Ctrl-V: High Fidelity Video Generation with Bounding-Box Controlled Object Motion

Anonymous authors
Paper under double-blind review

Abstract

Controllable video generation has attracted significant attention, largely due to advances in video diffusion models. In domains such as autonomous driving, it is essential to develop highly accurate predictions for object motions. This paper tackles a crucial challenge of how to exert precise control over object motion for realistic video synthesis. To accomplish this, we 1) control object movements using bounding boxes and extend this control to the renderings of 2D or 3D boxes in pixel space, 2) employ a distinct, specialized model to forecast the trajectories of object bounding boxes based on their previous and, if desired, future positions, and 3) adapt and enhance a separate video diffusion network to create video content based on these high quality trajectory forecasts. Our method, **Ctrl-V**, leverages modified and fine-tuned Stable Video Diffusion (SVD) models to solve both trajectory and video generation. Extensive experiments conducted on the KITTI, Virtual-KITTI 2, BDD100k, and nuScenes datasets validate the effectiveness of our approach in producing realistic and controllable video generation.

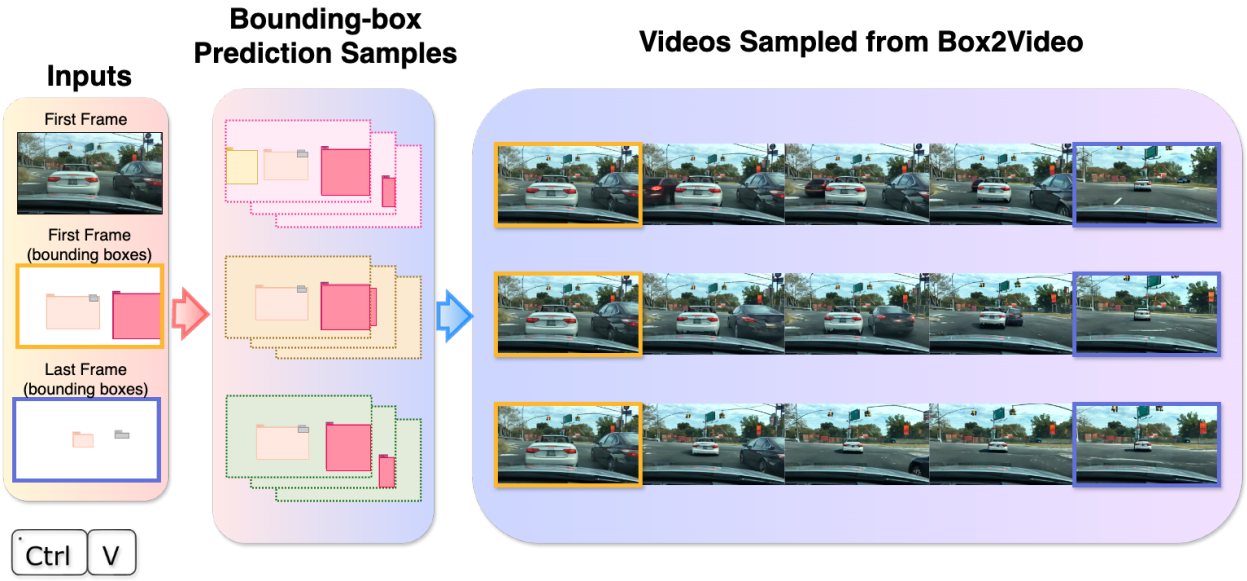

Figure 1: Overview of Ctrl-V's generation pipeline: **(Left) inputs:** Our inputs include an initial frame, its corresponding bounding box image and the final frame's bounding box image. **(Middle) generated bounding box trajectories:** We demonstrate three distinct possible trajectory sequences produced by our diffusion-based bounding box motion generation model – BBox Generator. **(Right) generated video clips:** Our Box2Video model conditions on the generated bounding box trajectory frames to produce the final video clips. We provide an anonymous project page `https://github.com/aprudentmouse/ctrlv.github.io`.

1 Introduction

Recent advances in controllable *image* generation have enabled the creation of highly realistic images from various conditioning inputs, including points, bounding boxes, scribbles, segmentation maps, and skeleton poses. Yet, translating this control to *video* generation is markedly more challenging due to the added temporal dimension. Incorporating time dynamics into diffusion models significantly complicates controllable video generation, as it requires accounting for object interactions, physical consistency, and coherent motion across frames.

Numerous recent studies have examined different forms of controllability for video generation. Researchers have used an array of methods for control, including conditioning on information such as canny edge and depth maps (Zhang et al. (2023b)), similar visual information (Chen et al. (2023)), optical flow (Hu & Xu (2023)), and pose sequences (Karras et al. (2023)). These control inputs are often expensive to produce, especially when sequences of them are required in order to condition a video. Models that use accessible conditioning such as bounding boxes require additional input such as text to help with the generation process (Wang et al. (2024)). A controllable video generation model with an accessible and simple mode of control is greatly desired.

In this work, we focus on creating such a model. Specifically, we aim to generate higher fidelity videos controlled by, at the minimum, the beginning and ending positions of 2D and 3D bounding boxes without the help of other modes of control. Our two-part method includes a diffusion-based model that generates the motions and dynamics of objects in the form of bounding box videos (2D images of the bounding boxes evolving over time), and a generative model of videos according to those bounding box videos. To this end, we choose to train and test our model on driving datasets as they contain challenging scenes rich with different types of bounding boxes as well as complex movement and irregular appearing and disappearing objects. In our experiments, we show that our model generates videos that adhere tightly to the desired bounding box motion conditioning, accurately depicting desired object movements. Additionally, through our novel pixel-level bounding box generator and conditioning, our method robustly handles the appearance and disappearance of different objects in a scene, including cars, pedestrians, bikers, and others.

In this paper, we present Ctrl-V, a diffusion-based bounding box conditional video generation method that addresses multiple challenges and makes the following contributions to generate higher-fidelity videos using diffusion techniques. Our contributions can be enumerated as follows:

1. **A Novel Diffusion Approach for Predicting Bounding Box Trajectories:** Our approach generates video frames with 2D or 3D bounding box *trajectories* at the pixel-level based on their initial and final states, and the first frame of RGB video. Our results show that our proposed approach yields higher quality bounding box trajectory predictions than a recently proposed state of the art method (see in Table 1).
2. **2D and 3D Bounding Box Conditioned Video Generation with Diffusion:** We present a method that allows video generation by conditioning on 2D or 3D bounding box trajectories which allows fine-grained control over the generated videos. Our results (in Table 2) indicate that this approach can generate video that is dramatically better than a state-of-the-art Stable Video Diffusion baseline fine-tuned without this conditioning mechanism across four commonly used quality metrics. Our approach further improves upon prior work by enabling the following capabilities: a. **Multi-subject Generation:** Synthesizing multiple subjects in videos poses significant challenges, requiring coherent object placement across frames, particularly during interactions. Existing models typically demonstrate capabilities with up to three subjects in online demo clips. Recent advances include: Boximator (Wang et al., 2024) (which can synthesize up to 8 subjects) and FACTOR (Li et al., 2024) (up to 12 subjects). Our method enables synthesis of scenes with any number of objects, limited only by the number of clearly renderable bounding boxes per frame; b. **Uninitialized Object Generation:** Most bounding box-based generation methods focus solely on objects that either remain present throughout the entire clip or appear in the first frame and persist until at least the middle of the clip. They typically overlook bounding boxes that appear only after the first frame (Wang et al., 2024). In this work, we train our model to be sensitive to all bounding boxes, whether they are present from the first frame or appear from the middle of the clip.

3. **A New Benchmark for a New Problem Formulation:** Given the novelty of our problem formulation, there is no existing standard way to evaluate models that seek to predict vehicle video with high fidelity. We therefore present a new benchmark consisting of a particular way of evaluating video generation models using the KITTI (Geiger et al., 2013), Virtual KITTI 2 (vKITTI) (Cabon et al., 2020), the Berkeley Driving Dataset (BDD 100k) (Yu et al., 2020) and nuScenes (Caesar et al., 2019).

2 Related Work

Video latent diffusion models (VLDMs) extend latent image diffusion techniques (Rombach et al., 2022) to video generation. Early VLDMs (Blattmann et al., 2023b;a; He et al., 2023; Zeng et al., 2023; Wu et al., 2023) shows temporally consistent frame generation and are tailored for text-prompted or image-prompted video generation. However, these models often struggle with complex scenes and lack the capability for precise local control.

Conditional Video Diffusion techniques providing a certain degree of control. Methods like VideoCompose (Wang et al., 2023a), Dreamix (Molad et al., 2023), Pix2Video (Ceylan et al., 2023), and DreamPose (Karras et al., 2023) propose various designs of novel adapters on top of VLDMs in order to incorporate different conditioning to achieve frame-level control. **ControlNet Adapted Video Diffusion**, on the other hand, achieve precise regional or pixel-level control in video generation by utilizing ControlNet (Zhang et al., 2023a) adapters within VLDM frameworks. Models such as Control-A-Video (Chen et al., 2023), Video ControlNet (Hu & Xu, 2023; Chu et al., 2023), ControlVideo (Zhang et al., 2023b), and ReVideo (Mou et al., 2024) show that these adapters are highly adaptable to various types of conditioning, easy to train, and allow for more precise manipulation and enhanced accuracy in editing and creating video content.

Motion Control with bounding box Conditioning There are many strategies of control that have been explored in controllable video generation research. Notably, ControlVideo (Zhang et al., 2023b) utilizes a training-free strategy that employs pre-trained image LDMs and ControlNets to generate videos based on *canny edge and depth maps*. Control-A-Video (Chen et al., 2023) leverages a controllable video LDM that combines a pre-trained text-to-video model with ControlNet to manipulate videos using *similar visual information*. Video ControlNets (Hu & Xu, 2023; Chu et al., 2023) uses *optical flow* information to enhance video generation, while ReVideo (Mou et al., 2024) depends on extracted *video trajectories*. DreamPose (Karras et al., 2023) injects *pose sequence* information into the initial noise. VideoComposer (Wang et al., 2023a) uses an array of *sketch, depth, mask,* and *motion vectors* as conditioning.

Many of these conditions, such as edge, depth, and optical flow maps, are costly to produce and lack the flexibility needed for customization. Bounding boxes emerge as a conditioning that are easily customizable and can be edited into different shape, size, locations and classes efficiently. To the best of our knowledge, six other research projects are currently exploring the use of bounding boxes for motion control in video generation. However, it is important to note that our work is distinct from these in several critical respects.

Direct-A-Video, **TrailBlazer** (Ma et al., 2024) and **Peekaboo** (Jain et al., 2024) are different training-free approaches that employ attention map adjustments to direct the model in generating a particular object within a defined region. Direct-A-Video, in particular, is a text-to-video model that learns to control camera motion during training and then adopts a training-free approach to manipulate object movements using bounding boxes. **FACTOR** (Huang et al., 2023) augmented the transformer-based generation model, Phenaki (Villegas et al., 2022), by integrating a box control module. TrailBlazer, Peekaboo and FACTOR necessitate textual descriptions for individual boxes, thus lacking direct visual grounding.

Our task setup shares mild similarities with **Boximator**(Wang et al., 2024) and **TrackDiffusion**(Fischer et al., 2023) because we also utilize bounding box conditioning during training without relying on text descriptions for individual boxes. However, our approach diverges from these text-to-video models, as our primary focus is on generating realistic videos conditioned only on a couple frames of bounding boxes, whereas Boximator and TrackDiffusion are designed to be conditioned on text information as they both are **text-to-video** models. Boximator and TrackDiffusion enhance their models by introducing new self-attention layers to 3D U-Net blocks. These layers incorporate additional conditional information, such as box coordinates and object IDs, into the pretrained VLDM model. Their bounding box information is processed using a

Fourier embedder (Mildenhall et al., 2020), which is then passed through multi-layer perceptron layers to encode. In contrast, our approach uses ControlNet and does not involve training additional encoding layers or utilizing Fourier embedder to handle the bounding box information. Moreover, Boximator introduces a *self-tracking technique* to ensure adherence to the bounding boxes in generated outputs, a technique also adopted by TrackDiffusion. This enables the network to learn the object tracking task alongside video generation, but requires a two-stage training process: one with target bounding boxes in frames, and another with the boxes removed. They demonstrate that without this technique, the model's performance markedly declines. Conversely, our model achieves alignment with the bounding box conditions without additional training.

Vehicle Oriented Generative Models DriveDreamer (Wang et al., 2023b) presents noteworthy contribution from autonomous driving domain. It takes an action-based approach to video simulation. It also makes use of bounding boxes and generate actions along with a video rendering. Within the DriveDreamer framework, Fourier embeddings (Mildenhall et al., 2020) are also employed to encode bounding box information, and CLIP embeddings (Radford et al., 2021) are used for box categorization. They focus on generating multiple camera views and do not condition on bounding box sequences, so cannot be directly compared with our problem setting. In contrast, the DriveGAN work of Kim et al. (2021) aims to learn a GAN based driving environment in pixel-space, complete with actions and an implicit model of dynamics encoded using the latent space of a VAE. While driving oriented, the approach does not focus on controlling the generation of vehicle video that respects well-defined object trajectories with high fidelity.

3 Our Method: Ctrl-V

3.1 Preliminaries

We begin here with an overview of the Stable Video Diffusion (SVD) model (Blattmann et al., 2023a), due to its importance in our approach. SVD is a diffusion based image-to-video (I2V) model performed in latent embedding space Blattmann et al. (2023b). Using an image $\boldsymbol{f}^{(0)}$ as initial condition, SVD is able to extend that single frame into a video $\boldsymbol{f} = [\boldsymbol{f}^{(0)}, \ldots, \boldsymbol{f}^{(N)}]$ where N is the length of the sequence. Notably, SVD operates in latent space, where the diffusion and denoising process act upon the latents $\boldsymbol{z}$ of the video $\boldsymbol{f}$. Here, SVD employs an image encoder ($\mathcal{E}$) and an image decoder ($\mathcal{D}$) to translate each frame into and out of latent space: $\mathcal{D}\big(\mathcal{E}(\boldsymbol{f}^{(i)})\big) = \mathcal{D}(\boldsymbol{z}^{(i)}) \approx \boldsymbol{f}^{(i)}$. At each diffusion step, SVD progressively introduces noise into the latent representations. In this work, the amount of noise is dictated by Euler discrete noise scheduling method (EDM) introduced in Karras et al. (2022). A UNet based denoiser network within the SVD is used to predict this noise in order to recover the original latent representations. The UNet, $\mathbb{U}_\theta$, is parameterized as:

$$\mathbb{U}_\theta\big(\hat{\boldsymbol{z}}_t, \boldsymbol{z}^{(0)}_{\text{pad}}, \boldsymbol{c}^{(0)}, t\big), \tag{1}$$

- $\hat{\boldsymbol{z}}_t \in \mathbb{R}^{N\times C'\times H'\times W'}$: latent representation of frames corrupted by noise at noise level t.
- $\boldsymbol{z}^{(0)} \in \mathbb{R}^{1\times C'\times H'\times W'}$: latent representation of the initial frame.
- $\boldsymbol{z}^{(0)}_{\text{pad}} \in \mathbb{R}^{N\times C'\times H'\times W'}$: Padded $\boldsymbol{z}^{(0)}$ by repeating itself along the first dimension N times.
- $\boldsymbol{c}^{(0)}$: CLIP encoding (Radford et al., 2021) of the initial frame.

The full denoiser network, $\mathbb{D}_\theta$, with an EDM noise scheduler, is formulated as

$$\mathbb{D}_\theta(\boldsymbol{z}; \boldsymbol{c}^{(0)}, \sigma_t) = \lambda_{\text{skip}}(\sigma_t)\boldsymbol{z} + \lambda_{\text{out}}(\sigma_t)\mathbb{U}_\theta\big(\lambda_{\text{in}}(\sigma_t)\boldsymbol{z}, \boldsymbol{z}^{(0)}_{\text{pad}}, \boldsymbol{c}^{(0)}; \lambda_{\text{noise}}(\sigma_t)\big) \tag{2}$$

Here λ_{skip}, λ_{out}, λ_{in} and λ_{noise} denote scaling functions, while σ_t represents the computed noise at level t. The precise mathematical definitions of these terms are detailed in Appendix B. Note that 3D UNet $\mathbb{U}_\theta$ in Equation 1, is a re-parameterized version of the one in Equation 2 (Ronneberger et al., 2015). The scaling terms are absorbed and the inputs are simplified for clarity. In the following sections, we follow the re-parameterized version in Equation 1 when refering to the UNets in our model.

3.2 Overview of our Method: Ctrl-V

Our controllable video generation method is illustrated in Figure 2. It consists of two sequential steps:

1. First, we generate bounding box frames using our diffusion based bounding box predictor network, the **BBox Generator**, which is shown on the left side of Figure 2. These frames contain only bounding boxes. They make up a video of moving (or stationary) bounding boxes and it serve as the "skeleton" for the generated video.
2. Then, we generate a video using our video generator network, **Box2Video**, shown on the right side of Figure 2, where the bounding boxes frames act as the control signal. The bounding boxes in each frame determine the objects generated in the corresponding frames of the video.

BBox Generator and Box2Video each utilizes a modified SVD backbone – illustrated by the SVD backbone in Figure 2. These backbones are adapted to their respective generation tasks. Details of each model are presented in their individual sections: BBox Generator – Section 3.3 and Box2Video – Section 3.4.

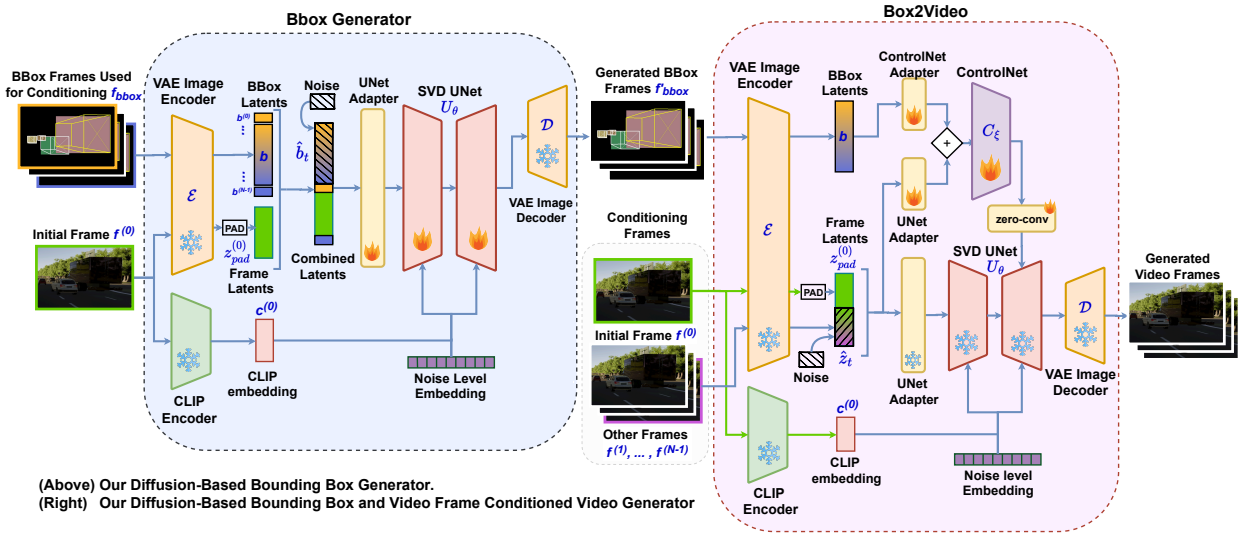

Figure 2: The diagram illustrates two components of **Ctrl-V**: (left) the **BBox Generator** and (right) **Box2Video**. For both models, we use a **frozen, off-the-shelf VAE** to encode images into latent space ($\mathcal{E}$) and decode them back into pixel space ($\mathcal{D}$). During training, (1) the **BBox Generator** (Sec. 3.3) learns to denoise the noisy bounding box frame latents $\hat{\boldsymbol{b}}_t$, conditioned on the first ($\boldsymbol{b}^{(0)}$) and last ($\boldsymbol{b}^{(N-1)}$) bounding box frame latents and the padded initial frame latent $\boldsymbol{z}^{(0)}_{pad}$ and (2) the **Box2Video** (Sec. 3.4) denoises the target frame latents $\hat{\boldsymbol{z}}_t$ by conditioning on the initial frame's latent $\boldsymbol{z}^{(0)}_{pad}$ (input to the SVD UNet) and the bounding box frame latents $\boldsymbol{b}$ (input to the ControlNet).

3.3 Ctrl-V: BBox Generator

The BBox Generator shown on the left in Figure 2 aims to predict object bounding boxes across all video frames using an SVD backbone. The four inputs to the model are $\hat{\boldsymbol{b}}_t$, $\boldsymbol{b}^{(0)}$, $\boldsymbol{b}^{(N-1)}$, $\boldsymbol{z}^{(0)}$, where: $\hat{\boldsymbol{b}}_t$ is the encoded "video" of bounding boxes with t levels of noise added; $\boldsymbol{b}^{(0)}$ is the encoded initial bounding box frame(s); $\boldsymbol{b}^{(N-1)}$ is the encoded final bounding box frame; $\boldsymbol{z}^{(0)}$ is the encoded initial video frame. During training, the model learns to predict the noise added in $\hat{\boldsymbol{b}}_t$ according to the EDM noise scheduler. The model recovers the original $\boldsymbol{b}$ from its noisy version $\hat{\boldsymbol{b}}_t$ by calculating the noise with UNet outputs and eliminating the noise through scaling functions. We opt to abstract this detail in the model diagram for readability.

In practice, the four inputs are transformed and concatenated into a vector format accepted by the UNet adapter within the SVD backbone. Specifically, as shown in Figure 2, $\boldsymbol{z}^{(0)} \in \mathbb{R}^{1\times C'\times H'\times W'}$ is replicated along the first dimension, and its front and end (in the first dimension) are replaced by $\boldsymbol{b}^{(0)}$, $\boldsymbol{b}^{(N-1)}$ respectively. This forms $\boldsymbol{z}^{(0)}_{\text{pad}} = \text{concat}(\boldsymbol{b}^{(0)}, \boldsymbol{z}^{(0)}, ..., \boldsymbol{z}^{(0)}, \boldsymbol{b}^{(N-1)}) \in \mathbb{R}^{N\times C'\times H'\times W'}$. The noise-added encoding of bounding box video $\hat{\boldsymbol{b}}_t$ is then concatenated with $\boldsymbol{z}^{(0)}_{\text{pad}}$ to form the final input to the UNet adapter. The network incorporates additional conditioning inputs, including a CLIP-encoded embedding of the initial frame $\boldsymbol{c}^{(0)}$

and a noise-level embedding t. These embeddings are individually integrated into every sub-block of the U-Net through a self-attention mechanism.

Representing Bounding Boxes in Pixel Space

An important element of Ctrl-V is our design choice of rendering bounding boxes in pixel space. The manner in which bounding box information is provided as a control signal to the video generator is important. For example, prior work such as Boximator (Wang et al., 2024) represents bounding boxes as a Fourier transformed concatenated vector of their raw coordinates, ID and other information. In contrast, in our work we choose to render bounding boxes into frames while maintaining minimal loss of meta information. Importantly, we also encode information such as track ID, object type, and orientation for each bounding box using a combination of visual attributes, including border color, fill color, and markings. Specifically, the *track ID* represents a unique identifier for each tracked object across frames, the *object type* specifies the category of the object (e.g., car, pedestrian), and the *orientation* indicates the direction the object is facing. Further details about how these bounding box frames are rendered can be found in Appendix C.1. Crucially, our approach allows us to leverage the highly effective ControlNet approach to provide pixel-level guidance to influence diffusion generated imagery.

3.4 Ctrl-V: Box2Video

Box2Video is shown on the right in Figure 2 and it aims to generate high-fidelity videos controlled by bounding box frames, such as those generated by the BBox Generator network. Box2Video consists of an SVD backbone for video generation, and an adapted ControlNet module to process the bounding box control signal. ControlNet is a widely used network for controlling image generation. In this work, we modify ControlNet and adapt it to the video diffusion framework (as shown on the right in Figure 2). This architecture allows us to train Box2Video in a single stage without the need for additional optimization criteria, in contrast to previous work such as Boximator and TrackDiffusion (Wang et al., 2024; Li et al., 2024), which require multi-stage learning with extra criteria to train their models.

The SVD component takes two inputs: $\boldsymbol{z}^{(0)}$ and $\hat{\boldsymbol{z}}_t$. Here, $\boldsymbol{z}^{(0)}$ is the encoded initial video frame and $\hat{\boldsymbol{z}}_t$ is the encoded full video with t levels of noise added to it. As shown in Figure 2, we process these inputs by padding $\boldsymbol{z}^{(0)}$ by repeating it along the first dimension before concatenating it with $\hat{\boldsymbol{z}}_t$ to create the final input to the UNet adapter of the SVD. The same input is also sent to the ControlNet module through its own UNet adapter layers. Additionally, ControlNet also receives the encoded bounding box frames, $\boldsymbol{b}$, as input, through ControlNet adapter layers. Both of these transformed input is then added together before processed by the ControlNet module. The output signal of the ControlNet module then goes through a zero-convolution before being sent to the SVD UNet decoder layers through residual paths as control signal. During training, the weights of the SVD model (θ) are frozen, while only the weights in the ControlNet (ξ) are updated.

4 Experimental Analysis and Ablation Studies

For quantitative evaluation, we assess the model's performance across four driving datasets on **three key aspects**:

1. *The overall visual quality of the generated results* (Section 4.3)
2. *The alignment of the predicted bounding box trajectories with the ground truth* (Section 4.2)
3. *The fidelity of the generated objects in the video to the bounding box control signal* (Section 4.4)

For visual assessment, Figure 3 and Appendix E showcase sample demonstrations generated by our model. To assess video quality, we randomly select 200 initial frames from each dataset's testing set and generate videos. The results in this section are based on analyses of these 200 generated videos per dataset. Furthermore, we explored different bounding box conditioning options: one or three initial bounding box frames, followed by a single final bounding box. Additional variations are discussed in Appendix E.8.

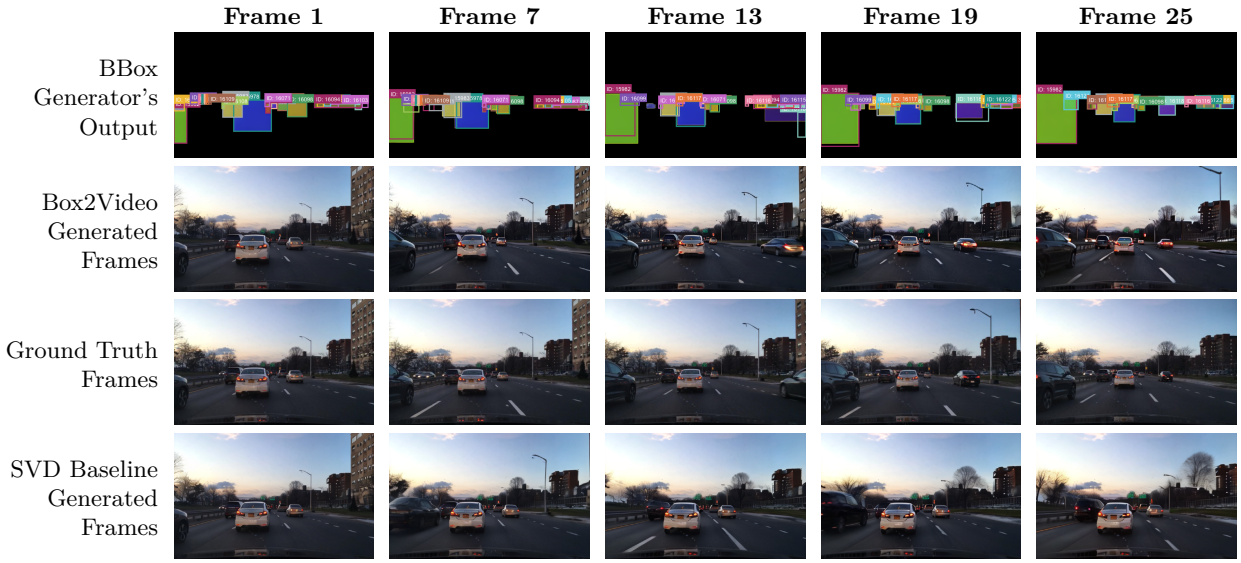

Figure 3: The first two rows illustrate video samples generated using the Ctrl-V pipeline, with one initial frame, three initial bounding box frames, and one final bounding box frame as input. The first row shows bounding box trajectories from the BBox-generator in pixel space (solid rectangles for predictions, wireframe rectangles for ground truth). The second row presents frames generated by the Box2Video model, conditioned on the BBox-generator's output. The third row displays ground-truth frames, while the fourth row shows frames generated by the Stable Video Diffusion (SVD) baseline. *In the Ctrl-V video, the car with the bright-green bounding box, which initially pokes out its nose in the lane to the left of the ego car, stays beside the ego car in the final frame. Meanwhile, the silver car with the olive bounding box, which starts in the lane to the right of the ego car, speeds off and is replaced by a new car (purple bounding box) entering the frame. These generated frames closely match the car positions seen in the conditioned inputs. In contrast, the SVD-generated video shows the black car on the left accelerating and moving ahead of the ego car, while the silver car remains in the same relative position to the ego car throughout.*

4.1 Datasets

We evaluate the performance of our models across four autonomous-vehicle datasets: KITTI (Geiger et al., 2013), Virtual KITTI 2 (vKITTI) (Cabon et al., 2020), Berkeley Driving Dataset (BDD) (Yu et al., 2020) with Multi-object Tracking labels (MOT2020), and the nuScenes Dataset (Caesar et al., 2019). KITTI comprises 22 real-world driving clips with 3D object labelling. vKITTI consists of 5 virtual simulated driving scenes, each offering 6 weather variants, all including 3D object labelling. BDD is a large-scale real-world driving dataset, featuring 1603 2D-labeled sequences of driving videos. The nuScenes dataset is a large-scale driving dataset that includes 1000 scenes 20-second scenes annotated with 3D bounding boxes, multiple sensor data (lidar, radar and cameras) and map information. Further details on dataset configurations are provided in Appendix C.3.

4.2 BBox Generator: Quantitative Evaluation

Figure 4 showcases two bounding box trajectory samples generated by the BBox Generator, conditioned on the same initial and final bounding box frames. To evaluate the quality of our bounding box generations, we create mask images for both the ground-truth and generated bounding box sequences. The mask images are generated by converting the bounding box frames into binary masks (details can be found in Appendix D.2). We then calculate the generated averaged mask Intersection over Union (maskIoU) scores, averaged mask Precision (maskP) scores, and averaged mask Recall (maskR) scores against the ground-truth bounding box masks. To assess our bounding box trajectories, we applied the "best-out-of-K" method, selecting the model with the highest maskIoU score for evaluation. In this instance, K equals 5. We compare our results with a

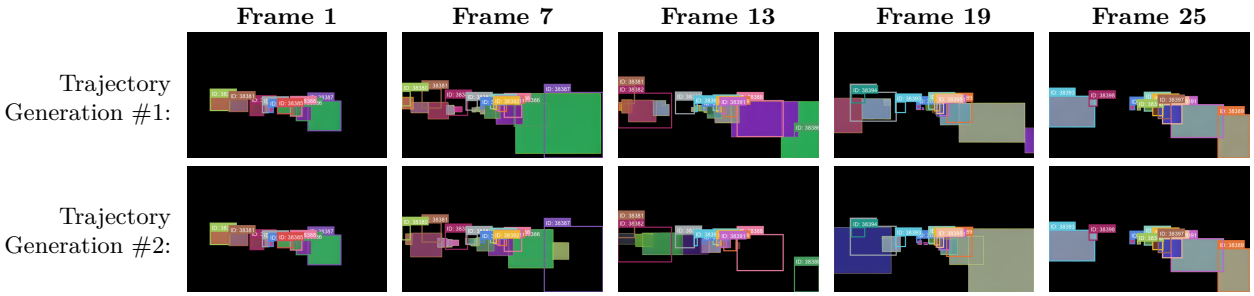

Figure 4: This figure visualizes two samples of bounding box trajectories generated by the BBox Generator, conditioned on the same set of three initial bounding box frames and one final bounding box frame (solid rectangles represent predictions, and wireframe rectangles represent ground truth). Although the intermediate frames show notable differences, the initial and final frames align closely with the ground-truth bounding boxes.

	Method	# Cond. BBox	maskIoU↑	maskP↑	maskR↑	maskIoU↑ (first+last)	maskP↑ (first+last)	maskR↑ (first+last)
KITTI	BBox Generator (ours)	1-to-1	**.629** ± .212	**.758** ± .176	**.763** ± .188	**.986** ± .012	**.994** ± .008	**.992** ± .009
	Trajeglish-Style		.447 ± .154	.568 ± .172	.679 ± .177	.561 ± .151	.663 ± .150	.789 ± .165
	BBox Generator (ours)	3-to-1	**.795** ± .112	**.881** ± .082	**.884** ± .078	**.986** ± .010	**.992** ± .007	**.994** ± .005
	Trajeglish-Style		.491 ± .164	.622 ± .173	.691 ± .175	.576 ± .154	.684 ± .149	.784 ± .163
vKITTI	BBox Generator (ours)	1-to-1	**.710** ± .205	**.828** ± .178	**.809** ± .171	**.943** ± .048	**.946** ± .046	**.997** ± .006
	Trajeglish-Style		.471 ± .171	.578 ± .200	.700 ± .187	.557 ± .171	.628 ± .194	.835 ± .135
	BBox Generator (ours)	3-to-1	**.767** ± .131	**.881** ± .126	**.853** ± .078	**.944** ± .039	**.948** ± .036	**.996** ± .006
	Trajeglish-Style		.520 ± .162	.630 ± .186	.741 ± .176	.575 ± .154	.657 ± .182	.836 ± .143
BDD	BBox Generator (ours)	1-to-1	**.587** ± .214	**.747** ± .187	**.712** ± .194	**.954** ± .047	**.955** ± .047	**.999** ± .002
	Trajeglish-Style		.305 ± .183	.372 ± .213	.658 ± .207	.432 ± .171	.483 ± .192	.840 ± .166
	BBox Generator (ours)	3-to-1	**.647** ± .176	**.784** ± .150	**.783** ± .156	**.955** ± .043	**.955** ± .042	**.997** ± .001
	Trajeglish-Style		.373 ± .185	.454 ± .206	.686 ± .193	.492 ± .190	.553 ± .208	.842 ± .154
nuScenes	BBox Generator (ours)	1-to-1	.364 ± .242	.433 ± .278	**.740** ± .186	**.983** ± .013	**.985** ± .0112	**.997** ± .003
	Trajeglish-Style		**.405** ± .202	**.506** ± .220	.661 ± .216	.511 ± .168	.603 ± .172	.789 ± .195
	BBox Generator (ours)	3-to-1	**.827** ± .150	**.892** ± .120	**.906** ± .099	**.983** ± .013	**.985** ± .012	**.998** ± .003
	Trajeglish-Style		.448 ± .194	.554 ± .213	.695 ± .196	.529 ± .172	.623 ± .177	.791 ± .192

Table 1: Comparing real and generated bounding boxes. We condition on 1 or 3 initial bounding box frame(s) and 1 final bounding box or trajectory frame. The first three columns show evaluations on the entire generated bounding box sequence, measuring the alignment scores between our generated bounding box generations and ground-truth labels. The last three columns focus on testing the auto-encoding capability of the network, evaluating only the first and last frames of the generated sequence. "BBox Generator" is our method and "Trajeglish-Style" is a baseline inspired from Philion et al. (2023) (see Appendix F for implementation details on this baseline).

baseline referred to as the "Trajeglish-Style" model, an autoregressive GPT-like encoder-decoder that models the bounding box trajectories as a sequence of discrete motion tokens. This baseline is inspired by the work of Philion et al. (2023) with implementation details provided in Appendix F. We present our findings in Table 1, and demonstrate examples of our bounding box generations on each dataset in Appendix E.

In the bounding box generation figures, our generator model achieves the closest alignment with the ground-truth in the first and last frames. This near-perfect alignment is primarily attributed to conditioning the model on the bounding boxes of these key frames. When considering all generated frames, the alignment scores decrease, as shown by the plotted demonstrations and metric results in Table 1. This is because

objects in frames do not move deterministically. *The role of the bounding box generator is to generate a plausible trajectory for moving objects from the initial bounding box frame to the last.*

Despite the disparity between the ground-truth trajectory and the generated trajectory, our Box2Video consistently generates high-fidelity videos based on either trajectory provided. Further analysis of this aspect is provided in the subsequent sections.

4.3 Ctrl-V: Generation Quality

		Pipeline	# Cond. BBox	FVD↓	LPIPS↓	SSIM↑	PSNR↑
KITTI		Stable Video Diffusion Baseline (Blattmann et al., 2023a)	None	1118.4	0.4575	0.2919	10.63
		Stable Video Diffusion Fine-tuned (Blattmann et al., 2023a)	None	552.7	0.3504	0.4030	13.01
		Ctrl-V: BBox Generator + Box2Video(Ours)	1-to-1	467.7	0.3416	0.3241	13.21
		Ctrl-V: BBox Generator + Box2Video(Ours)	3-to-1	**422.2**	**0.2981**	0.4277	13.85
		Ctrl-V: Teacher-forced Box2Video(Ours)	All	435.6	0.2963	**0.4394**	**14.10**
vKITTI		Stable Video Diffusion Baseline (Blattmann et al., 2023a)	None	922.7	0.3636	0.4740	14.61
		Stable Video Diffusion Fine-tuned (Blattmann et al., 2023a)	None	331.0	0.2852	0.5540	16.60
		Ctrl-V: BBox Generator + Box2Video(Ours)	1-to-1	400.2	0.3179	0.4714	15.78
		Ctrl-V: BBox Generator + Box2Video(Ours)	3-to-1	341.4	0.2645	0.5841	17.60
		Ctrl-V: Teacher-forced Box2Video(Ours)	All	**313.3**	**0.2372**	**0.6203**	**18.41**
BDD		Stable Video Diffusion Baseline (Blattmann et al., 2023a)	None	933.6	0.4880	0.3349	12.70
		Stable Video Diffusion Fine-tuned (Blattmann et al., 2023a)	None	409.0	0.3454	0.5379	16.99
		Ctrl-V: BBox Generator + Box2Video(Ours)	1-to-1	412.8	0.2967	0.5470	17.52
		Ctrl-V: BBox Generator + Box2Video(Ours)	3-to-1	373.1	0.3071	0.5407	17.37
		Ctrl-V: Teacher-forced Box2Video(Ours)	All	**348.9**	**0.2926**	**0.5836**	**18.39**
nuScenes	Single-View	Stable Video Diffusion Baseline (Blattmann et al., 2023a)	None	1179.4	0.5004	0.2877	13.31
		Stable Video Diffusion Fine-tuned (Blattmann et al., 2023a)	None	316.6	0.2730	0.4787	18.58
		Ctrl-V: BBox Generator + Box2Video(Ours)	1-to-1	285.3	0.2647	0.5050	18.93
		Ctrl-V: BBox Generator + Box2Video(Ours)	3-to-1	**235.0**	0.2235	0.5500	20.33
		Ctrl-V: Teacher-forced Box2Video(Ours)	All	235.5	**0.2104**	**0.5705**	**23.36**
		DriveGAN (Kim et al., 2021)	None	390.8	-	-	-
		DriveDreamer (Wang et al., 2023b)	All	340.8	-	-	-
	Multi-view	WoVoGen (Lu et al., 2023)	All	417.7	-	-	-
		Drivingdiffusion (Li et al., 2023)	All	332.0	-	-	-
		Drive-WM (Lu et al., 2023)	None	212.5	-	-	-
		BEVWorld (Zhang et al., 2024)	None	154.0	-	-	-
		Panacea (Wen et al., 2024)	All	139.0	-	-	-
		Drive-WM (Lu et al., 2023)	All	122.7	-	-	-
		DriveDreamer-2 (Zhao et al., 2024)	None	**105.1**	-	-	-

Table 2: Comparing the quality and diversity of the generated video models. The generated videos consist of 25 frames (except for our nuScenes models which consist of 11 frames videos at 4 Hz) at a resolution of 312×520, while the reported metrics from this table are evaluated at a resolution of 256×410. The "# Cond. BBox" column reports the number of ground-truth input bounding box frames used by the generation pipelines. "None" indicates that no ground-truth frames are used, while "All" indicates that all ground-truth bounding box frames are utilized. If "# Cond. BBox" is n-to-m, then it represents the number of initial bounding box frames used by the pipeline is n and the number of final bounding box frames used by the pipeline is m.

To assess the quality of video generation, we compare videos generated through 4 distinct pipelines:

1. **Pre-trained Stable Video Diffusion (SVD) baselines**[1] **without fine-tuning** (initial frame → video)
2. **Fine-tuned Stable Video Diffusion (SVD) baselines on the provided dataset** (initial frame → video)
3. **Teacher-forced Box2Video generation** (initial frame and all bounding box frames → video)

[1]Stable Video Diffusion (SVD) baseline is an image-to-video (I2V) model that generates a video sequence conditioned on a single video frame.

4. **bounding box generation with BBox Generator and Box2Video** (initial frame, one or three initial and one last bounding box frames → in-between bounding box frames and video).

We evaluate our generation across four metrics: Fréchet Video Distance (FVD) (Unterthiner et al., 2019), Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018), Structural Similarity Index Measure (SSIM) (Wang et al., 2004b) and Peak Signal-to-Noise Ratio (PSNR). These metrics either measure the consistency of frame pixels with the ground truth or the consistency of the frame latents extracted by another network. FVD[2] is an exception; it evaluates the generation distribution against the ground truth's distribution. It is important to note that while many papers report their best-out-of-K results on these metrics, due to computational constraints, we evaluate our model on a single sample for each input.

The evaluated results are reported in Table 2 and visualizations are available in Appendix E.1. These results indicate that the generation quality improves as we condition on more ground-truth bounding box frames. Details regarding the metrics and their limitations are discussed in Appendix D.1.

4.4 Box2Video: Motion Control Evaluation

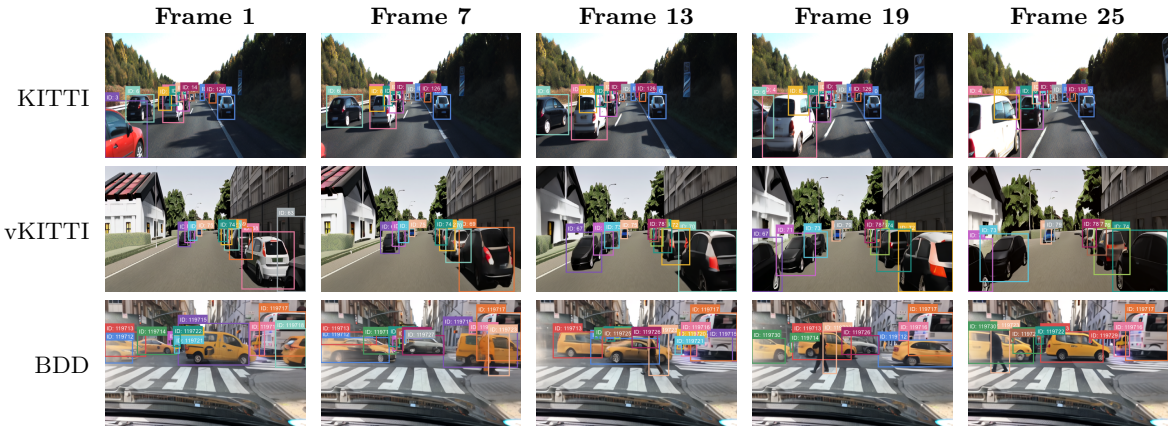

Figure 5: Illustrations of the Box2Video generations conditioned on ground truth 3D bounding box trajectories (2D for BDD) across various datasets. The 2D outlines of the ground-truth bounding boxes are overlaid on top.

Our Box2Video is trained to control object motions through bounding boxes using a teacher-forcing approach, where only ground-truth bounding box frames are provided during the training phase. In this section, we analyze the fidelity of our Box2Video generations to the ground-truth bounding box conditions. To access the consistency of objects' locations between our generated content and ground-truth, we compute the average precision of the bounding boxes in the generated frames and the ground-truth frames.

Average precision (AP) scores gauge the alignment of predicted/generated bounding boxes with the ground-truth labeling. In all related prior studies, average precision (AP) scores have been consistently reported. However, it is important to acknowledge that AP scores can vary across studies, depending on the specifics of the task setup. Boximator (Wang et al., 2024)'s motion control model predicts object locations in the scene, focusing solely on objects with consistent appearances across all frames. Their AP implementation disregards the object locations in the intermediate frames, comparing the objects' locations only in the final frame. In contrast, TrackDiffusion (Li et al., 2024) uses TrackAP for evaluation, employing a QDTrack model (Fischer et al., 2023) to track instances in generated videos and comparing them to ground-truth labels. However, these evaluated datasets have limited instances, and TrackAP requires consistent tracking across frames, making it unsuitable for our project without modifications. Therefore, our AP score differs slightly from those in previous works.

[2]FVD is highly sensitive to video configuration parameters—such as frame rate, clip duration, and spatial resolution—making direct comparisons of FVD values across studies challenging. Additionally, the metric's sensitivity to sample sizes raises concerns, as some datasets may lack sufficient samples for convergence, leading to unreliable estimates.

Method	Dataset	Dataset Type	# Frames	mAP↑	AP_{50}↑	AP_{75}↑	AP_{90}↑
Ctrl-V	KITTI	Driving	25	0.547	0.712	0.601	0.327
	vKITTI	Driving-sim	25	0.599	0.776	0.667	0.356
	BDD	Driving	25	0.685	0.855	0.781	0.401
	nuScenes	Driving	25	0.661	0.833	0.734	0.381
Boximator[3] **(Wang et al., 2024)**	MSR-VTT(Xu et al., 2016)	Web videos	16	0.365	0.521	0.384	-
	ActivityNet (Heilbron et al., 2015)	Human-action	16	0.394	0.607	0.409	-
	UCF-101 (Soomro et al., 2012)	Human-action	16	0.212	0.343	0.205	-
TrackDiffusion (Li et al., 2024)	YTVIS (Yang et al., 2019)	YouTube videos	16	0.467	0.656	-	-
	UCF-101 Soomro et al. (2012)	Human-action	16	0.205	0.326	-	-

Table 3: Average Precision scores obtained by comparing the YOLOv8 bounding box estimations of real and generated samples. Prior works (Wang et al., 2024; Li et al., 2024) do not report results on driving datasets; thus, we draw upon their reported performances on alternative datasets to provide a comparative context. The backbone model of Ctrl-V produces videos with 25 frames, while Boximator and TrackDiffusion create videos with 16 frames. Longer videos tend to have reduced quality and lower detection rates, which presents an extra challenge for our model (as it generates 56.25% more frames); yet it achieves greater precision compared to the other baseline models.

Autonomous driving datasets often contain numerous object instances within a scene, with objects continuously entering, exiting, and interacting with each other. In line with this complexity, we have introduced our own version of the AP metric in this work. Our AP metric is designed to comprehensively compare all objects across every scene: encompassing those that newly enter, those that exist during the intermediate frames, and those that overlap with others.

First, we utilize the state-of-the-art object detection tool, YOLOv8 (Reis et al., 2024), to obtain the objects' trackings from the generated and ground-truth scenes. Detailed information about the tool and our configurations is reported in Appendix D.3. Next, we match objects in each generated-vs-ground-truth frame pair based on *spatial similarity* – calculating the intersection over union (IoU) score to determine the similarity in location between objects' bounding boxes. Our metric disregards object type and tracking IDs equivalence – assuming that objects close in location should naturally have the same type and IDs. Finally, we compute the average precision score following MS COCO protocol (Lin et al., 2015). Details are provided in Appendix D.4 and results are listed in Table 3. These results indicate that our Box2Video model is particularly adept at adhering to the specified conditions, especially when evaluated with a more lenient metric (i.e., a lower IoU threshold for the AP computation).

5 Conclusions

We have presented **Ctrl-V**, a novel model capable of generating controllable autonomous vehicle videos via bounding box trajectory conditioning. Our approach demonstrates that our **BBox Generator** technique can closely follow generation requirements for the first and last frames and produce a coherent bounding box track for intermediate frames. Moreover, our **Box2Video** network generates high-fidelity videos that strictly conform to the provided bounding boxes. Furthermore, our model accommodates both 2D and 3D bounding boxes and handles uninitialized objects appearing in the middle of the videos. Ctrl-V provides future researchers with an efficient way to simulate driving video data with flexible controllability in the form of bounding boxes. In addition, we further define an improved metric to evaluate bounding box conditioned video generation to account for objects that are not present in the first frame, and those that do not remain until the last frame. In Appendix G, we discuss potential future work for this project. With Ctrl-V and an improved metric for more accurate evaluation, we aim to establish a solid foundation for future research in controllable video generation.

References

Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. Bot-sort: Robust associations multi-pedestrian tracking. *arXiv preprint arXiv:2206.14651*, 2022.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets, 2023a.

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

Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2, 2020.

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *CoRR*, abs/1903.11027, 2019. URL `http://arxiv.org/abs/1903.11027`.

Duygu Ceylan, Chun-Hao Paul Huang, and Niloy J. Mitra. Pix2video: Video editing using image diffusion, 2023.

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.

Ernie Chu, Shuo-Yen Lin, and Jun-Cheng Chen. Video controlnet: Towards temporally consistent synthetic-to-real video translation using conditional image diffusion models, 2023.

Tobias Fischer, Thomas E. Huang, Jiangmiao Pang, Linlu Qiu, Haofeng Chen, Trevor Darrell, and Fisher Yu. Qdtrack: Quasi-dense similarity learning for appearance-only multiple object tracking, 2023.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation, 2023.

Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 961–970, 2015. doi: 10.1109/CVPR.2015.7298698.

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

Hsin-Ping Huang, Yu-Chuan Su, Deqing Sun, Lu Jiang, Xuhui Jia, Yukun Zhu, and Ming-Hsuan Yang. Fine-grained controllable video generation via object appearance and context, 2023.

Yash Jain, Anshul Nasery, Vibhav Vineet, and Harkirat Behl. Peekaboo: Interactive video generation via masked-diffusion, 2024.

Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLO, January 2023. URL `https://github.com/ultralytics/ultralytics`.

Johanna Karras, Aleksander Holynski, Ting-Chun Wang, and Ira Kemelmacher-Shlizerman. Dreampose: Fashion image-to-video synthesis via stable diffusion, 2023.

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

Seung Wook Kim, Jonah Philion, Antonio Torralba, and Sanja Fidler. Drivegan: Towards a controllable high-quality neural simulation, 2021.

Zoran Kotevski and Pece Mitrevski. Experimental comparison of psnr and ssim metrics for video quality estimation, 01 2010.

Pengxiang Li, Kai Chen, Zhili Liu, Ruiyuan Gao, Lanqing Hong, Guo Zhou, Hua Yao, Dit-Yan Yeung, Huchuan Lu, and Xu Jia. Trackdiffusion: Tracklet-conditioned video generation via diffusion models, 2024.

Xiaofan Li, Yifu Zhang, and Xiaoqing Ye. Drivingdiffusion: Layout-guided multi-view driving scene video generation with latent diffusion model. *arXiv preprint arXiv:2310.07771*, 2023.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

Jiachen Lu, Ze Huang, Jiahui Zhang, Zeyu Yang, and Li Zhang. Wovogen: World volume-aware diffusion for controllable multi-camera driving scene generation. *arXiv preprint arXiv:2312.02934*, 2023.

Wan-Duo Kurt Ma, J. P. Lewis, and W. Bastiaan Kleijn. Trailblazer: Trajectory control for diffusion-based video generation, 2024.

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

Eyal Molad, Eliahu Horwitz, Dani Valevski, Alex Rav Acha, Yossi Matias, Yael Pritch, Yaniv Leviathan, and Yedid Hoshen. Dreamix: Video diffusion models are general video editors, 2023.

Chong Mou, Mingdeng Cao, Xintao Wang, Zhaoyang Zhang, Ying Shan, and Jian Zhang. Revideo: Remake a video with motion and content control, 2024.

Jonah Philion, Xue Bin Peng, and Sanja Fidler. Trajeglish: Learning the language of driving scenarios. *arXiv preprint arXiv.2312.04535*, 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. *CoRR*, abs/2103.00020, 2021. URL `https://arxiv.org/abs/2103.00020`.

Dillon Reis, Jordan Kupec, Jacqueline Hong, and Ahmad Daoudi. Real-time flying object detection with yolov8, 2024.

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

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

Mahmood Sharif, Lujo Bauer, and Michael K. Reiter. On the suitability of l_p-norms for creating and preventing adversarial examples. *CoRR*, abs/1802.09653, 2018. URL `http://arxiv.org/abs/1802.09653`.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012. URL `http://arxiv.org/abs/1212.0402`.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges, 2019.

Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description, 2022.

Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. `https://github.com/huggingface/diffusers`, 2022.

Jiawei Wang, Yuchen Zhang, Jiaxin Zou, Yan Zeng, Guoqiang Wei, Liping Yuan, and Hang Li. Boximator: Generating rich and controllable motions for video synthesis, 2024.

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

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-driven world models for autonomous driving, 2023b.

Zhou Wang and Alan C. Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE Signal Processing Magazine*, 26(1):98–117, 2009. doi: 10.1109/MSP.2008.930649.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004a. doi: 10.1109/TIP.2003.819861.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004b. doi: 10.1109/TIP.2003.819861.

Yuqing Wen, Yucheng Zhao, Yingfei Liu, Fan Jia, Yanhui Wang, Chong Luo, Chi Zhang, Tiancai Wang, Xiaoyan Sun, and Xiangyu Zhang. Panacea: Panoramic and controllable video generation for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6902–6912, 2024.

Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, 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, 2023.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5288–5296, 2016. doi: 10.1109/CVPR.2016.571.

Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. *CoRR*, abs/1905.04804, 2019. URL `https://arxiv.org/abs/1905.04804`.

Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning, 2020.

Yan Zeng, Guoqiang Wei, Jiani Zheng, Jiaxin Zou, Yang Wei, Yuchen Zhang, and Hang Li. Make pixels dance: High-dynamic video generation, 2023.

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

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018.

Yabo Zhang, Yuxiang Wei, Dongsheng Jiang, Xiaopeng Zhang, Wangmeng Zuo, and Qi Tian. Controlvideo: Training-free controllable text-to-video generation, 2023b.

Yumeng Zhang, Shi Gong, Kaixin Xiong, Xiaoqing Ye, Xiao Tan, Fan Wang, Jizhou Huang, Hua Wu, and Haifeng Wang. Bevworld: A multimodal world model for autonomous driving via unified bev latent space. *arXiv preprint arXiv:2407.05679*, 2024.

Guosheng Zhao, Xiaofeng Wang, Zheng Zhu, Xinze Chen, Guan Huang, Xiaoyi Bao, and Xingang Wang. Drivedreamer-2: Llm-enhanced world models for diverse driving video generation. *arXiv preprint arXiv:2403.06845*, 2024.