Diffusion Models. arXiv:2305.13840 [cs.CV]

[8] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. 2023. Structure and Content-Guided Video Synthesis with Diffusion Models. *ArXiv* abs/2302.03011 (2023). https://api.semanticscholar.org/CorpusID:256615582

[9] Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. 2023. Preserve Your Own Correlation: A Noise Prior for Video Diffusion Models. *Proceedings of the IEEE/CVF International Conference on Computer Vision 2023* (2023).

[10] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. 2022. Flexible Diffusion Modeling of Long Videos. arXiv:2205.11495 [cs.CV]

[11] 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).

[12] 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. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf

[13] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey A. Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. 2022. Imagen Video: High Definition Video Generation with Diffusion Models. *ArXiv* abs/2210.02303 (2022). https://api.semanticscholar.org/CorpusID:252715883

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems* 33 (2020).

[15] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. 2022. Video Diffusion Models. arXiv:2204.03458 [cs.CV]

[16] Zhihao Hu and Dong Xu. 2023. VideoControlNet: A Motion-Guided Video-to-Video Translation Framework by Using Diffusion Model with ControlNet. arXiv:2307.14073 [cs.CV]

[17] Tobias Höppe, Arash Mehrjou, Stefan Bauer, Didrik Nielsen, and Andrea Dittadi. 2022. Diffusion Models for Video Prediction and Infilling. https://doi.org/10.48550/ARXIV.2206.07696

[18] Yash Jain, Anshul Nasery, Vibhav Vineet, and Harkirat Behl. 2023. PEEKABOO: Interactive Video Generation via Masked-Diffusion. arXiv:2312.07509 [cs.CV]

[19] Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. 2023. Text2Video-Zero: Text-to-Image Diffusion Models are Zero-Shot Video Generators. *arXiv preprint arXiv:2303.13439* (2023).

[20] Pengxiang Li, Zhili Liu, Kai Chen, Lanqing Hong, Yunzhi Zhuge, Dit-Yan Yeung, Huchuan Lu, and Xu Jia. 2023. Trackdiffusion: Multi-object tracking data generation via diffusion models. *arXiv preprint arXiv:2312.00651* (2023).

[21] 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. *CoRR* abs/2301.07093 (2023). arXiv:2301.07093

[22] Long Lian, Baifeng Shi, Adam Yala, Trevor Darrell, and Boyi Li. 2023. LLM-grounded Video Diffusion Models. arXiv:2309.17444 [cs.CV]

[23] Jun Hao Liew, Hanshu Yan, Daquan Zhou, and Jiashi Feng. 2022. MagicMix: Semantic Mixing with Diffusion Models. *CoRR* abs/2210.16056 (2022).

[24] Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jingren Zhou, and Tieniu Tan. 2023. VideoFusion: Decomposed Diffusion Models for High-Quality Video Generation. arXiv:2303.08320 [cs.CV]

[25] Wan-Duo Kurt Ma, J. P. Lewis, Avisek Lahiri, Thomas Leung, and W. Bastiaan Kleijn. 2023. Directed Diffusion: Direct Control of Object Placement through Attention Guidance. arXiv:2302.13153 [cs.CV]

[26] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby. 2022. Simple Open-Vocabulary Object Detection. In *Computer Vision – ECCV 2022*, Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (Eds.). Springer Nature Switzerland, Cham, 728–755.

[27] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, 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).

[28] 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.

[29] Alex Nichol and Prafulla Dhariwal. 2021. Improved Denoising Diffusion Probabilistic Models. arXiv:2102.09672 [cs.LG]

[30] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. 2022. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. In *ICML*.

[31] OpenAI. 2024. Sora: Creating video from text. https://openai.com/sora

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv:1912.01703 [cs.LG]

[33] Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. 2023. FateZero: Fusing Attentions for Zero-shot Text-based Video Editing. *arXiv:2303.09535* (2023).

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proc. ICML*.

[35] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical Text-Conditional Image Generation with CLIP Latents. *CoRR* abs/2204.06125 (2022). arXiv:2204.06125

[36] 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 (CVPR)*. 10684–10695.

[37] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *CoRR* abs/2205.11487 (2022).

[38] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, Seyedeh Sara Mahdavi, Raphael Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *ArXiv* abs/2205.11487 (2022). https://api.semanticscholar.org/CorpusID:248986576

[39] Mike Seymour. 2023. She-Hulk: Wētā's Shade of Green. https://www.fxguide.com/fxfeatured/she-hulk-wetas-shade-of-green/ Accessed: 2022-10-27.

[40] Mike Seymour. 2024. Actually Using Sora. https://www.fxguide.com/fxfeatured/actually-using-sora/

[41] Yujun Shi, Chuhui Xue, Jiachun Pan, Wenqing Zhang, Vincent YF Tan, and Song Bai. 2023. DragDiffusion: Harnessing Diffusion Models for Interactive Point-based Image Editing. *arXiv preprint arXiv:2306.14435* (2023).

[42] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*.

[43] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021. Denoising Diffusion Implicit Models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

[44] Yang Song and Stefano Ermon. 2019. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, Vol. 32.

[45] Wei Sun and Tianfu Wu. 2022. Learning Layout and Style Reconfigurable GANs for Controllable Image Synthesis. *TPAMI* 44 (2022), 5070–5087.

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30.

Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[47] Vikram Voleti, Alexia Jolicoeur-Martineau, and Christopher Pal. 2022. MCVD: Masked Conditional Video Diffusion for Prediction, Generation, and Interpolation. In *(NeurIPS) Advances in Neural Information Processing Systems*. https://arxiv.org/abs/2205.09853

[48] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. 2023. ModelScope Text-to-Video Technical Report. arXiv:2308.06571 [cs.CV]

[49] Jiawei Wang, Yuchen Zhang, Jiaxin Zou, Yan Zeng, Guoqiang Wei, Liping Yuan, and Hang Li. 2024. Boximator: Generating Rich and Controllable Motions for Video Synthesis. arXiv:2402.01566 [cs.CV]

[50] 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. In *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., 7594–7611. https://proceedings.neurips.cc/paper_files/paper/2023/file/180f6184a3458fa19c28c5483bc61877-Paper-Conference.pdf

[51] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. 2024. MotionCtrl: A Unified and Flexible Motion Controller for Video Generation. In *ACM SIGGRAPH 2024 Conference Papers* (Denver, CO, USA) *(SIGGRAPH '24)*. Association for Computing Machinery, New York, NY, USA, Article 114, 11 pages. https://doi.org/10.1145/3641519.3657518

[52] Lilian Weng. 2021. What are diffusion models? https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

[53] Wikipedia contributors. 2023. keyframe. https://en.wikipedia.org/wiki/Key-frame Accessed: 2023-10-01.

[54] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. 2023. 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.

[55] Jinheng Xie, Yuexiang Li, Yawen Huang, Haozhe Liu, Wentian Zhang, Yefeng Zheng, and Mike Zheng Shou. 2023. BoxDiff: Text-to-Image Synthesis with Training-Free Box-Constrained Diffusion. *CoRR* abs/2307.10816 (2023). arXiv:2307.10816

[56] Hanshu Yan, Jun Hao Liew, Long Mai, Shanchuan Lin, and Jiashi Feng. 2023. MagicProp: Diffusion-based Video Editing via Motion-aware Appearance Propagation. arXiv:2309.00908 [cs.CV]

[57] Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. 2022. Diffusion Probabilistic Modeling for Video Generation. arXiv:2203.09481 [cs.CV]

[58] Shiyuan Yang, Liang Hou, Haibin Huang, Chongyang Ma, Pengfei Wan, Di Zhang, Xiaodong Chen, and Jing Liao. 2024. Direct-a-Video: Customized Video Generation with User-Directed Camera Movement and Object Motion. arXiv:2402.03162 [cs.CV]

[59] Zuopeng Yang, Daqing Liu, Chaoyue Wang, J. Yang, and Dacheng Tao. 2022. Modeling Image Composition for Complex Scene Generation. *CVPR* (2022), 7754–7763.

[60] 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. https://www.microsoft.com/en-us/research/publication/dragnuwa-fine-grained-control-in-video-generation-by-integrating-text-image-and-trajectory/

[61] Lvmin Zhang and Maneesh Agrawala. 2023. Adding Conditional Control to Text-to-Image Diffusion Models. arXiv:2302.05543 [cs.CV]

[62] Bo Zhao, Weidong Yin, Lili Meng, and Leonid Sigal. 2020. Layout2image: Image Generation from Layout. *Int. J. Comput. Vis.* 128, 10 (2020), 2418–2435. https://doi.org/10.1007/s11263-020-01300-7

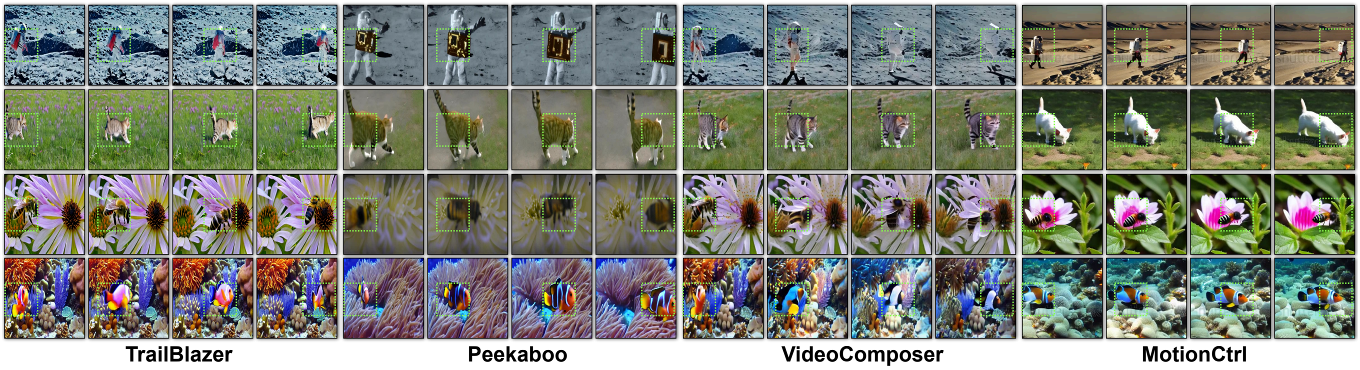

Fig. 5. **Baseline Comparison: Rigid bbox moving from left to right.** The same prompt is used across each method (from top row): “An [**astronaut**] walking on the moon”; “A [**cat**] walking on the grass field.”; “A macro video of a [**bee**] pollinating a flower”; “A [**clown fish**] swimming in a coral reef”.

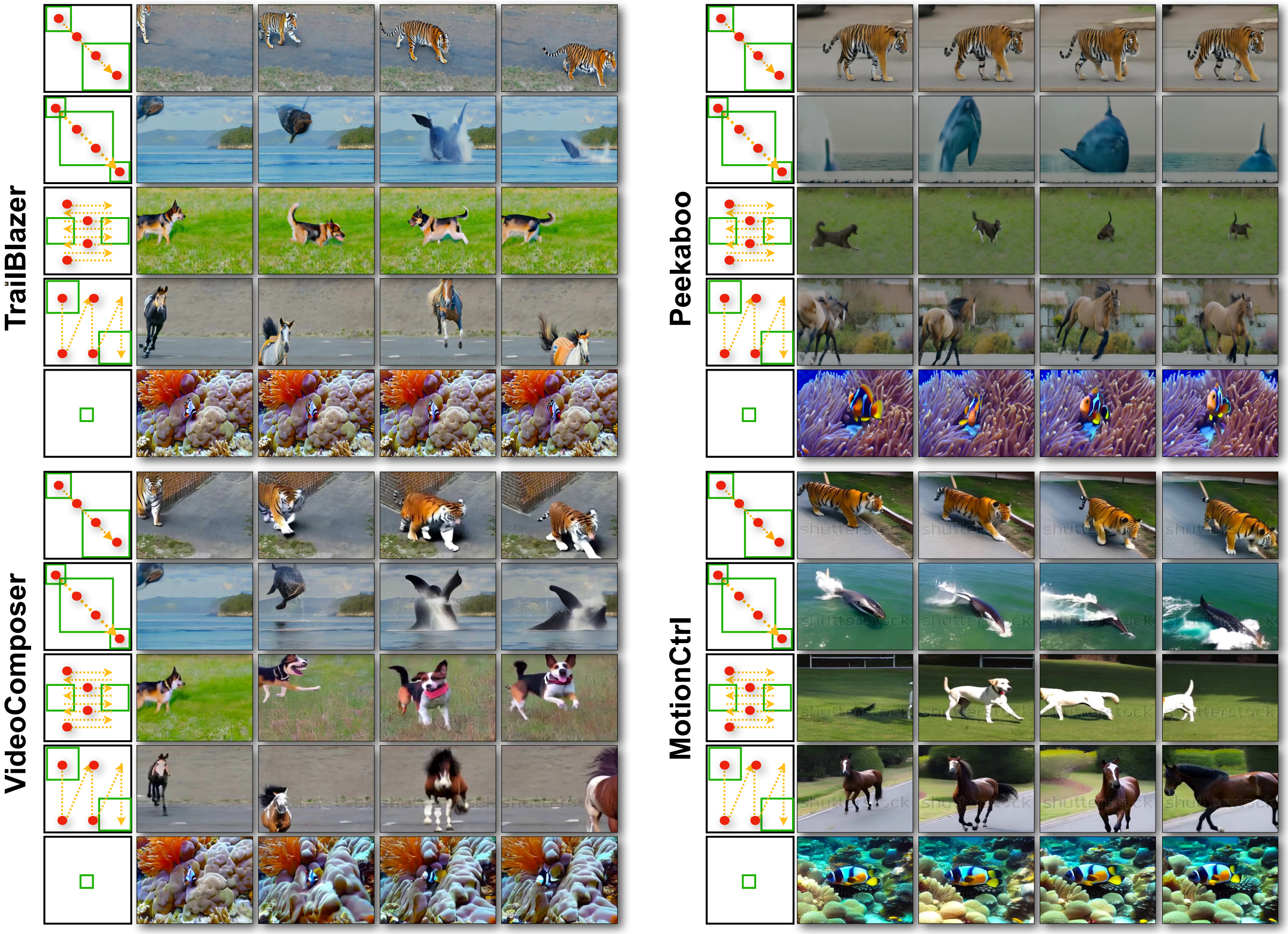

Fig. 6. **Baseline Comparison: Dynamic moving bbox.** The first column of each block schematically illustrates the bbox keyframes, where the green bbox is guided by the almond-colored motion vector. For the synthesized sequences with complex motion from 1st to 4th row, the frames shown in the figure are denoted by the red dots along the trajectory in the first column. The last sequence challenges the methods to fit the subject in an extremely static small bbox. Prompt used: “The [**tiger**] walking on the street; A photo realistic [**whale**] jumping out of water while smoking a cigar”; “A [**dog**] is running on the grass”; “A [**horse**] galloping fast on a street”; “A [**clownfish**] swimming in a coral reef.”

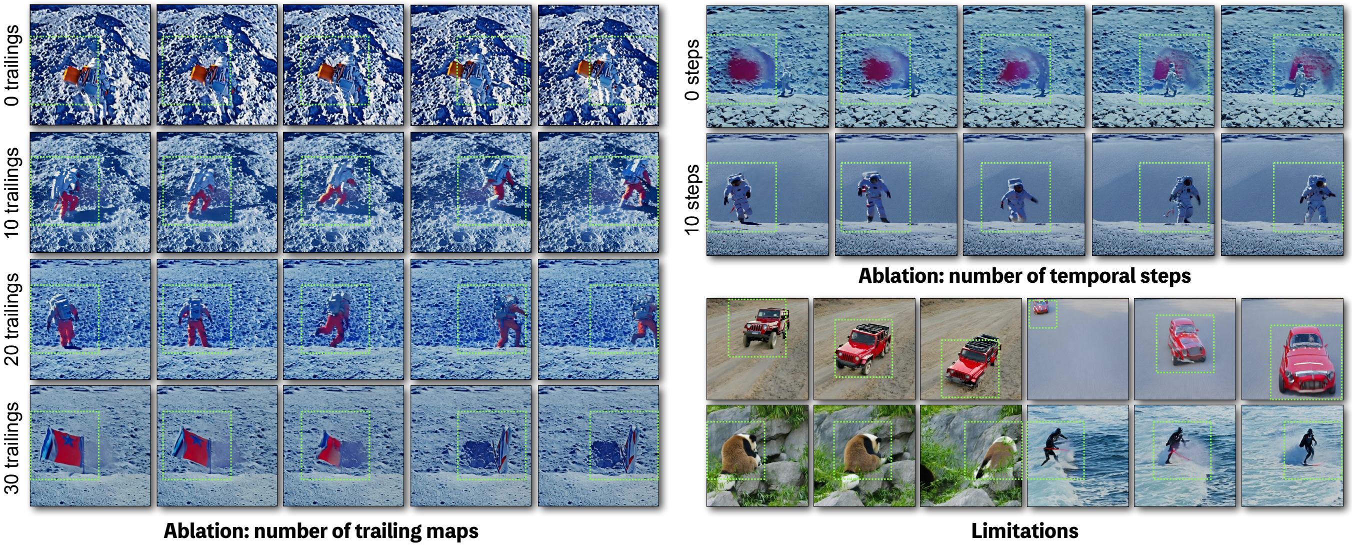

Fig. 7. **Ablations and Limitations.** (Left) The rows from top to bottom show the video synthesis with 0 (no trailing maps), 10, 20, and 30 trailing maps. The number of temporal edit steps is five in all cases. (Top-Right) The first and second rows show the result produced by no temporal attention editing, and 10 editing steps, respectively. The number of trailing maps is 10 for both cases. (Bottom-Right) Failure cases that we discussed. Prompts used for this figure: "An [**astronaut**] walking on the moon", "A [**red jeep**] driving on the road", "A [**red car**] driving on the highway", "a [**panda**] eating bamboo", and "[**Darth Vader**] surfing in waves"

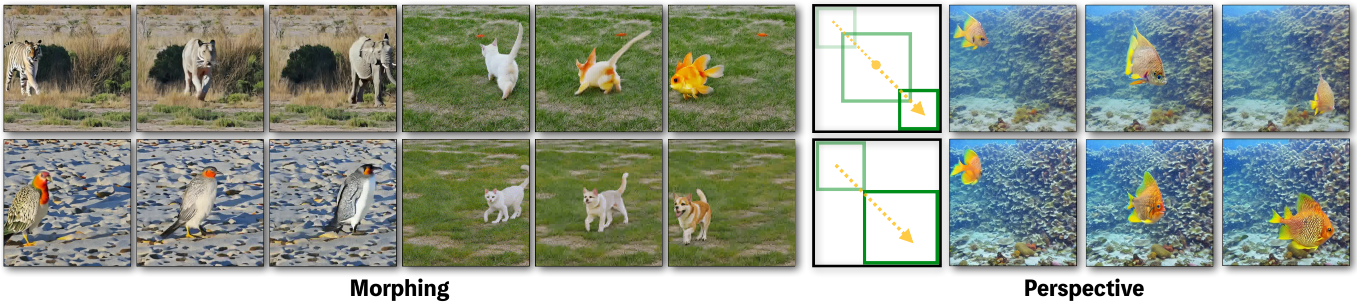

Fig. 8. **Advanced functionality.** (Left: Morphing) Each experiment shows a subject linearly interpolated between the first and the last frame. Prompts used starting from the first row: "A [**tiger** → **elephant**] walking in the wild park.", "A [**parrot** → **king penguin**] walking on the beach", "A [**cat** → **dog**] walking on the grass", and "A [**cat** → **golden fish**] walking on the grass". (Right: Perspective) Size changes in the bboxes result in movement towards/away from the virtual camera. Prompt: "A [**fish**] swimming in the sea".

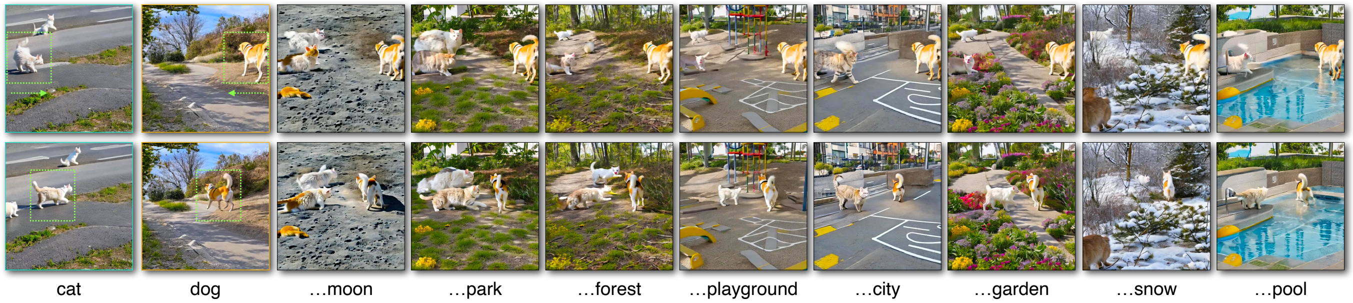

Fig. 9. **Scene compositing.** The two images in each column are the first and end frame of the synthesized video. The first two columns on the left with annotated bboxes show the video synthesis of the two subjects: "cat" and the "dog" guided by the green bbox. Each subsequent column shows the compositing result of a varied environment appended to the prompt.