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 like autonomous driving in particular it can be critical 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 in a safety critical setting. To achieve this, we 1) use a separate, specialized model to predict object bounding-box trajectories given the past and optionally future locations of bounding boxes, and 2) generate video conditioned on these high quality trajectory predictions. This formulation allows us to test the quality of different model components separately and together. To address the challenges of conditioning video generation on object trajectories in settings where objects may disappear and appear within a scene, we propose an approach based on rendering 2D or 3D boxes as videos. 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, BDD 100k, and nuScenes datasets validate the effectiveness of our approach in producing realistic and controllable video generation.

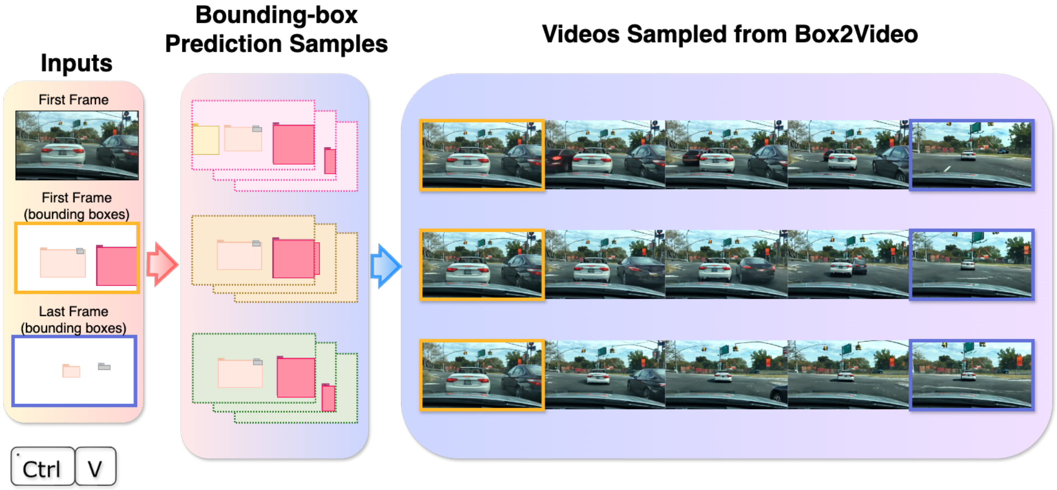

Figure 1: Overview of Ctrl-V's generation pipeline: *Inputs (left):* Our inputs include an initial frame, its corresponding bounding-box image and the final frame's bounding-box image. *Bounding-box generation samples (middle):* We illustrate three different sequences generated from our diffusion based bounding-box motion generation model. *Videos sampled from our Box2Video diffusion model (right):* Our Box2Video model conditions on the generated bounding-box videos to produce the final video clips.

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:

1. **2D-Bounding-Box and 3D-Bounding-Box Conditioning:** We condition on 2D or 3D Bounding boxes in order to provide a fine-grained control over the generated videos.
2. **Bounding-box Motion Generations with Diffusion:** We devise a novel diffusion based approach for generating 2D/3D bounding-box *trajectories* at the pixel-level (as 2D videos) based on their initial and final states, and the first frame.
3. **Uninitialized Object Generation:** Tracking boxes coordinates outside the current window (boxes that will eventually appear or that are leaving the view) is extremely difficult. With only the first frame, we cannot easily predict these outside-view coordinate movements. This is why, most coordinate-based bounding-box generations methods do not account for non-persisting or new bounding boxes (Wang et al., 2024). In this work, we propose a simple solution to this difficult problem: by utilizing on bounding boxes rendered at the pixel-level, we train our model to be sensitive to all bounding boxes, whether present from the first frame or appearing in the middle of the video.
4. **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 textprompted 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

In this section, we outline our simple yet robust method for creating video clips that strictly adhere to specified frame-level bounding-box guidelines. Ctrl-V encompasses two closely related sub-models: 1) BBox Generator (Section 3.3): a modified Video Latent Diffusion Model (VLDM) to generate bounding-box frames, images of only bounding boxes; 2) Box2Video (Section 3.4): our video generation model that allows precise frame-level conditioning. The overall model diagram is shown in Figure 2. In the following, we provide a brief overview of the preliminaries before delving into the details of Ctrl-V.

3.1 PRELIMINARIES

Stable Video Diffusion (SVD) (Blattmann et al., 2023b;a) is a cutting-edge video generation model that excels in converting still images into dynamic video sequences. The pretrained SVD model employed in this project has been pretrained using the Euler discrete noise scheduling method outlined by Karras et al. (2022). The objective of the SVD model is to generate a video sequence $\boldsymbol{f} = [\boldsymbol{f}^{(0)}, \ldots, \boldsymbol{f}^{(N)}]$ starting from an initial frame $\boldsymbol{f}^{(0)}$, where N represents the sequence length.

The SVD model utilizes an image autoencoder to encode and decode each frame, converting them between pixel space and latent space. Mathematically, this process is represented as $\mathcal{D}\big(\mathcal{E}(\boldsymbol{f})\big) = \mathcal{D}(\boldsymbol{z}) \approx \boldsymbol{f}$, where $\mathcal{E}$ is the image encoder, $\mathcal{D}$ is the image decoder and $\boldsymbol{z}$ is the latent space representation of frame $\boldsymbol{f}$. In SVD, the diffusion process occurs within the latent space. This process entails gradually introducing noise to the latent representations. Subsequently, the denoiser network is responsible for removing this added noise from the noisy latent representations. The denoiser network, trained under the Euler discrete noise scheduling schema, can be described by the formulation:

$$\mathbb{D}_\theta(\boldsymbol{z};\sigma_t) = \lambda_{\text{skip}}(\sigma_t)\boldsymbol{z} + \lambda_{\text{out}}(\sigma_t)\mathbb{U}_\theta\big(\lambda_{\text{in}}(\sigma_t)\boldsymbol{z}, \boldsymbol{z}^{(0)}_{\text{pad}}, \boldsymbol{c}^{(0)}; \lambda_{\text{noise}}(\sigma_t)\big) \tag{1}$$

Here λ_{skip}, λ_{out} and λ_{in} are scaling functions and σ_t is the computed noise level at time t; the mathematical definitions of these terms can be found in the work of Karras et al. (2022). $\mathbb{U}_\theta$ represents the 3D U-Net (Ronneberger et al., 2015), whose parameters θ are optimized for the denoising task during training.

To gain a clearer understanding of the U-Net inputs, we eliminate the scaling terms and reparameterize its input as follows:

$$\mathbb{U}_\theta\big(\hat{\boldsymbol{z}}_t, \boldsymbol{z}^{(0)}_{\text{pad}}, \boldsymbol{c}^{(0)}, t\big) \tag{2}$$

where $\hat{\boldsymbol{z}}_t \in \mathbb{R}^{N,C',H',W'}$ denotes the latent representation of frames corrupted by noise at the current noise level t; $\boldsymbol{z}^{(0)} \in \mathbb{R}^{1,C',H',W'}$ represents the latent representation of the initial frame, and $\boldsymbol{c}^{(0)}$ represents the encoding of the initial frame image token using CLIP (Radford et al., 2021). These parameters are introduced into the U-Net through distinct layers. Initially, at the input layer, $\boldsymbol{z}^{(0)}$ undergoes repeated padding along the frame's time dimension to match the dimensions of the $\hat{\boldsymbol{z}}_t$ tensor, namely $\boldsymbol{z}^{(0)}_{\text{pad}}$. Subsequently, the $\boldsymbol{z}^{(0)}_{\text{pad}}$ is concatenated with $\hat{\boldsymbol{z}}_t$ along the channel dimension and passed into the network's input layer. Meanwhile, the noise level indicator t is transformed into a time-step embedding vector and is subsequently fed into each U-Net block along with $\boldsymbol{c}^{(0)}$ via cross-attention conditioning mechanism (Rombach et al., 2022).

3.2 PREPROCESSING: BOUNDING-BOX RENDERING

Bounding boxes can be represented in many ways to be used as input to a network. While work such as Boximator Wang et al. (2024) represents bounding boxes as a Fourier transformed concatenated vector of its coordinate, ID and other information, we choose to create an information-rich image rendering of the bounding boxes as representation. By utilizing different pattern and colors, we are able to encode meta information – in addition to the location of the bounding boxes – such as track ID, object types, orientation about every bounding box in each frame; furthermore, by using an image representation, we take advantage of the SVD model's ability to interpret image data without the need to introduce customized adapter. We also implement a method to render trajectory frames, similar to the bounding-box frames, but without drawing out the entire box. These trajectory frames only contain mid-point location information. The procedure for creating bounding-box/trajectory plots is detailed in Appendix A.1. These bounding-box frames are ultimately sent to a off-the-shelf VAE image-encoder ($\mathcal{E}$) to be encoded (the encoder's performance is discussed in Appendix A.2).

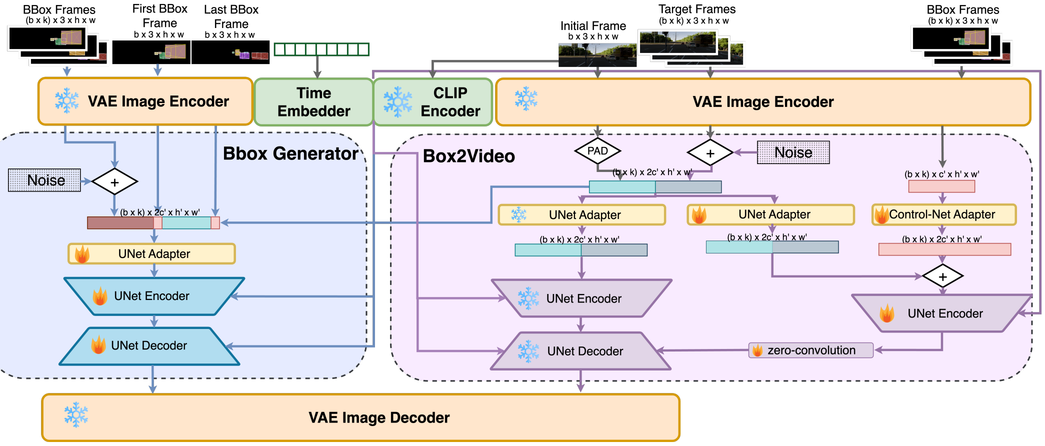

Figure 2: The diagram illustrates two components of **Ctrl-V**: (left; *blue*) the **BBox Generator** and (right; *purple*) **Box2Video** . In this diagram, *red vectors* represent the latents of bounding box frames, while *teal vectors* represent the latents of video frames. *Darker colored vectors* indicate latents that are contaminated by noise. The *trapezoidal shapes* in the diagram represent the UNet architecture, with the inverted trapezoids corresponding to the encoder and decoder modules of the UNet. For both models, we use a **frozen, off-the-shelf VAE** (represented by the *orange blocks*) to encode images into latent space and decode them back into pixel space. During training, (1) the **BBox Generator** (Sec. 3.3) learns to denoise the noisy bounding box frame latents (represented by the *dark red vector*), conditioned on the first and last bounding box frame latents (the *red vectors*) and the padded initial frame latent (the *teal vector*) and (2) the **Box2Video** (Sec. 3.4) denoises the target frame latents (represented by the *darker teal vector*) by conditioning on the initial frame's latent (*teal vector*; input to the UNet) and the bounding box frame latents (*red vectors*; input to the ControlNet).

3.3 CTRL-V: BBOX GENERATOR

Our BBox Generator (as shown on the left in Figure 2) aims to forecast object positions across intermediate frames, given the initial frame, initial bounding-box frame(s) and the final bounding-box/trajectory frame as references. Our approach involves adjusting the SVD network and leveraging it to generate bounding-box frames. Specifically, we modify the inputs of the U-Net, namely $\boldsymbol{z}_t$, $\boldsymbol{z}_{\text{pad}}^{(0)}$, and $\boldsymbol{c}^{(0)}$ (refer to Equation 2), to achieve this objective.

Initially, we pass the bounding-box frames through the image encoder ($\mathcal{E}$) to obtain their latent representations, denoted as $\mathcal{E}(\boldsymbol{f}_{\text{bbox}}) = \boldsymbol{b} \in \mathbb{R}^{N,C',H',W'}$. Furthermore, we feed the initial frame into both the image encoder and the CLIP encoder, resulting in $\boldsymbol{z}^{(0)} \in \mathbb{R}^{1\times C'\times H'\times W'}$ and $\boldsymbol{c}^{(0)}$ respectively.

Similar to the original SVD configuration, we replicate the padding procedure from $\boldsymbol{z}^{(0)} \in \mathbb{R}^{1 \times C' \times H' \times W'}$ to $\boldsymbol{z}_{\text{pad}}^{(0)} \in \mathbb{R}^{N \times C' \times H' \times W'}$. In this case, we replace the first and last entries of $\boldsymbol{z}_{\text{pad}}^{(0)}$ with $\boldsymbol{b}^{(0)}$ and $\boldsymbol{b}^{(N-1)}$ respectively. Additionally, depending on the desired number of initial bounding-box frames for conditioning, we experiment with substituting different quantities of $\boldsymbol{z}^{(0)}$ frames with various amounts of bounding-box frames. We use $\tilde{\boldsymbol{b}}$ to denote the modified tensor.

Furthermore, we replaced the $\hat{\boldsymbol{z}}_t$ in Equation 2 with $\hat{\boldsymbol{b}}_t$, where noise at level t is incorporated into $\boldsymbol{b}$. The primary objective of the denoiser network is to estimate the amount of noise introduced to $\boldsymbol{b}$.

Overall, the input to the BBox Generator's U-Net is represented as $\mathbb{U}_{\omega}^{\text{bbox}}\big(\hat{\boldsymbol{b}}_t, \tilde{\boldsymbol{b}}, \boldsymbol{c}^{(0)}, t\big)$.

3.4 CTRL-V: BOX2VIDEO

Given an initial frame and its corresponding sequence of bounding boxes, our aim is to produce a video clip in which the object positions align with the bounding-box sequence. To achieve this, we introduced Box2Video network, where we adapt ControlNet for the SVD framework (as shown on the right in Figure 2). Unlike previous works such as Boximator and TrackDiffusion(Wang et al., 2024; Li et al., 2024), this motion-control model is trained in a single-stage without requiring additional optimization criteria.

Our Box2Video comprises two main components: an SVD and a ControlNet. The input to the SVD model remains consistent with Equation 2. Regarding the ControlNet, we utilize the image encoder to encode the bounding-box frames $\mathcal{E}(\boldsymbol{f}_{\text{bbox}}) = \boldsymbol{b} \in \mathbb{R}^{N, C', H', W'}$. Additionally, we incorporate an adapter layer to preprocess the bounding-box latents before adding them to the frame latents. The inputs to the ControlNet are $\mathbb{C}_{\xi}\big(\boldsymbol{b}, \hat{\boldsymbol{z}}_t, \boldsymbol{z}_{\text{pad}}^{(0)}, \boldsymbol{c}^{(0)}, t\big)$.

The architecture of ControlNet closely resembles the contraction and mid-blocks of the SVD's U-Net architecture, a concept introduced by (Zhang et al., 2023a). Additionally, at the start of the training phase, ControlNet's weights are initialized to match those of their mirrored layers. Similar to U-Net, the CLIP encoded token of the initial frame ($\boldsymbol{c}^{(0)}$) and the time-step embedding are also injected into every block within the ControlNet. Finally, the output from each ControlNet block is added to the residual path within the SVD model after zero-convolution computation.

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

We evaluate our model on three driving datasets, and Figure 3 illustrates some generated samples produced by our model. All experiments are trained on the training sets and evaluated on the testing sets. To evaluate the quality of the generated videos, we randomly select 200 frames from the testing set of each dataset as the initial frames and generated videos based on them. For each dataset, the results presented in this section are based on analyses of the 200 generations.

Furthermore, each bounding-box generator discussed in the following section requires conditioning on ONE initial frame, ONE or THREE initial bounding-box frames and ONE final bounding-box/trajectory frame.

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 clips. 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 A.3.

4.2 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 1: 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 SVD baselines without fine-tuning (initial frame → video)
2. Fine-tuned SVD baselines (initial frame → video)
3. Teacher-forced Box2Video generation (initial frame and bounding-box frames → video)
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[1] is an exception; it evaluates the generation distribution

---

[1] The Fréchet Video Distance is very sensitivity to video configuration parameters—frame rate, clip duration (number of frames), and spatial resolution—poses a significant challenge. Specifically, discrepancies in these settings across existing studies render direct comparisons of FVD values problematic. Furthermore, the Fréchet distance metric exhibits high sensitivity to sample sizes. We suspect that certain datasets lack sufficient samples for the metric to achieve convergence, potentially leading to unreliable estimates.

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 1 and visualizations are available in Appendix C.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 B.1.

4.3 BBOX GENERATOR: QUANTITATIVE EVALUATION

	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 2: 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 D for implementation details on this baseline).

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 B.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 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 D. We present our findings in Table 2, and demonstrate examples of our bounding-box generations on each dataset in Appendix C.

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 2. 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.

4.4 BOX2VIDEO : MOTION CONTROL EVALUATION

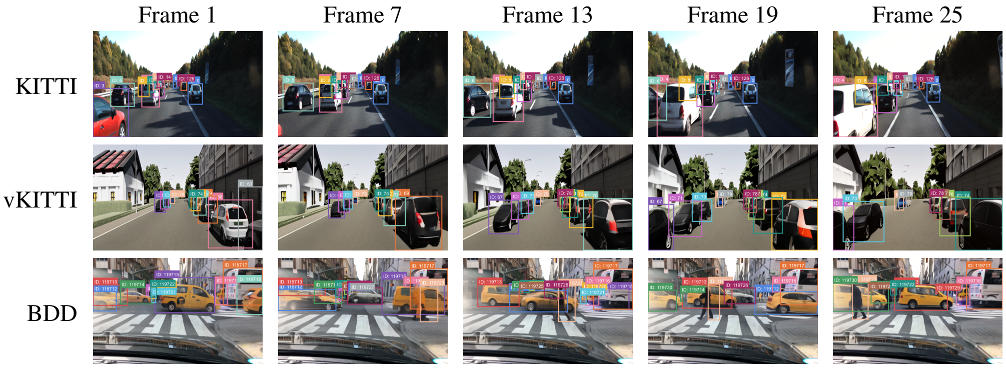

Figure 3: Illustrations of the generations conditioned on ground truth 3D bounding-boxes (2D for BDD) across various datasets. The 2D outlines of bounding-boxes are overlayed on top.

Our Box2Video is trained to control object motions through bounding-boxes using a teacher-forcing approach. This means that 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.

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 B.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 B.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 present **Ctrl-V**, a novel model capable of generating controllable autonomous vehicle videos. Our approach demonstrates that the BBox Generator model can closely follow generation require-

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[2] **(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. Longer videos are associated with decreased quality and lower detection rates, posing an additional challenge for our model (since it generates 56.25% more frames), yet it obtains higher precision than the other baselines.

ments for the first and last frames and produce a coherent bounding-box track for the intermediate frames. We also show that our Box2Video network generates high fidelity videos, strictly adhering to the given 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 E, 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.