TRAILBLAZER: TRAJECTORY CONTROL FOR DIFFUSION-BASED VIDEO GENERATION

A PREPRINT

Wan-Duo Kurt Ma
Victoria University of Wellington
mawand@ecs.vuw.ac.nz

J. P. Lewis
NVIDIA Research
jpl@nvidia.com

W. Bastiaan Kleijn
Victoria University of Wellington
bastiaan.kleijn@vuw.ac.nz

January 3, 2024

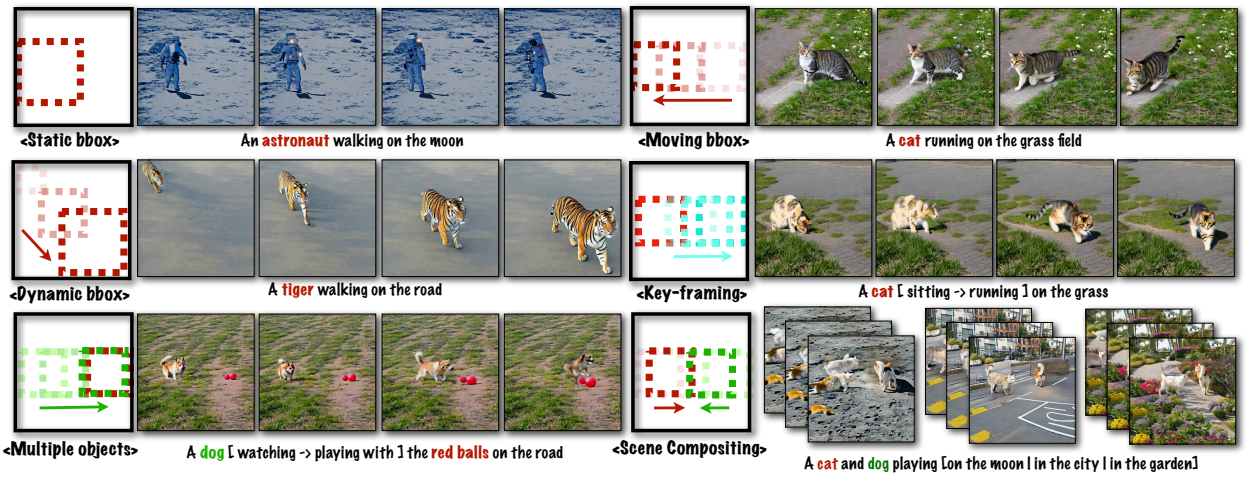

Figure 1: *TrailBlazer* extends a pre-trained video diffusion model to introduce trajectory control over one or multiple subjects. Its primary contribution lies in the ability to animate the synthesized subject using a bounding box *(bbox)*, whether it remains static (Top-left) or dynamic in terms of location (Top-right), bbox size (Middle-left), and varied movement speed (Middle-right, the cat sitting in the early half of video in red bbox, and then moving with cyan bbox), achieved through keyframing. 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).

ABSTRACT

Within recent approaches to text-to-video (T2V) generation, achieving controllability in the synthesized video is often a challenge. Typically, this issue is addressed by providing low-level per-frame guidance in the form of edge maps, depth maps, or an existing video to be altered. However, the process of obtaining such guidance can be labor-intensive. This paper focuses on enhancing controllability in video synthesis by employing straightforward bounding boxes to guide the subject in various ways, all without the need for neural network training, finetuning, optimization at inference time, or the use of pre-existing videos. Our algorithm, *TrailBlazer*, is constructed upon a pre-trained (T2V) model, and easy to implement.[1] The subject is directed by a bounding box through the proposed spatial and temporal attention map editing. Moreover, we introduce the concept of keyframing, allowing the subject trajectory and overall appearance to be guided by *both* a moving bounding box and corresponding prompts, without the need to provide a detailed mask. The method is efficient, with negligible additional computation relative to the underlying pre-trained model. Despite the simplicity of the bounding box guidance, the resulting motion is surprisingly natural, with emergent effects including perspective and movement toward the virtual camera as the box size increases.

[1]Our project page: https://hohonu-vicml.github.io/Trailblazer.Page/

1 Introduction

Advancements in generative models for text-to-image (T2I) have been dramatic Ramesh et al. (2022); Saharia et al. (2022a); Rombach et al. (2022); Balaji et al. (2022). Recently, text-to-video (T2V) systems have made significant strides, enabling the automatic generation of videos based on textual prompt descriptions Ho et al. (2022a,b); Wu et al. (2023); Esser et al. (2023). One primary challenge in video synthesis lies in the extensive memory and training data required. Methods based on the pre-trained Stable Diffusion (SD) model have been proposed to address the efficiency issues in Text-to-Video (T2V) synthesis. These approaches address the problem from several perspectives including finetuning and zero-shot learning Khachatryan et al. (2023); Qi et al. (2023).

However, text prompts do not provide good control over the spatial layout and trajectories of objects in the generated video. This control is known to be required for understandable narration of a story Arijon (1976). Existing work such as Hu and Xu (2023) has approached this problem by providing low-level control signals, e.g., using Canny edge maps or tracked skeletons to guide the objects in the video using ControlNet Zhang and Agrawala (2023). These methods achieve good controllability, but they can require considerable effort to produce the control signal. For example, capturing the desired motion of an animal (e.g., a tiger) or an expensive object (a jet plane) would be quite difficult, while sketching the desired movement on a frame-by-frame basis would be tedious.

To address the needs of casual users, we introduce a high-level interface for control of object trajectories in synthesized videos. Users simply provide bounding boxes (bboxes) specifying the desired position of an object at several points in the video, together with the text prompt(s) describing the object at the corresponding times. Taking inspiration from the observation Liew et al. (2022) that object position is established early in the denoising process, and leveraging the clear spatial interpretation of spatial and temporal attention maps as illustrated in Fig. 2, we propose a general approach: Our strategy involves editing *both spatial and temporal attention maps* for a specific object during the initial denoising diffusion steps to concentrate activation at the desired object location. Our inference-time editing approach achieves this without disrupting the learned text-image association in the pre-trained model, and requires minimal code modifications. We use ZeroScope cerspense (2023) in a pre-trained fashion as the underlying model.

Our contributions are three-fold:

- **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, the low-level guidance signals (detailed masks, edge maps) used by some other approaches have two disadvantages: it is difficult for non-artists to draw these shapes, and processing existing videos to obtain these signals limits the available motion to copies of existing sources.
- **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 be similarly controlled, thereby producing perspective effects (Figs. 1,6). Finally, users can also keyframe the text prompt to influence the behavior of the subject in the synthesized video (Figs. 1,6).
- **Simplicity.** Our method operates by directly editing the spatial and temporal attention in the pre-trained denoising UNet. It requiring 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 Sohl-Dickstein et al. (2015); Song and Ermon (2019); Ho et al. (2020) or deterministic Song et al. (2021) mapping between the data space and a corresponding-dimension multivariate Gaussian. Signals are synthesized by sampling from the Gaussian and performing a sequence of denoising steps. A number of works Nichol et al. (2022); Nichol and Dhariwal (2021); Ramesh et al. (2022); Saharia et al. (2022b) have performed T2I synthesis using images conditioned on the text embedding from a model such as CLIP Radford et al. (2021). Performance is significantly improved in the Latent Diffusion Model Rombach et al. (2022) (LDM) by doing 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 good tutorials are available, e.g., Weng (2021).

Despite the success of image generation using SD, it is widely acknowledged that SD lacks controllability in synthesis. SD faces challenges in synthesizing multiple objects, often resulting in missing objects or incorrect assignment of

prompt attributes to different objects. Recently, ControlNet Zhang and Agrawala (2023) and T2I-Adapter Mou et al. (2023) used additional fine-tuning layers to train the model with various forms of image conditioning (edge maps, skeletons).

The methods of Zhao et al. (2020); Sun and Wu (2022); Yang et al. (2022b); Ma et al. (2023); Xie et al. (2023); Bar-Tal et al. (2023) have addressed the layout-to-image (L2I) issue using few-shot learning. Directed Diffusion Ma et al. (2023), BoxDiff Xie et al. (2023), and MultiDiffusion Bar-Tal et al. (2023) use coarse bounding boxes to control subject position, achieving good result by manipulating the spatial latent and text embeddings cross attention map Hertz et al. (2022).

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 requirement for a large paired text and video dataset. Ho et al. (2022b); Harvey et al. (2022); Höppe et al. (2022); Voleti et al. (2022); Yang et al. (2022a); Ge et al. (2023) show methods that build on top of image diffusion models. Some works Blattmann et al. (2023); Luo et al. (2023) also introduce 3D convolutional layers in the denoising UNet to learn temporal information. Imagen Video Ho et al. (2022a) achieves higher resolution by computing temporal and spatial super-resolution on initial low resolution videos. VideoLDM Blattmann et al. (2023) and ModelScope Luo et al. (2023) insert a temporal attention layer by reshaping the latent tensor. Text2Video-Zero Khachatryan et al. (2023) and FateZero Qi et al. (2023) investigate how the temporal coherence can be improved by cross frame attention manipulation with pre-trained T2I models. Ge et al. (2023) address the same problem by introducing temporal correlation in the diffusion noise. However, these pioneering studies generally lack position control in the video synthesis.

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 Chen et al. (2023) and MagicProp Yan et al. (2023) use depth maps with ControlNet to train a temporal-aware network. Text2Video-Zero Khachatryan et al. (2023) partially achieves controllability by initializing the latent frames conditioned on the first frame with applied linear translation. However, it does not know about the reconstructed subject, making it difficult to edit the video (e.g., the user might need to know the initial position after synthesis, and then adjust the the shifting factor). 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. We note that our method can position and animate the bounding boxes with keyframing to control the subject.

3 Method

Our approach is based on the open-source pre-trained model ZeroScope cerspense (2023). This is a fine-tuned version of ModelScope Luo et al. (2023), known for its ability to generate high-quality videos without significant temporal flickering. It is noteworthy that our approach preserves this desirable temporal coherence effect achieved in their work. Our approach does not require any training, optimization, or low-level control signals (e.g., edge, depth maps with ControlNet Zhang and Agrawala (2023)). 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: 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*.

Similar to the work in Ma et al. (2023), our method draws significant inspiration from visual inspection of cross-attention maps. Consider the final cross-attention result depicted in Fig. 2, generated from the prompt "an astronaut walking on the moon". The spatial cross attention (denoted as SAttn) associated with the prompt word "astronaut" is highlighted in the second row, showcasing the overall position of the subject. Furthermore, we visualize the attention map from the temporal module in the pre-trained model. The third row displays "self-frame" temporal attention maps TAttn-Self, which consistently align with SAttn.

The last row of Fig. 2 presents the visualization of cross-frame temporal attention maps, denoted as TAttn-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 and becomes more correlated in the background area. This observation aligns with the reconstructed video shown in the first row, where the background remains nearly static while the astronaut's motion varies frame by frame. We will consider the temporal attention in detail in Sec. 3.3.

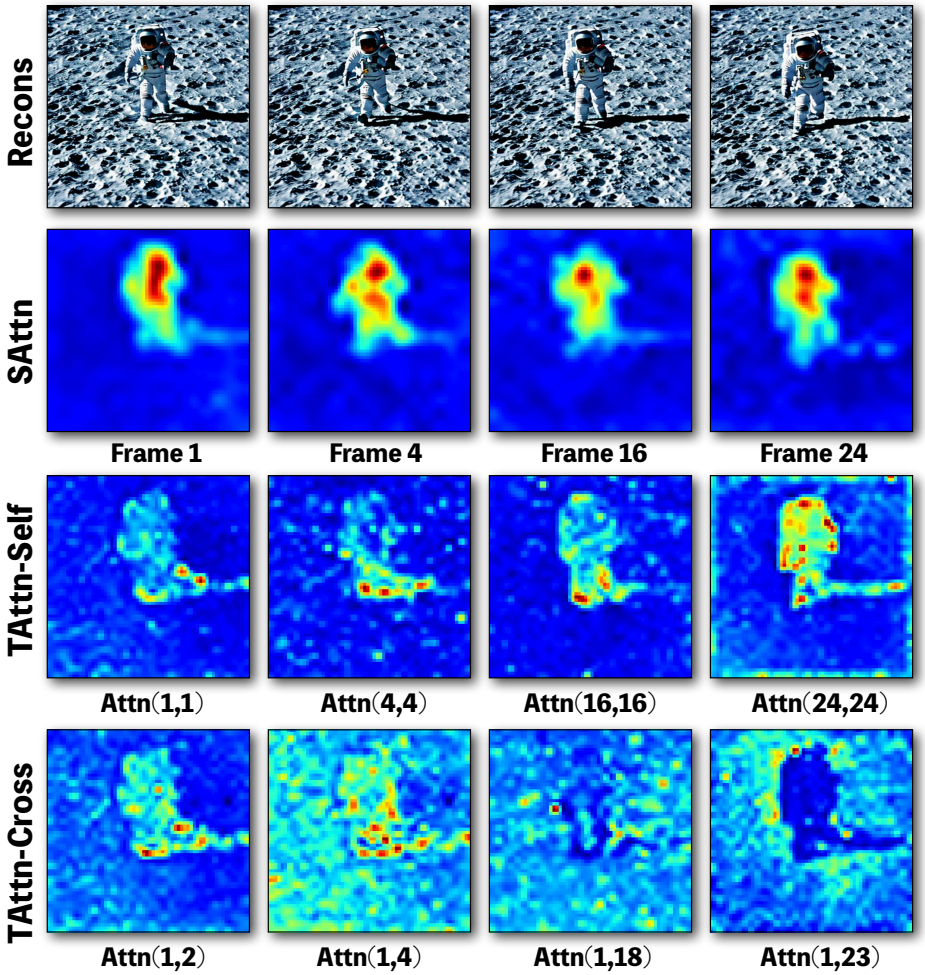

Figure 2: **Basis of our method.** We draw inspiration from inspection of the spatial (SAttn) and temporal (TAttn) cross-attention maps viewed with self-frame attention (Self) and cross-frame attention (Cross).Thus TAttn-Self and TAttn-Cross denote the self- and cross-frame attention map, respectively. SAttn is the spatial cross-attention map with the prompt word "astronaut". The symbol "Attn(i,j)" denotes the temporal attention map between frame i and frame j. The first row shows reconstructions sampled from frames 1, 4, 16, and 24, respectively. In the TAttn-Cross images, the frame number were manually chosen to best illustrate the cross-frame attention between the astronaut and the background. Please refer to the main text for more details.

3.1 Pipeline

As mentioned above, keyframing wiki (2023) is a technique that defines properties of images at particular frames (keys) in timeline and then automatically interpolates these values to achieve a smooth transition between the keys. It is widely used in the movie animation and visual effects industries since it reduces the artist's work while simultaneously producing temporally smooth motion that would be hard to achieve if the artist directly edited every image. Our system takes the 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 three frames only. Specifically, the sitting cat in the first part of the video is obtained with two identically positioned bboxes on the left, with the keyframes at the beginning and middle of the timeline and the prompt word "sitting" associated with both. A third keyframe is placed at the end of the video, with the bbox positioned on the right together 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 cerspense (2023) in all our experiments with no neural network training, finetuning, or optimization at inference time. Our pipeline is shown in Fig. 3. The spatial cross attention and the

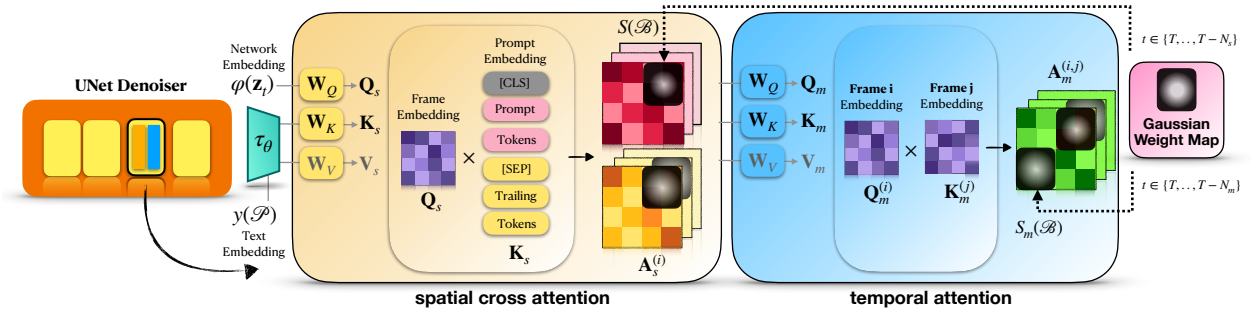

Figure 3: **Pipeline Overview.** Our pipeline highlights the central components of spatial cross-attention editing (left, in the blanched almond-colored section) and temporal cross-frame attention editing (right, in the blue section). This operation is exclusively applied during the denoising process in the early stage. 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.

temporal attention is discussed in detail in the Sec. 3.2 and Sec. 3.3, respectively. The editing of all the spatial and the temporal editing is performed in the early steps $t \in \{T, ..., T - N_S\}$, and $t \in \{T, ..., T - N_M\}$ of 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 spatial and temporal attention editing. The parameter setting is detailed in our supplementary material.

In the subsequent sections we describe how our algorithm is implemented by modifying the spatial and temporal attention in a pre-trained diffusion model. Please refer to Rombach et al. (2022); Song et al. (2021); Ho et al. (2020); Weng (2021) for background on overall diffusion model architectures.

Our system processes a set of keyframes, encompassing associated bbox regions $\mathcal{R}_f$ and prompts $\mathcal{P}_f$ at frame f, where f denotes the frame index within the range $f \in \{0, ..., N_F\}$. In practice, users are required to specify a minimum of two keyframes at the start and end of the video sequence. Then, the information in these keyframes is linearly interpolated, such as the bbox $\mathcal{B}_f$ and the prompt text embedding $y(\mathcal{P}_f)$ through the text encoder $y(\cdot)$. To enhance readability, we omit subscript f and the linearly blended video sequence between the keyframes when discussing the core method.

The region $\mathcal{R}$ is characterized of a set of parameters $\mathcal{R} = \{\mathcal{B}, \mathcal{I}, \mathcal{T}\}$: a set of bbox positions (e.g., $\mathcal{B}$), the indices of the subject we would like to constrain (e.g., $\mathcal{I}$), and the indices of the trailing maps (e.g., $\mathcal{T}$) to enhance the controllability. 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 indices $\mathcal{T}$ and the bounding box (bbox) $\mathcal{B}$ are defined as below.

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 the cross-attention maps generated without a prompt word association, where N_P denotes the maximum prompt length that a tokenizer model can take, which is $N_P = 77$ when CLIP is used Radford et al. (2021). It serves as a means of controlling the spatial location of the synthesized subject and its attributes. A larger number of trailing indices set provides greater controllability but comes with the risk of failed reconstruction, as illustrated in Fig. 8. We refer reader to the work Ma et al. (2023) for more detail.

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 $w \times h$. In our implementation, $\mathcal{B}$ is produced by 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 $b_{\text{left}}, b_{\text{top}}, b_{\text{right}}, b_{\text{bottom}} \in [0, 1]$ in the ratio of the synthesis resolution. The height and the width, denoted as h and w respectively, are defined by the resolution of the UNet intermediate representation Rombach et al. (2022).

3.2 Spatial Cross Attention Guidance

The spatial cross attention modules are implemented in the denoising UNet module of Rombach et al. (2022). This module finds the cross attention between the query representation $\mathbf{Q}_s \in \mathbb{R}^{N_F \times d_h \times d}$ obtained from the SD latent $\mathbf{z}_t$, and the representations $\mathbf{K}_s, \mathbf{V}_s \in \mathbb{R}^{N_F \times |W| \times d}$ of the $|W|$ prompt words from the text model, where d is the feature dimension of the keys and queries. Usually $|W| \equiv 77$ when the text embedding model is CLIP Radford et al. (2021).

The cross attention map Hertz et al. (2022) is then defined as $\mathbf{A}_s = \text{Softmax}(\mathbf{Q}_s\mathbf{K}_s^T/\sqrt{d}) \in \mathbb{R}^{N_F \times d_h \times |W|}$, [2] where $d_h \equiv w \times h$, defined by the spatial resolution height and width at the specific layer. Note for simplicity we omit the batch size and the number of the attention heads Vaswani et al. (2017) in our definition.

As illustrated in the blanched almond-colored section in Fig. 3, we guide the denoising path by editing the spatial cross attention (e.g., Ma et al. (2023)) for the attention maps $\mathbf{A}_s^{(i)} \in \mathbb{R}^{N_F \times d_h}$ associated with a particular prompt word and trailing indices $i \in \mathcal{I} \cup \mathcal{T}$. Given $\mathcal{B}$, our spatial attention editing is defined by

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

where x, y are are the spatial location indices of the attention map and $\mathcal{B}'$ is the complement of $\mathcal{B}$. $\mathrm{S}_s(\mathcal{B})$ uses a function $g(\cdot,\cdot)$ that "injects" attention inside $\mathcal{B}$, as illustrated in the gray box in Fig. 3. The parameters $c_w \leq 1$, $c_s > 0$ attenuate the attention outside of $\mathcal{B}$ and strengthen it inside. We define $g(\cdot,\cdot)$ as a Gaussian window 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, $\mathrm{W}_s(\cdot)$ 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 $a = \frac{f}{N_F}$, and $\mathcal{B}_b$, $\mathcal{B}_e$ denotes the bbox for the beginning and the end of keyframe.

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

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

where $\odot$ 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 $\mathrm{W}_s(\cdot)$. The overall result is that the attention in the cross-attention map for the particular prompt word as well as the trailing maps, is stronger in the user-specified bbox region.

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 Blattmann et al. (2023). This 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, 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, and $\mathbf{Q}_m \in \mathbb{R}^{d_h \times N_F \times d}$, and $\mathbf{K}_m \in \mathbb{R}^{d_h \times N_F \times d}$, respectively.

What is different from the spatial counterpart is that now $\mathbf{A}_m$ learns about the relation between the all activation components across frames. For instance, $\mathbf{A}_m^{(x,y,i,j)}$ denotes the activation 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 consistency. As seen in our visual investigation (Fig. 2, last row), the background attention is higher when the cross frame attention compares the frames that are temporally far from each other, and the foreground attention is higher when the frames are closer in the video sequence.

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

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

Here the normalized video temporal distance d determines the level of the weight injection as the 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*, thereby approximating the temporal "anti-correlation" effect seen in Fig. 2. The editing by $\mathrm{S}_m(\cdot)$ is performed during the initial N_M frames of the denoising process. See Fig. 9 and section 4.3 for an ablation on this parameter.

Then, similarly to Eq. 1, the temporal cross-frame attention map editing is performed as

$$\mathbf{A}_m^{(i,j)}(x,y) \coloneqq \mathbf{A}_m^{(i,j)}(x,y) \odot \mathrm{W}_m(x,y) + \mathrm{S}_m(x,y) \tag{2}$$

where $\mathrm{W}_m(\cdot)$ is defined the same as $\mathrm{W}_s(\cdot)$.

[2] Note that this is a "batch" matrix multiplication (e.g., the method torch.bmm in PyTorch Paszke et al. (2019)) 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 $\mathbf{A}^\top \in \mathbb{R}^{b \times p \times m}$.

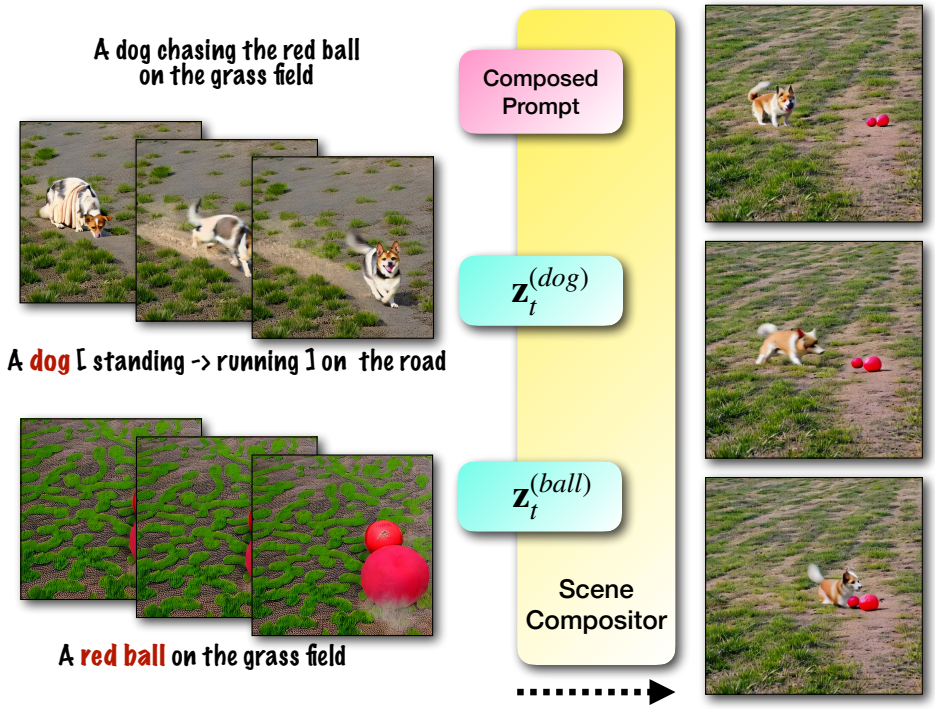

Figure 4: **Scene Compositing.** Given the set of latents generated from our system using a single bbox denoted as $\mathbf{z}_t^{(\text{ball})}$ and $\mathbf{z}_t^{(\text{dog})}$ for the case of prompts related to ball and dog. Then, the scene compositor produces a synthesis of multiple subjects with 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. 1 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 exponential in the number of subjects to find the best synthesis. If the prompt $\mathcal{P}$ and bbox $\mathcal{B}$ are in conflict then the result might be poor. For instance, a user may specify the moving of $\mathcal{B}$ from left to right associated with the prompt word "dog", while the $\mathcal{P}$ is given as "a dog is sitting on the road".

Considering the reasons above, we follow work such as Ma et al. (2023); Bar-Tal et al. (2023) that combine multiple subjects, each with their own prompt, during the latent denoising. These individual subjects are then composited into an overall image under the control of a global "composed" prompt. Our approach is illustrated in Fig. 4.

Given a sequence of subject latents $\mathbf{z}_t^{(r)}$, each generated with different prompts using the core approach in Sec. 3.2 and Sec. 3.3, the final image is produced by a second denoising process in which the N_R individual latents for the denoising step are composited using Eq. 3 for N_C steps, followed by a global denoising in which the composited latent $\mathbf{z}_t$ is denoised using the complete ("composed") prompt (pink box in Fig. 4),

$$\mathbf{z}_t(x,y) \coloneqq \frac{1}{R}\sum_{r=0}^{N_R} w\,\mathbf{z}_t(x,y) + (1-w)\,\mathbf{z}_t^{(r)}(x,y), \qquad \forall t \in \{T, ..., T-N_C\}, \quad (x,y) \in \mathcal{B}_r, \tag{3}$$

where $w \in [0,1]$ 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,

$$w = 1 - \frac{\big(N_C - (T-t)\big)}{N_C}.$$

At the very beginning of the denoising step ($t = T$), 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 $w = 1$ which stops using the subject latent in the remaining denoising steps.

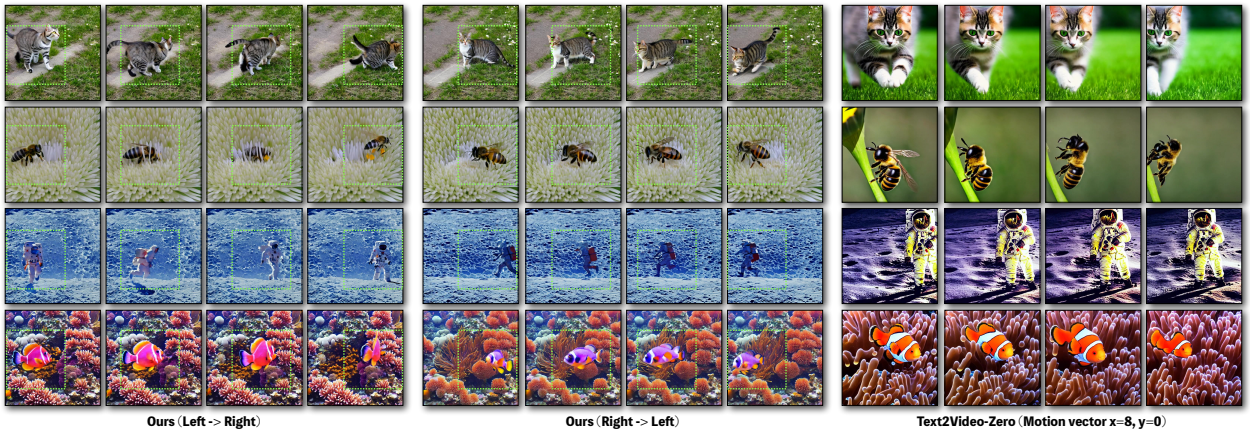

Figure 5: **Main result: Rigid moving bbox.** Transitioning the bbox movement from left to right is observed in the left set of four columns, while the transition from right to left occurs in the middle set of four columns. The set on the right is produced from Text2Video-Zero. The same prompt is used across each row. The video synthesis for each set of columns utilizes the same random seed and is generated from the identical prompt. (1st row): a **cat** walking on the grass field. (2nd row): A macro video of a **bee** pollinating a flower. (3rd row): An **astronaut** walking on the moon. (4th row): A **clown fish** swimming in a coral reef. The bold text represents the directed object.

4 Experiments

Here we briefly present some experiments, evaluation, and limitations of our work. For full experiments and the implementation details we refer the reader to our supplementary and the project video for further examination. The figures here show an evenly spaced sampling of frames from the videos. Please zoom in to see the finer details of the results.

4.1 Main result

Fig. 5 shows our main result on trajectory control of a single subject. As a baseline, the right side of the figure shows Text2Video-Zero Khachatryan et al. (2023)[3] using the same prompts, and we used it without conditioning guidance (e.g., edge or depth maps) to provide a fair comparison. Text2Video-Zero accepts motion guidance in the form of a (x,y) translation vector. We set this vector to (8,0) to produce horizontal motion.

In Fig. 5, the left two sets are our results generated with linear interpolated bboxes starting at one side of the image and moving to the other boundary. It is evident that our result shows anatomically plausible motion of the subject. For instance, all subjects (e.g., cat, bee, astronaut, and clown fish) face in the direction that they move. This also applies for the other experiments in this paper. However, this does not generally happen in Text2Video-Zero as seen on the right. This is because they apply the the warping operation on the latent space directly, which simply translates the subject without re-orienting it. In addition it can be seen that background details are not preserved across frames.

Fig. 6 illustrates dynamically changing the bbox size, producing an effect of the subject moving toward or away from the virtual camera. Similar to the Fig. 5, the bbox setup is from the top-left corner to the bottom-right corner. The dynamically changing bbox size is annotated with a green box. Note that the generated subjects share a desirable characteristic, that the subject naturally faces toward the virtual camera when the bbox transitions from small to large as seen in the sequences on the left, and vice versa on the right. The results also show a desirable perspective effect. Increasing or reducing the bbox size over time enables the synthesized object to produce the motion of "coming to" and "going away from" the camera. The red car also mimics motion blur and rising dust, simulating the effect of fast driving. We believe these effects arise naturally as a result of manipulating a model that was trained on video sequences rather than images. For example, while a video where a side-facing car appears bigger in the image is conceivable, for example if the camera is approaching the car from the side, it is much less common than videos where the car appears bigger due to it approaching.

Multi-subject synthesis is generally challenging, particularly when the number of objects exceeds two. We will delve into these limitations in Sec. 4.4. In Fig. 7, we present experiments with two subjects, a cat and a dog, guided by the

[3]We used the Huggingface implementation: https://huggingface.co/docs/diffusers/api/pipelines/text_to_video_zero.

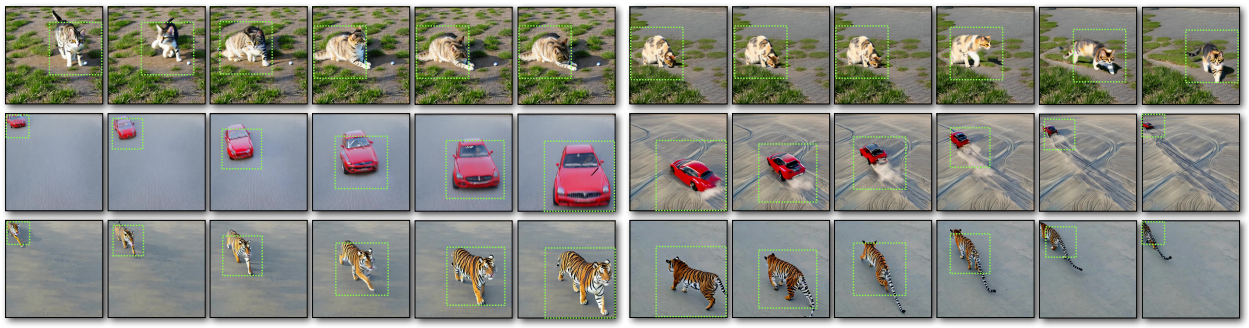

Figure 6: **Main result: Dynamic moving bbox.** (Top row): A cat [X] on the road, where [X] denotes [walking → sitting] and [sitting → walking] for the left and right sequence, respectively. The bbox is initially moving, or keeping still in the left and the right sequence. (Middle row): A **red car** driving on the road. (Bottom row): The **tiger** walking on the street. The bbox used in middle and the bottom row are linearly interpolated with varied sizes. Please refer to the main text for more detail.

Method	FID(↓)	IS(↑)	KID(↓)
Text2Video-Zero Khachatryan et al. (2023)	348.87	**3.68** $\pm$ 0.19	0.19 $\pm$ 0.01
TrailBlazer	**317.35**	3.63 $\pm$ 0.25	**0.18** $\pm$ 0.01

Table 1: Quantitative results.

green bbox in the sub-figure. The synthesis of the dog and cat in isolation is depicted in the top row, serving as a quality sanity check. Starting from the second row, we show the subsequent eight results combining environment prompts (e.g., "... on the moon") after subject prompts (e.g., "A white cat and a yellow dog running..."). Each experiment demonstrates the flexibility of our method to synthesize subjects under varied environmental conditions. Notably, the interactions between the background and subjects appear plausible, as seen in reflections and splashes in the swimming pool case and consistent shadows across all samples. The results also show some artifacts such as extra limbs that are inherited from the underlying model. Indeed such artifacts are common in some diffusion models.

4.2 Quantitative evaluation

Following the methodology in Blattmann et al. (2023); Hu and Xu (2023), we report Frechet Inception Distance Heusel et al. (2017) (FID), Inception Score (IS), and Kernel Inception Distance (KID) metrics against the CIFAR10 dataset Krizhevsky et al. on all images of video sequences, using the PyTorch implementation Obukhov et al. (2020). For a fair quantitative evaluation, we generated baseline results using Text2Video-Zero Khachatryan et al. (2023) without additional conditioning input. We set the motion vectors to $x = 8, y = 0$ and create a 24-frame video sequence as our baseline comparison. Subsequently, we manually annotated the beginning and end frames of their result with bboxes around the subject, and use these bboxes as keyframes to drive our system for comparison. We randomly sample 10 videos from each of the four prompts in Fig. 5, resulting in 960 testing images for each method. The result is summarized in Table. 1. As observed, our performance is roughly equivalent in terms of IS and KID, while our FID is significantly lower than that of Text2Video-Zero.

Unfortunately there is no standardized quantitative measure for our task. For example, while the image samples from Text2Video-Zero in Fig. 5 appear visually reasonable, its synthesized video tends to exhibit temporal flickering, and the motion kinematics are not realistic (e.g., the cat moves to the left in Fig. 5 while consistently facing the observer). For a full and accurate assessment it is essential to see the video results. Please refer to our supplementary material and the accompanying video.

4.3 Ablations

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

Trailing attention maps. Fig. 8 shows an ablation varying the number of trailing attention maps used in our spatial cross attention process, where the top row shows our method without trailing attention maps (e.g, $|\mathcal{T}| = 0$) to $|\mathcal{T}| = 30$. The guided bbox is identical to our earlier experiments in Fig. 5. It is observed that the astronaut remains static at the image center without the trailing attention maps. In contrast, the synthesis with a large number of trailing attentions can

Figure 7: **Main result: Subjects compositing.** Each set of the three sub-figures representing the first, middle, and the end frame of the synthesized video. The first row shows the video synthesis of the two subjects: "cat" and the "dog" guided by the bbox directed by the annotated arrows, respectively. Starting from the second row, each set of results show the varied post-fixed prompt.

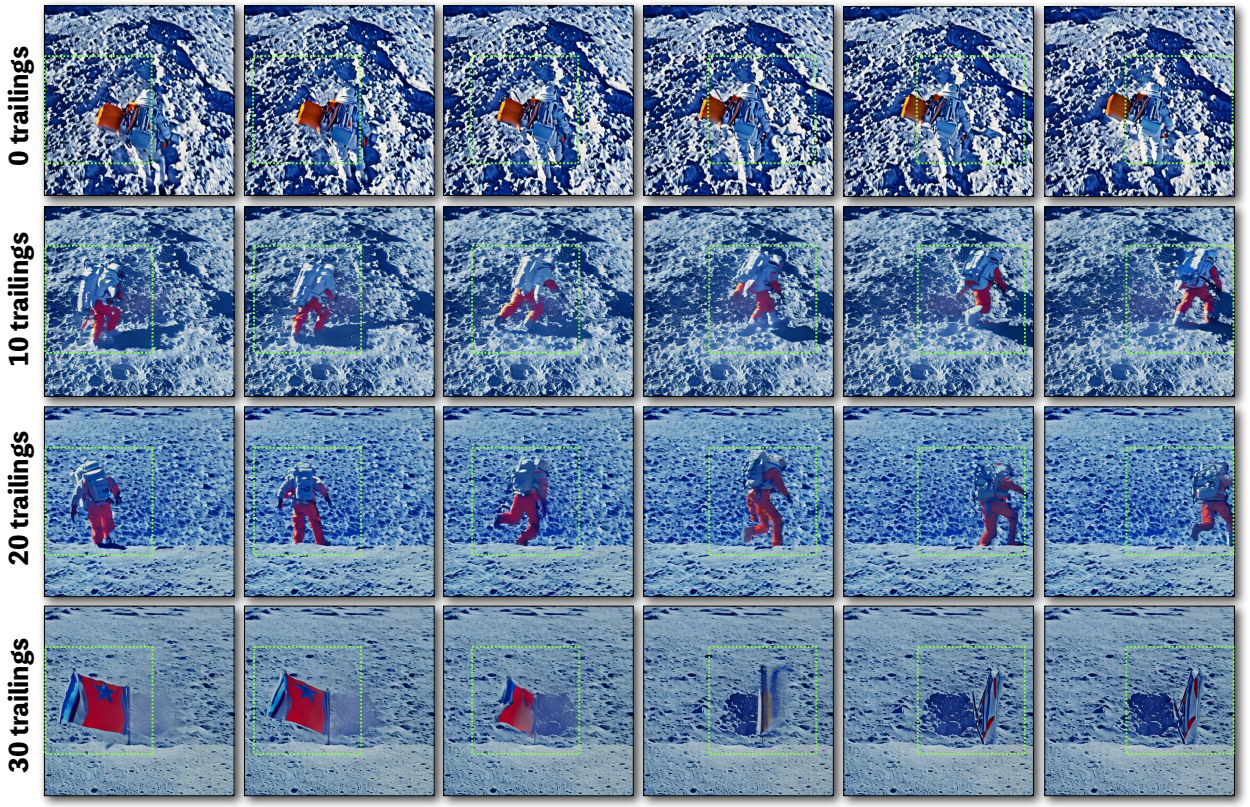

Figure 8: **Ablation: Trailing maps.** The rows from top to bottom show the video synthesis with 0 (no trailing maps), 10, 20, and 30 trailing maps. Prompt: "The **astronaut** walking on the moon", where "astronaut" is the directed subject. The number of temporal edit steps is five in all cases.

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. 9 with varied number of temporal attention editing steps. We take the case of the astronaut from Fig. 8 with $|\mathcal{T}| = 10$, 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 satisfactory result on 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.

4.4 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. 10. 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 (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 bounding box guides object placement without precisely constraining it. This is an advantage as well, however, since otherwise the user would have to specify the correct x-y aspect ratio for objects, a complicated task for non-artists.

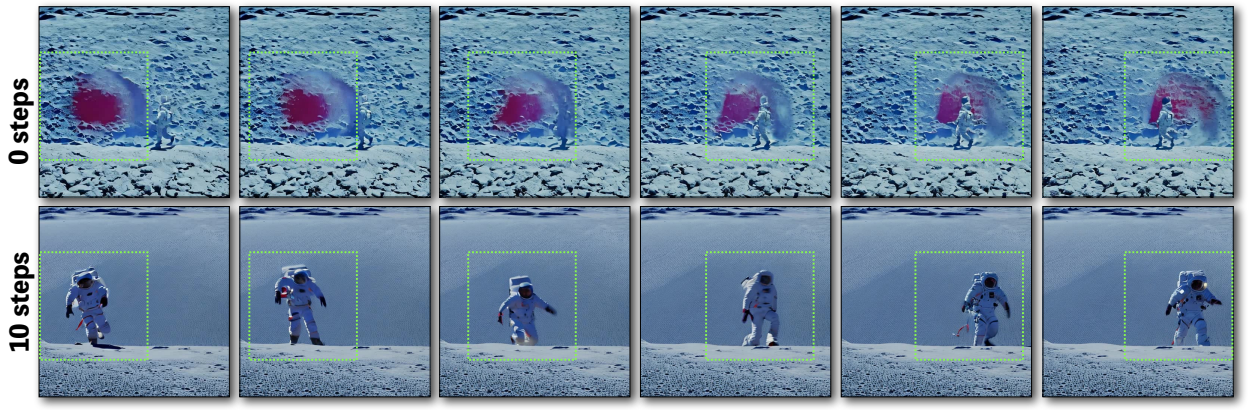

Figure 9: **Ablation: Temporal edits.** Following up the experiments in Fig. 8, the ablation test on the temporal attention editing with varied number of steps of the first and last frame of video reconstruction, shown at the left/right of each set of experiments. (Left/Right): No temporal attention editing, and 10 steps editing, respectively. The number of trailing is 10 for the two cases.

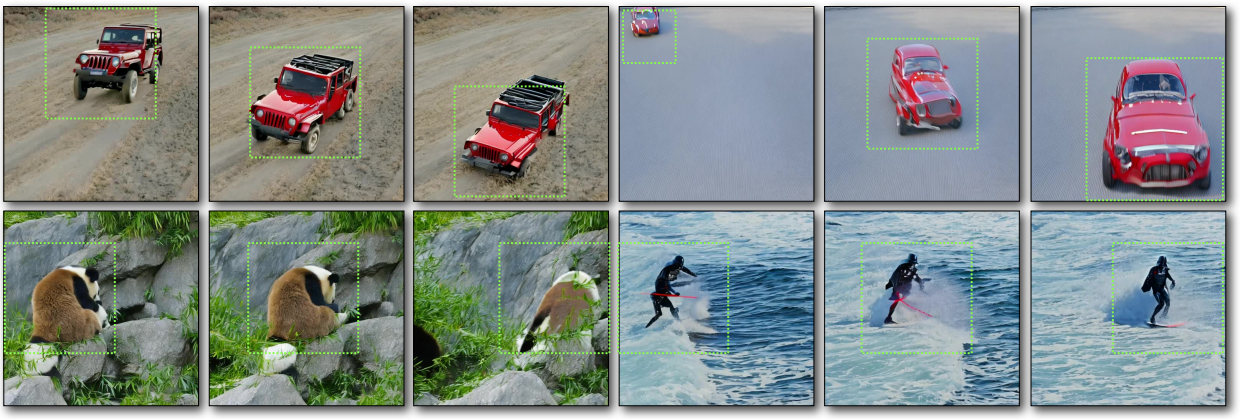

Figure 10: **Failure cases.** Prompts used from (a) to (d): "**A red jeep** driving on the road", "**A red car** driving on the highway", "**a panda** eating bamboo", and "**Darth Vader** surfing in waves", where the bold prompt word is the directed subject.

5 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 operating in the pre-trained ZeroScope model. Our contributions are several. Firstly, the spatial location of a subject can be guided through simple bounding boxes. Secondly, 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 pipeline for 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 effects such as perspective, motion with the correct object orientation, and object/environment interactions arising automatically.

TrailBlazer: Trajectory Control for Diffusion-Based Video Generation

Supplementary Material

6 Implementation

In this section, we describe details of our implementation, including the core library, hyperparameters, and other pertinent information. Our method is developed using PyTorch 2.01 Paszke et al. (2019), and the Diffusers library version 0.21.4 from Huggingface Huggingface (2023). We override the the Diffusers pipeline `TextToVideoSDPipeline` to produce our implementation.

Parameters are selected as follows: We use classifier-free guidance with a strength of 9, conduct 40 denoising steps, and maintain a video resolution of 512x512 for the conventional stable diffusion backward denoising process. Regarding the parameters specific to our proposed method, the default values are as follows: We perform 5 editing steps for both spatial and temporal attention (denoted as $N_S \equiv N_M \equiv 5$). The editing coefficients $c_m \equiv 0.001$ and $c_s \equiv 0.1$ are used in both spatial and temporal attention in most cases. The number of trailing attention maps $|\mathcal{T}|$ is the only parameter that needs to be tuned. Generally, $10 \leq |\mathcal{T}| \leq 20$ yields satisfactory results in practice.

As highlighted in section 1, we adapt the pre-trained ZeroScope[4] cerspense (2023) T2V model. This model is fine-tuned from the initial weights of ModelScope Luo et al. (2023)[5] utilizing nearly ten thousand clips, each comprising 24 frames as training data. Consequently, we adhere to the recommended practice of setting the length of the synthesized sequence to 24 frames, drawing insights from user experiences shared in relevant blogs[6].

Spatial attention editing is performed at several resolutions with a module with the following architecture:

```
transformer_in.transformer_blocks.0.attn2
down_blocks.0.attentions.0.transformer_blocks.0.attn2
down_blocks.0.attentions.1.transformer_blocks.0.attn2
down_blocks.1.attentions.0.transformer_blocks.0.attn2
down_blocks.1.attentions.1.transformer_blocks.0.attn2
down_blocks.2.attentions.0.transformer_blocks.0.attn2
down_blocks.2.attentions.1.transformer_blocks.0.attn2
up_blocks.1.attentions.0.transformer_blocks.0.attn2
up_blocks.1.attentions.1.transformer_blocks.0.attn2
up_blocks.1.attentions.2.transformer_blocks.0.attn2
up_blocks.2.attentions.0.transformer_blocks.0.attn2
up_blocks.2.attentions.1.transformer_blocks.0.attn2
up_blocks.2.attentions.2.transformer_blocks.0.attn2
up_blocks.3.attentions.0.transformer_blocks.0.attn2
up_blocks.3.attentions.1.transformer_blocks.0.attn2
up_blocks.3.attentions.2.transformer_blocks.0.attn2
```

For temporal attention editing, we found that a multiple-resolution approach was not necessary and produced unpredictable results. Instead, temporal attention editing uses a single layer:

```
mid_block.attentions.0.transformer_blocks.0.attn2
```

[4]Huggingface (2023):cerspense/zeroscope_v2_576w

[5]Huggingface (2023):damo-vilab/modelscope-damo-text-to-video-synthesis

[6]https://zeroscope.replicate.dev/

7 Comprehensive ablations

Given the limited space in the primary text, here we offer more supplementary ablation tests to substantiate our proposed approach. Broadly, we illustrate the impact of the spatial and temporal placement of guidance bounding boxes *(bboxes)* on the overall result quality, exploring the effect of various bbox speed and size choices directed by user keyframing. To see details, please zoom in to the experiment images, and **especially refer to our supplementary video.**

Fig. 11 illustrates video synthesis using the pre-trained ZeroScope model **without** applying our approach. Broadly, all the synthesized results exhibit fine details with plausible temporal coherence as would be seen in a real video featuring relatively slow motion. However, several side effects may be introduced alongside this realism. For example, the synthesized subject is often positioned in the same general area near the center of the images regardless of portrayed motion, and subjects like a galloping horse do not conveying the notion of speed. Additionally, artifacts such as extra or missing limbs (e.g., the cat in the second row) or other implausible results occasionally occur.

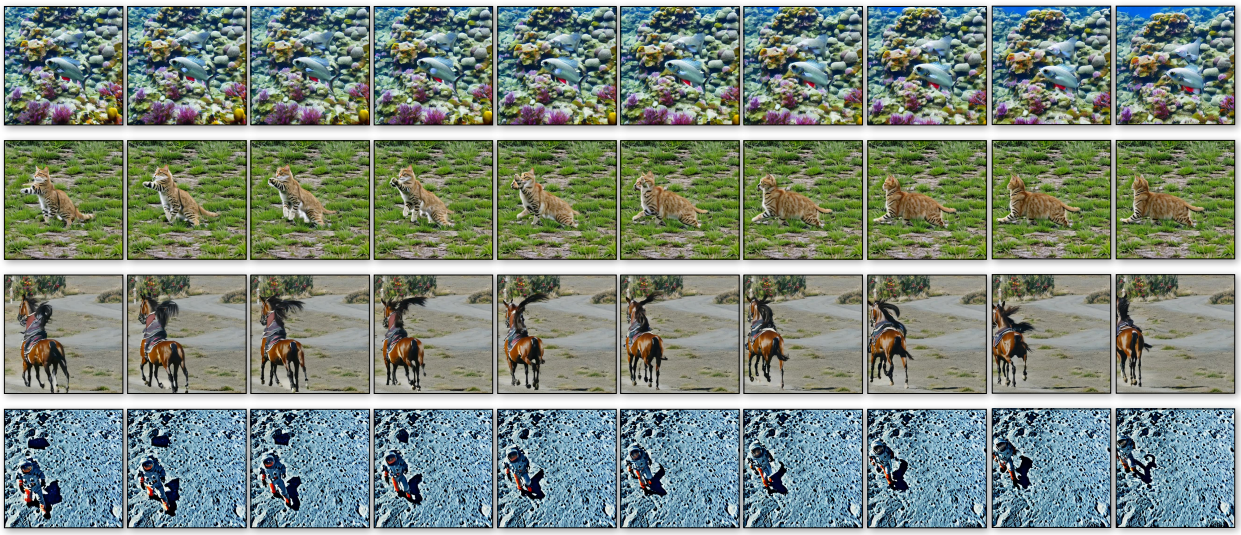

Figure 11: **Baseline results.** Each row shows equally-spaced frames sampled from a video generated using ZeroScope *without applying our trajectory control approach*. The prompts used starting from the first row: “A fish swimming in the sea”, “The cat running on the grass field”, “The horse galloping on the road”, and “An astronaut walking on the moon”. These prompts are reused in subsequent examples in these supplementary results.

7.1 Exploration and Ablation: Varied static bbox sizes

Fig. 12 shows the effect of the size of the bbox without considering motion. The results indicate that the bbox size significantly influences the outcome. In extreme cases, the top row illustrates that a smaller bbox may yield unexpected entities in the area (e.g., white smoke next to the horse) or information leakage to the neighboring area (e.g., the blue attribute affecting the road). In contrast, the bottom row demonstrates that a overly large bbox can lead to broken results in general (e.g., the fish disappearing into the coral reef, and the strange blue pattern in place of the expected blue car). We expect this may be in large part due to the centered-object bias Szabó and Horváth (2021) in the pre-trained model's training data.

Our recommended bbox size falls within the range of 30% to 60% for optimal reconstruction quality. Note that very small- or large-sized bboxes can still be employed in our approach, but they are best specified for a particular frame rather than the entire sequence. This is demonstrated, for example, in Fig. 13 guiding the swimming fish.

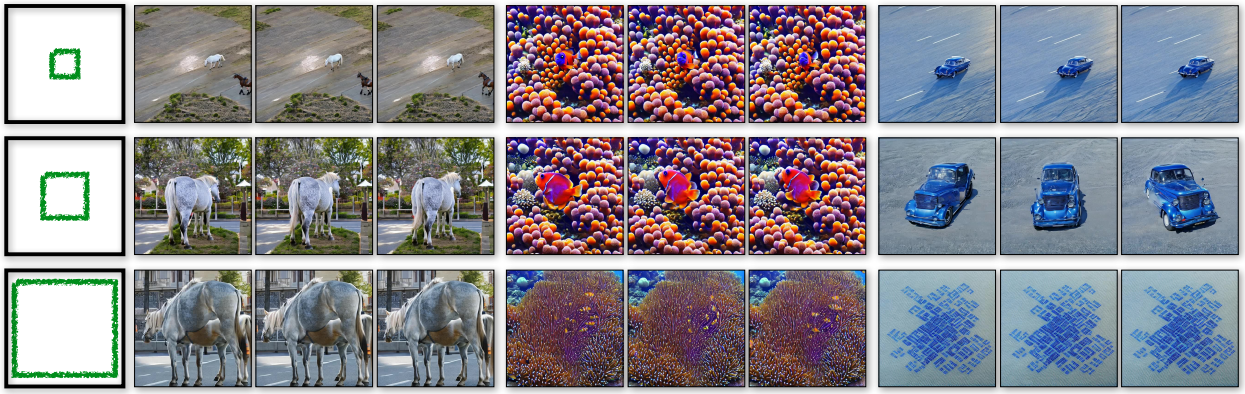

Figure 12: **Static bbox sizes.** Each row shows the result of a static square bbox positioned at the center, where the width and height are 25%, 50%, and 90% of the original image size (represented by the the green square on the left). The prompts used in the three sets of the experiments are: "The **white horse** standing on the street", "The **fish** swimming in the sea", and "The **blue car** running on the road".

7.2 Exploration and Ablation: Varied dynamic bbox sizes

Fig. 13 demonstrates video synthesis with a dynamically changing bbox size. In the top-left example, the bbox grows larger and then shrinks, resulting in a perspective effect where the fish swims towards the camera and then away from it. The frame highlighted in red indicates the middle keyframe with a large bbox. This aligns with our main text results in Fig. 6, showcasing that the animated tiger and car respect the bbox size. The top-right example is a comparison to the top-left, portraying the fish only swimming toward the camera.

The second and the third rows show a comparison of the same bbox condition with the prompt words "fish" (second row), and "sardine" (third row), respectively. This experiment aims to assess how well our method adapts to large bbox size variations, represented by the short/wide target bbox on the left and tall/thin target bbox on the right. The result on the left indicates that the output from the "fish" prompt does not adequately conform to the short-wide aspect ratio of the bounding box, whereas the result from the "sardine" prompt can more closely adjust to the desired bbox thanks to the elongated shape of the sardine. Conversely, in the experiment on the right, both "fish" and "sardine" perform well with the tall/thin bounding box, since the tall aspect ratio can be satisfied by a fish facing directly toward or away from the camera. In general we expect that the obtained results will mimic the situations found in ZeroScope's training data, while views that are outside the typical data (such as a fish swimming vertically, or a horse at the top of the image) will be difficult to synthesize.

As with all our results, we see that the guided subject *approximately* follows the specified bounding box, but does not exactly lie within the bbox. While this is a disadvantage for some purposes, we argue that it is also an advantage for casual users – if the subject exactly fit the bounding box it would require the user to imagine the correct aspect ratio of the subject under perspective (a difficult task for a non-artists) as well as do per-frame animation of the bbox to produce the oscillating motion of the swimming fish seen here.

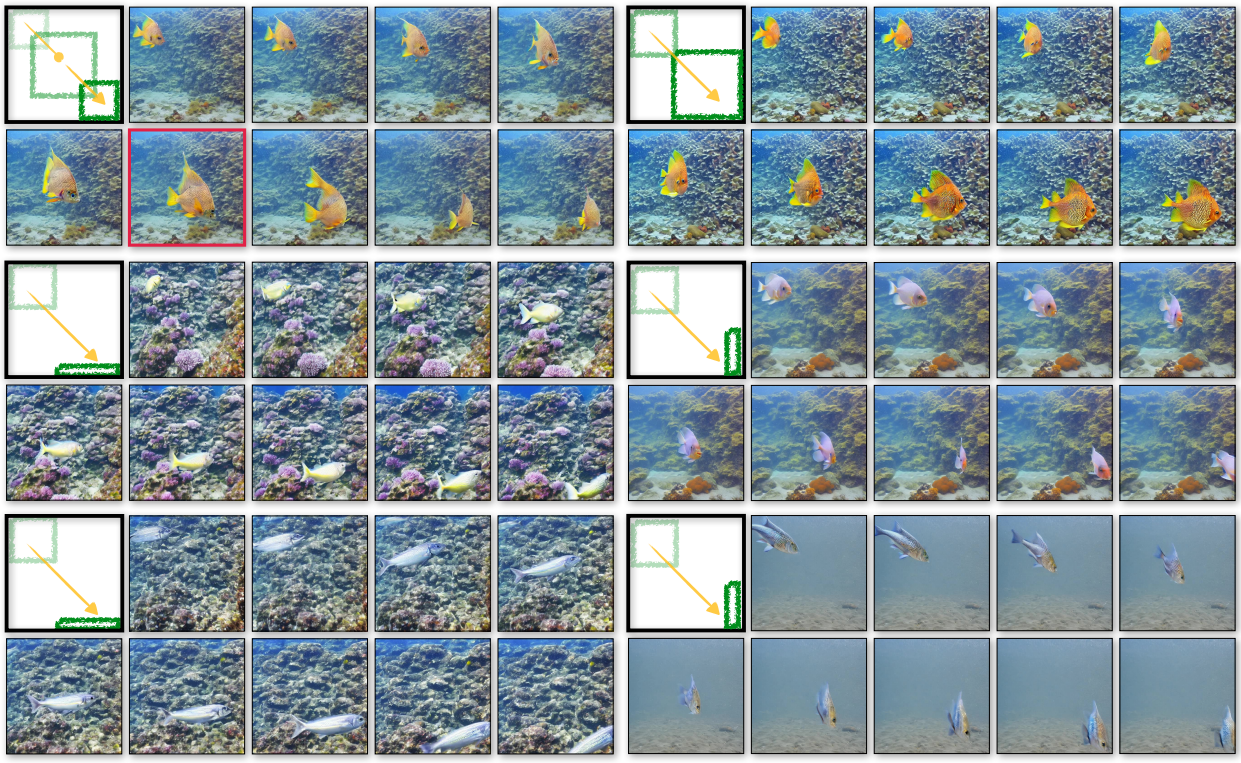

Figure 13: **Dynamic bbox sizes.** The result showcases six synthesized video sequences with the subject directed by the yellow arrow starting at the position indicated by green bbox. The number of the bboxes (corresponding to the number of keyframes used in the experiment) is, clockwise from top-left, $|\mathcal{K}| = 3$, 2, 2, 2, 2, and 2, respectively. The prompt used in each result: "The [X] swimming in the sea", where "[X]" denotes the "fish" for the first and second rows, and "sardine" for the third row.

7.3 Exploration and Ablation: Speed control with multiple keys

Fig. 14 demonstrates controlling the subject's speed through varying the number of keyframes in the video synthesis. Given the recommended sequence length $N_f = 24$ for ZeroScope, we show the result of adding different keyframes in between the start and end keyframes at the left/right image boundary, simulating the cat running back and forth on the grass field. It is clear that the cat moves relatively naturally according to the motion flow indicated by the yellow arrows. For instance, the cat looks back first before turning around, rather than showing an unnatural motion where the position of the head and tail is instantaneously swapped. As the cat moves faster, motion blur also introduced in the result. We found that this motion blur is hard to eliminate using negative prompts.

Figure 14: **Speed Test: number of keyframes.** This result shows four synthesized video sequences with the cat's motion directed according to the yellow arrows starting from the position indicated by green bbox. The number of the arrows denotes the number of keyframes (excluding the start/end keyframes) used in each experiment. Specifically, starting from the top-left and proceeding in left/right top/down (English reading) order, there are $|\mathcal{K}| = 2$, 3, 4, and 5, keyframes, respectively. The frames highlighted with red correspond to the user-specified keyframes, excluding the start and end keyframes. The prompt used for all experiments is "A **cat** running on the grass field". The red arrows in the bottom-right example shows the introduced motion blur representing fast-moving speed.

7.4 Exploration and Ablation: Controlling speed with different placement of a single keyframe

Fig. 15 shows the results of moving the subject with increasing speeds. The first row shows the astronaut moving with constant speed obtained by the linearly interpolating bboxes at the left and right of the image. Starting from the second row, the astronaut holds the position of the first bbox on the left side of the image for some period of time, then moves more rapidly to the right side of the image, as illustrated in the second column of the figure. This is obtained by changing the timing of a single "middle" keyframe $\mathcal{K}_{f_1}$, where the first keyframe and the middle keyframe have the same bbox location (e.g., $\mathcal{B}_{f_0} \equiv \mathcal{B}_{f_1}$). Similar to the results in Fig. 14, the synthesis may generate motion blur and artifacts when the speed is high (e.g., last row).

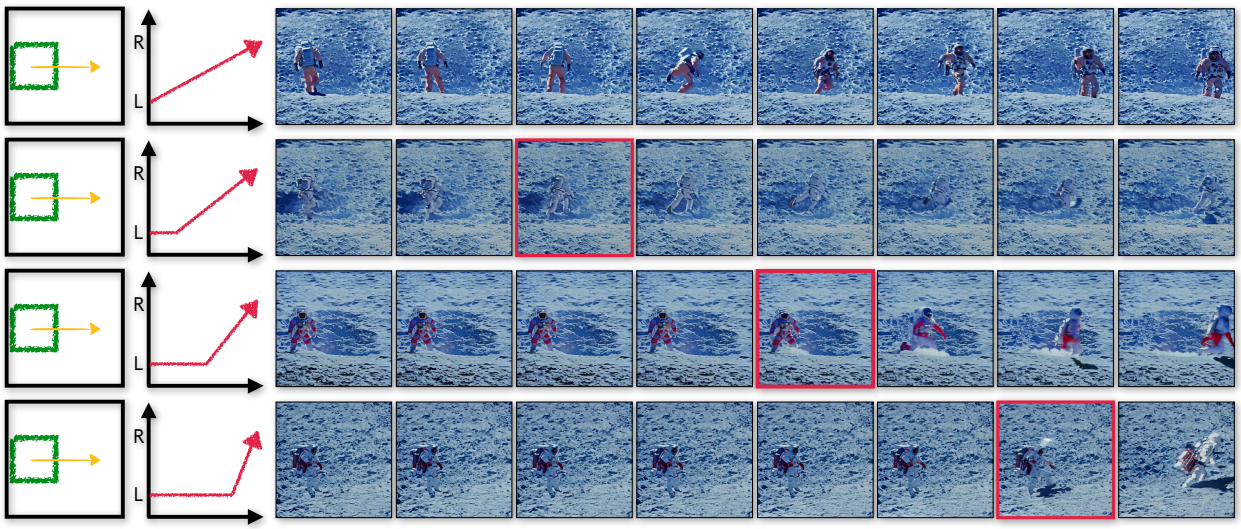

Figure 15: **Speed Test: the timing of a keyframe.** The result shows four synthesized video sequences with the subject directed according to the yellow arrow starting at the position indicated by green bbox, as illustrated in the first column. All experiments except the first use three keyframes ($|\mathcal{K}| = 3$), where the timing of the internal keyframe (e.g., $\mathcal{K}_{f_1}$) controls the duration of a stationary phase and the speed of the subsequent motion, as illustrated in the second column. The horizontal and vertical axis in the second column represent the left/right position and timing, respectively. The frame outlined in red indicates the frame controlled by $\mathcal{K}_{f_1}$, corresponding to the time when the astronaut starts to move. The prompt used for all experiments: "The **astronaut** walking on the moon".

7.5 Exploration and Ablation: Irregular trajectory

We illustrate irregular trajectories determined by varied keyframes in Fig. 16. The four experiments involve a zigzag trajectory (top-left), a triangle trajectory (top-right), a *discontinuous* trajectory (bottom-left), and a down-pointing triangle trajectory (bottom-right). In every result the horse shows high-speed running with motion blur. However, the results with turning points show limitations in depicting the horse quickly turning around and may show artifacts. For example, in the third frame of the down-pointing triangle case, the horse appears to swap its head and tail. Difficulty portraying this turn is somewhat expected, as horses cannot naturally execute tight high-speed turns, unlike cats or dogs. On the other hand, the down-pointing triangle video naturally introduces a perspective-like size change as the horse moves higher in the image, similar to the previous results in Fig. 13, and also the car/tiger example Fig. 6. In summary, maintaining consistency between the prompt and the timing and location of the keyframed bounding boxes is crucial for producing realistic results.

Figure 16: **Irregular trajectory.** The figure shows four synthesized video sequences with the horse subject directed according to the yellow arrows starting from the position indicated by green bbox. The frames highlighted in red indicates correspond to keyframes. The start and end keyframes are not indicated. The prompt used for all examples: "A **horse** galloping on the road".

References

Daniel Arijon. *Grammar of the Film Language*. Focal Press, 1976.

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. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. *CoRR*, abs/2211.01324, 2022.

Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. *CoRR*, abs/2302.08113, 2023.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

cerspense. zeroscope-v2-576w, 2023. Accessed: 2023-10-01.

Weifeng Chen, Jie Wu, Pan Xie, Hefeng Wu, Jiashi Li, Xin Xia, Xuefeng Xiao, and Liang Lin. Control-a-video: Controllable text-to-video generation with diffusion models, 2023.

Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. *ArXiv*, abs/2302.03011, 2023.

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

William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos, 2022.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017.

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

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. Imagen video: High definition video generation with diffusion models. *ArXiv*, abs/2210.02303, 2022a.

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

Zhihao Hu and Dong Xu. Videocontrolnet: A motion-guided video-to-video translation framework by using diffusion model with controlnet, 2023.

Huggingface. Stable diffusion 1 demo, 2023. Accessed: 2023-01-01.

Tobias Höppe, Arash Mehrjou, Stefan Bauer, Didrik Nielsen, and Andrea Dittadi. Diffusion models for video prediction and infilling, 2022.

Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. *arXiv preprint arXiv:2303.13439*, 2023.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

Jun Hao Liew, Hanshu Yan, Daquan Zhou, and Jiashi Feng. Magicmix: Semantic mixing with diffusion models. *CoRR*, abs/2210.16056, 2022.

Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jingren Zhou, and Tieniu Tan. Videofusion: Decomposed diffusion models for high-quality video generation, 2023.

Wan-Duo Kurt Ma, J. P. Lewis, Avisek Lahiri, Thomas Leung, and W. Bastiaan Kleijn. Directed diffusion: Direct control of object placement through attention guidance, 2023.

Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023.

Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models, 2021.

Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2022.

Anton Obukhov, Maximilian Seitzer, Po-Wei Wu, Semen Zhydenko, Jonathan Kyl, and Elvis Yu-Jing Lin. High-fidelity performance metrics for generative models in pytorch, 2020. Version: 0.3.0, DOI: 10.5281/zenodo.4957738.

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. Pytorch: An imperative style, high-performance deep learning library, 2019.

Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. Fatezero: Fusing attentions for zero-shot text-based video editing. *arXiv:2303.09535*, 2023.

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

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, abs/2204.06125, 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022.

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. Photorealistic text-to-image diffusion models with deep language understanding. *CoRR*, abs/2205.11487, 2022a.

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. Photorealistic text-to-image diffusion models with deep language understanding. *ArXiv*, abs/2205.11487, 2022b.

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

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

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

Wei Sun and Tianfu Wu. Learning layout and style reconfigurable gans for controllable image synthesis. *TPAMI*, 44:5070–5087, 2022.

Gergely Szabó and András Horváth. Mitigating the bias of centered objects in common datasets. *CoRR*, abs/2112.09195, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017.

Vikram Voleti, Alexia Jolicoeur-Martineau, and Christopher Pal. Mcvd: Masked conditional video diffusion for prediction, generation, and interpolation. In *(NeurIPS) Advances in Neural Information Processing Systems*, 2022.

Lilian Weng. What are diffusion models?, 2021.

wiki. keyframe, 2023. Accessed: 2023-10-01.

Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7623–7633, 2023.

Jinheng Xie, Yuexiang Li, Yawen Huang, Haozhe Liu, Wentian Zhang, Yefeng Zheng, and Mike Zheng Shou. Boxdiff: Text-to-image synthesis with training-free box-constrained diffusion. *CoRR*, abs/2307.10816, 2023.

Hanshu Yan, Jun Hao Liew, Long Mai, Shanchuan Lin, and Jiashi Feng. Magicprop: Diffusion-based video editing via motion-aware appearance propagation, 2023.

Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion probabilistic modeling for video generation, 2022a.

Zuopeng Yang, Daqing Liu, Chaoyue Wang, J. Yang, and Dacheng Tao. Modeling image composition for complex scene generation. *CVPR*, pages 7754–7763, 2022b.

Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

Bo Zhao, Weidong Yin, Lili Meng, and Leonid Sigal. Layout2image: Image generation from layout. *Int. J. Comput. Vis.*, 128(10): 2418–2435, 2020.