DriveEditor: A Unified 3D Information-Guided Framework for Controllable Object Editing in Driving Scenes

Yiyuan Liang[1, 2*], **Zhiying Yan**[1, 2*], **Liqun Chen**[1, 2*], **Jiahuan Zhou**[3], **Luxin Yan**[1, 2], **Sheng Zhong**[1, 2], **Xu Zou**[1, 2†]

[1]Huazhong University of Science and Technology
[2]National Key Laboratory of Multispectral Information Intelligent Processing Technology
[3]Wangxuan Institute of Computer Technology, Peking University
{yvanliang, yanzhiying, chenliqun, yanluxin, zhongsheng, zoux}@hust.edu.cn, jiahuanzhou@pku.edu.cn

Abstract

Vision-centric autonomous driving systems require diverse data for robust training and evaluation, which can be augmented by manipulating object positions and appearances within existing scene captures. While recent advancements in diffusion models have shown promise in video editing, their application to object manipulation in driving scenarios remains challenging due to imprecise positional control and difficulties in preserving high-fidelity object appearances. To address these challenges in position and appearance control, we introduce DriveEditor, a diffusion-based framework for object editing in driving videos. DriveEditor offers a unified framework for comprehensive object editing operations, including repositioning, replacement, deletion, and insertion. These diverse manipulations are all achieved through a shared set of varying inputs, processed by identical position control and appearance maintenance modules. The position control module projects the given 3D bounding box while preserving depth information and hierarchically injects it into the diffusion process, enabling precise control over object position and orientation. The appearance maintenance module preserves consistent attributes with a single reference image by employing a three-tiered approach: low-level detail preservation, high-level semantic maintenance, and the integration of 3D priors from a novel view synthesis model. Extensive qualitative and quantitative evaluations on the nuScenes dataset demonstrate DriveEditor's exceptional fidelity and controllability in generating diverse driving scene edits, as well as its remarkable ability to facilitate downstream tasks.

Code — https://github.com/yvanliang/DriveEditor

Introduction

In autonomous driving, perception tasks (Liu et al. 2023; Wang et al. 2023a; Chen et al. 2023b; Zhou and Krähenbühl 2022) necessitate extensive data to construct robust models. To facilitate this, large-scale, open-source datasets (Caesar et al. 2020; Sun et al. 2020; Mao et al. 2021) for autonomous driving have been introduced, which contain thousands of driving scenes. While these datasets offer a rich repository of road-collected driving data, they exhibit a long-tailed distribution if not further processed. This imbalance leads to an overrepresentation of common driving scenes and an underrepresentation of rare yet critical events, such as unexpected obstacles or lane changes, posing challenges for training and evaluating perception tasks under these scenarios.

To mitigate this diversity challenge in driving data, recent approaches (Wang et al. 2023b; Ma et al. 2024; Gao et al. 2024) have leveraged the capabilities of Latent Diffusion Models (LDMs) (Rombach et al. 2022) to generate a variety of driving scenes. By leveraging Bird's-Eye-View (BEV) layouts to constrain scene structure, including lane lines and object positions, these methods generate diverse scenes aligned with semantic driving scenarios. However, they offer limited control over objects, operating at a semantic level without fine-grained constraints on detailed appearance. Despite using a subject bank for object control, SubjectDrive (Huang et al. 2024) still offers limited control over fine-grained object details. Beyond video generation, LDMs have also demonstrated remarkable capabilities in video editing. Leveraging natural language descriptions, some artworks (Couairon et al. 2024; Jin, Wang, and Pokorny 2024; Hu et al. 2023; Shin et al. 2024) achieve content alteration and style transfer within videos, producing remarkable outcomes. Nonetheless, these methods struggle to dictate precise visual appearances due to inherent linguistic ambiguities, and they also find it difficult to change object positions, which hinders object editing in driving contexts.

This paper introduces DriveEditor, a unified diffusion-based framework designed for the precise manipulation of objects within driving scenario videos. While seamlessly preserving the original background, DriveEditor enables a comprehensive set of object editing operations, including repositioning, replacement, deletion, and insertion, as illustrated in Figure 1. Notably, these diverse manipulations are all achieved via a shared set of varying inputs processed by two core modules: the position control module and the appearance maintenance module. The position control module projects each face of the 3D bounding box individually onto the image plane, preserving depth information. This projected data is then hierarchically injected into the diffusion process, enabling control over object position and orientation. To guarantee faithful visual appearance consistency

*These authors contributed equally.
†Corresponding author.

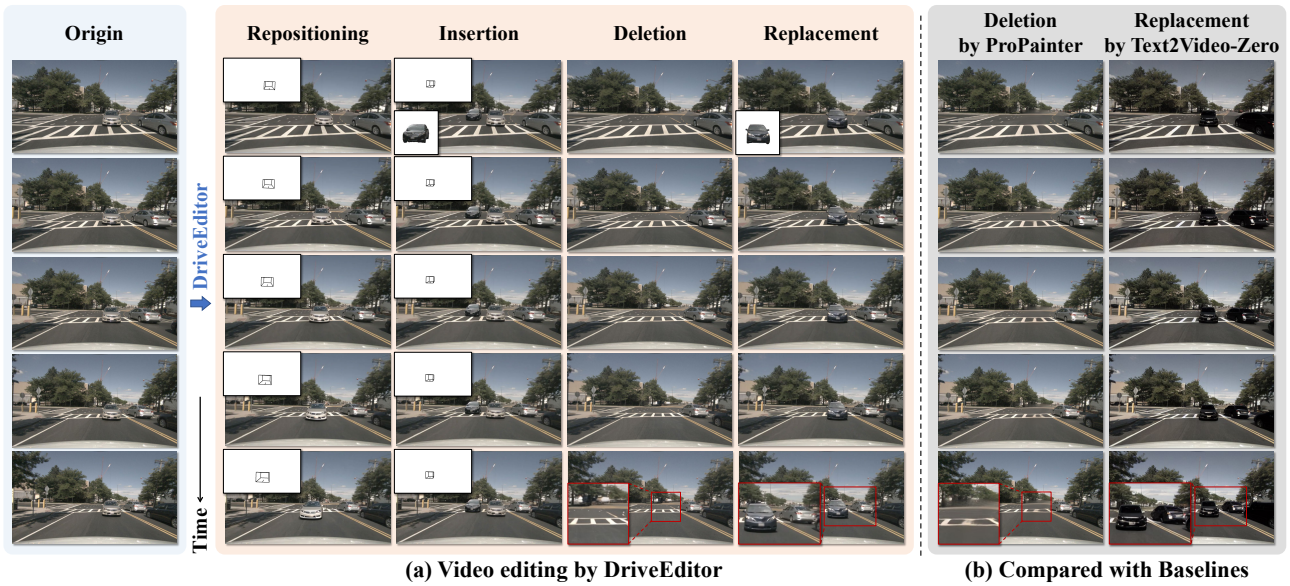

Figure 1: **Visualizations of the editing capability of DriveEditor and baselines. (a)** DriveEditor enables user-friendly *repositioning*, *insertion*, *replacement*, and *deletion* within a unified framework. It precisely controls an object's position and orientation based on the 3D bounding box (top left; required for repositioning and insertion tasks that alter object position), and maintains high-fidelity appearance attributes of the object from a single reference image (bottom left, required for insertion and replacement tasks that alter object appearance). **(b)** The *deletion* and *replacement* results compared with baselines. ProPainter's deletion results suffer from artifacts. Text2Video-Zero employs the text prompt "replace the champagne-colored car with dark gray van" to guide the replacement process. Yet, it produces unrealistic visual results and alters the appearance of other vehicles.

with the single reference image, we introduce three distinct levels of appearance control: low-level details preservation, high-level semantics maintenance, and incorporation of 3D priors derived from the novel view synthesis model.

DriveEditor comprehensively covers a wide range of object editing operations, significantly enriching the diversity of autonomous driving data. On one hand, it operates in a user-friendly manner, requiring only a single reference image and 3D bounding boxes per frame, which can be easily obtained through our interactive tool. On the other hand, it offers high controllability, enabling precise adjustments to object appearance, position, and orientation based on user instructions, or through automatic means.

Our main contributions can be summarized as follows:

- We propose DriveEditor, a diffusion-based framework for object editing in videos of driving scenarios.
- DriveEditor accomplishes four distinct editing tasks, leveraging shared inputs and a common network to enhance each task.
- We demonstrate DriveEditor's exceptional fidelity, controllability, and efficacy in facilitating downstream tasks through comprehensive experiments.

Related Works

Driving Scene Manipulation. The increasing demand for driving scenario data, coupled with the high cost of detailed manual annotation, highlights the need for efficient methods of acquiring driving scenes. Leveraging the powerful generative capabilities of LDMs, several approaches (Yang et al. 2023b; Li, Zhang, and Ye 2023; Wen et al. 2024) generate driving scenes with rich diversity. However, due to the complexity of driving scenarios and the multitude of targets, generated scenes often suffer from quality issues and lack fine-grained control over specific attributes or color variations of individual objects. Other NeRF-based (Mildenhall et al. 2021) artworks (Wei et al. 2024; Yang et al. 2023d) enable driving scene reconstruction, facilitating scene simulation and editing. These methods can produce high-fidelity scenes and support viewpoint changes. However, due to limited generative capabilities, they lack the flexibility for object-level editing, such as the removing static objects.

Diffusion Models for Video Editing. Video editing involves manipulating foreground, background, or style within videos, guided by a target prompt. Tune-A-Video (Wu et al. 2023) introduces one-shot tuning for video editing by adapting spatial self-attention layers in text-to-image diffusion models to their spatial-temporal counterparts. However, this approach incurs substantial fine-tuning costs. To address this, other methods (QI et al. 2023; Khandelwal 2023; Wang et al. 2024) enable zero-shot video editing by obtaining hidden features of the original video through inversion and preserving the information via attention map injection. To achieve finer-grained control, recent approaches leverage control modules like ControlNet (Zhang, Rao, and Agrawala 2023) to guide editing using multimodal conditions such as depth (Chen et al. 2023a), sketches (Yang et al. 2023c), images (Zhang et al. 2024) and keypoints (Jin, Wang, and Pokorny 2024), enabling diverse editing instructions.

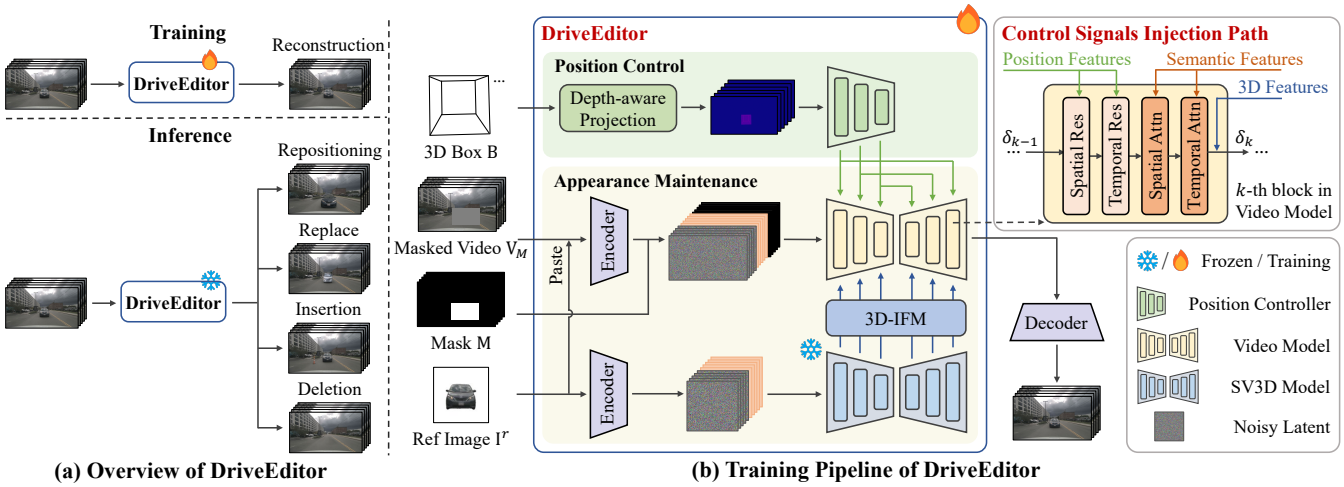

Figure 2: **(a)** High-level overview of DriveEditor. **(b)** Diagram of the training pipeline of DriveEditor. Three levels of appearance control are established based on the single reference image $\mathbf{I}^r$: low-level details preservation through a cut-and-paste approach, high-level semantics maintenance through cross-attention (omitted in the pipeline for brevity), and incorporation of 3D priors derived from the frozen SV3D U-Net. For position control, we perform a projection that preserves depth information, followed by the Pose Controller to extract multi-scale features. Control signals are injected through three distinct paths in block of the video model: position features into ResBlocks, semantic features via cross-attention, and 3D features added to block outputs.

Data Construction

DriveEditor is trained via a *reconstruction* task. In this task, we occlude a specific object in a video sequence. DriveEditor is trained to reconstruct the occluded object given a single image and ground-truth bounding boxes of that object. As no existing dataset directly caters to our editing requirements, we construct a dataset based on the widely-used nuScenes dataset (Caesar et al. 2020). This large-scale dataset for autonomous driving comprises 1,000 driving scenes, each scene consists of 20 seconds of videos captured from six camera views at 10 Hz.

Specifically, for each object that remains unobstructed within a 20-meter radius of the camera across N consecutive frames, those frames are concatenated to form an original video $\mathbf{V} \in \mathbb{R}^{N\times3\times H\times W}$. To obtain object images for editing, we employ SAM (Kirillov et al. 2023) to extract the object from each frame of $\mathbf{V}$. These images serve as both training data and an object bank for object replacement. To enhance the model's generalization ability, we randomly select the r-th image as the reference image, denoted as $\mathbf{I} \in \mathbb{R}^{3\times h\times w}$. We apply masks $\mathbf{M}$ to the object region in the video, creating a masked video $\mathbf{V}_m$. $\mathbf{B} = \left\{(x_i, y_i, z_i)^j\right\} \in \mathbb{R}^{N\times8\times3}$ represents the 3D bounding box of the object in the camera coordinate system. The camera's elevation and azimuth angles relative to the object, denoted by $\mathbf{e} \in \mathbb{R}^N$ and $\mathbf{a} \in \mathbb{R}^N$ respectively, can be calculated based on the center point of $\mathbf{B}$ and the camera's position in the world coordinate system. To enhance the model's ability to remove objects, we additionally apply random masks to object-free regions in the videos, generating inpainting training data.

This process yields a training dataset of 10,110 video clips, including a dedicated inpainting training subset of 2,000 clips, and a validation set of 800 clips.

Method

Brief Introduction of Video Diffusion Models

Stable Video Diffusion (SVD) (Blattmann et al. 2023) is a latent diffusion model specialized in high-quality image-to-video generation. Given a reference image $\mathbf{I} \in \mathbb{R}^{3\times H\times W}$, SVD can generate a video sequence $\mathbf{V} \in \mathbb{R}^{N\times3\times H\times W}$ consisting of N frames, initiated from $\mathbf{I}$. Following the EDM-preconditioning framework (Karras et al. 2022), SVD employs a learnable denoiser D_θ to parameterize the U-Net (Ronneberger, Fischer, and Brox 2015) network, iteratively estimating $\mathbf{z_0}$, the latent representation of $\mathbf{V_0}$, from $\mathbf{z_M} \sim \mathcal{N}(\mathbf{0}, {\sigma_{max}}^2)$ through denoising score matching:

$$\mathbb{E}_{\mathbf{z}_0\sim p_{data},(\sigma,\mathbf{n})\sim p(\sigma,\mathbf{n})}\left[\lambda_\sigma \left\|D_\theta\left(\mathbf{z}_0+\mathbf{n};\sigma,\mathbf{y},\mathbf{c}\right)-\mathbf{z}_0\right\|_2^2\right], \quad (1)$$

where $p(\sigma,\mathbf{n}) = p(\sigma)\mathcal{N}\left(\mathbf{n};\mathbf{0},\sigma^2\right)$, σ denotes the noise level. $\lambda_\sigma : \mathbb{R}_+ \to \mathbb{R}_+$ is a weighting function. Vector conditionings $\mathbf{y}$ (e.g., fps, motion rate), along with σ, are embedded and injected into the ResBlocks in the U-Net for guidance. The control signal $\mathbf{c}$ comprises tokens generated by a CLIP (Radford et al. 2021) image encoder from x^0, and latent representation produced by a VAE encoder. These are integrated into the diffusion model via cross-attention and channel-wise concatenation with frame latents, respectively.

Stable Video 3D (SV3D) (Voleti et al. 2024) represents a state-of-the-art latent video diffusion model capable of generating multi-view orbital videos around a 3D object. SV3D takes a single object image $\mathbf{I} \in \mathbb{R}^{3\times H\times W}$ as the initial viewpoint and generates an orbital video $\mathbf{V} \in \mathbb{R}^{N\times3\times H\times W}$ of the object by controlling the camera pose trajectory $\boldsymbol{\pi} = \left\{\left(e^i, a^i\right)\right\}_{i=1}^N \in \mathbb{R}^{N\times2}$ through its vector conditionings $\mathbf{y}$. Here, e represents the elevation angle, and a represents the azimuth angle.

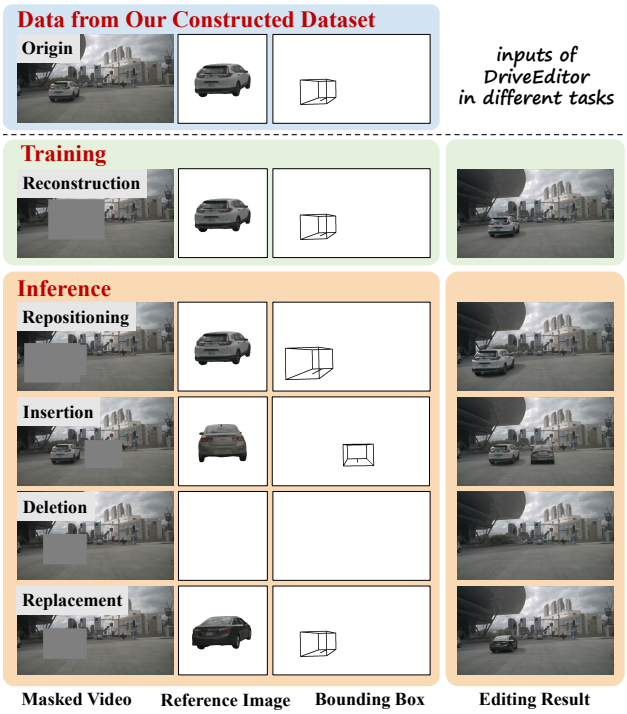

Figure 3: DriveEditor is trained to reconstruct occluded objects using inputs from our dataset. At inference time, it performs various editing tasks based on specific input prompts.

Unlocking Unified Editing

Here we explain the underlying principles that enable seamless repositioning, insertion, replacement, and deletion within DriveEditor framework through a unified pipeline.

During training, masks $\mathbf{M}$ are randomly sized and positioned, preventing the model from relying on any visual cues. DriveEditor reconstructs the masked area to recover the object, leveraging appearance cues from $\mathbf{I}$ and location information from $\mathbf{B}$. It is also trained on inpainting data to restore masked backgrounds, facilitating object deletion.

During inference, DriveEditor leverages varying inputs to facilitate a variety of editing tasks as illustrated in Figure 3.

Repositioning The object is segmented from the original video, obtaining $\mathbf{I}'$ as a reference to preserve its appearance. The desired position $\mathbf{B}'$, is controlled by the user. A mask is applied to the original video, encompassing the union of the projected areas defined by the original and desired positions, $\mathbf{B}$ and $\mathbf{B}'$, respectively, to derive $\mathbf{V}_m$.

Insertion. The desired object image and its target bounding box denoted as $\mathbf{I}'$ and $\mathbf{B}'$, respectively, are provided as user inputs. $\mathbf{V}_m$ is generated within the projected areas of $\mathbf{B}'$.

Deletion. As no target object exists, $\mathbf{I}'$ and $\mathbf{B}'$ are simply omitted. $\mathbf{V}_m$ is generated within the projected areas of $\mathbf{B}$.

Replacement. The user provides a reference image $\mathbf{I}'$ from the same viewpoint as in the video. If the viewpoints differ, SV3D model can be used for alignment. $\mathbf{V}_m$ is generated using the same strategy as in the deletion process.

Finally, the derived reference image, 3D bounding box, and masked video are input to DriveEditor for object editing.

Appearance Control

To control the appearance of objects using the reference image, we employ a three-level approach that facilitates details preservation, semantics maintenance, and 3D priors incorporation.

Low-Level Details Preservation. To preserve fine-grained details from the reference image, we adopt a straightforward yet efficient cut-and-paste approach. During training, the reference image $\mathbf{I}$ is directly pasted onto the r-th frame of $\mathbf{V}_m$, denoted as $\mathbf{V}_m^r$, as they share the same viewpoint. During inference, the provided reference image $\mathbf{I}'$ is pasted onto the frame in $\mathbf{V}_m$ with the closest viewpoint. We determine the size and location of the object in the r-th frame of the video based on the projection of $\mathbf{B}^r$. The object in $\mathbf{I}$ is then pasted onto the corresponding region in $\mathbf{V}_m^r$ to preserve fine-grained details of the object's appearance. The pasted video is concatenated with the masks $\mathbf{M}$ and derive c_{concat}, which is then injected into the diffusion process through channel-wise concatenation with the latent representation.

High-Level Semantics Maintenance. The pioneering work (Yang et al. 2023a) demonstrated the CLIP image encoder's ability to extract high-level semantic image information. We leverage this capability to preserve visual concepts by injecting CLIP features of the target object and origin background scene, denoted as c_{attn}, into the video U-Net via cross-attention. The c_{attn} is obtained as follows:

$$c_{attn} = [\text{CLIP}(\mathbf{V}_m^r), \text{CLIP}(\mathbf{I})], \tag{2}$$

where $\text{CLIP}(\cdot)$ is the CLIP image encoder, $[\cdot]$ represents channel-wise concatenation operation. For the deletion task, we employ a trainable null embedding as a replacement for the CLIP image features to guide object removal.

3D Prior Incorporation. While relying on a single reference image simplifies the usage of DriveEditor, it limits access to multi-view object information. To address this limitation, we leverage the SV3D model, which is capable of generating novel views of objects. Sharing the same architecture and latent space with SVD, we seamlessly integrate its intermediate features to guide the appearance of objects during video editing.

Starting from the reference image as an initial view, we employ a pretrained SV3D model to generate M novel object views. Intermediate features extracted from each block of the SV3D model are then injected into the corresponding block of the video model as strong object priors. To address the SV3D model's limitations in trajectory control, especially for fine-grained view changes and non-full-circle trajectories, we use a carefully crafted fixed azimuth angle $\tilde{\mathbf{a}} \in \mathbb{R}^M$ during generation (details in supplementary materials). Given the relatively consistent elevation angle across the video, we set $\tilde{\mathbf{e}} \in \mathbb{R}^M$ to its mean value. For the i-th video frame, we identify the corresponding j-th frame in the generated novel views based on its azimuth angle $\mathbf{a}^i$:

$$j = \underset{i \in \{0,1,\ldots,N-1\}}{\arg\min} \left| \tilde{\mathbf{a}}^j - \mathbf{a}^i \right|. \tag{3}$$

To address discrepancies in scale and position between the object in the video and the reference image, we introduce a 3D Information Fusion Module (3D-IFM) for feature

alignment and fusion. Specifically, $\boldsymbol{\delta}_k^i$, $\boldsymbol{\theta}_k^i$ indicates the feature representations of the k-th block for the i-th frame extracted from the U-Net of the video model, SV3D model, respectively. We firstly transform $\boldsymbol{\theta}_k^j$ to align with the scale and position of the object in the video, as indicated by the projection of $\mathbf{B}^i$. Similar to ControlNet (Zhang, Rao, and Agrawala 2023), we then employ zero convolution layers to prevent the pretrained video model from being corrupted by noise during the initial training phase. The 3D-IFM serves as a layer-wise intermediary between the SV3D model and the video model, facilitating hierarchical refinement of the representation. The fusion process can be formulated as:

$$\boldsymbol{\delta}_k^i = \boldsymbol{\delta}_k^i + \mathbf{M}^i \times \mathcal{Z}\left(\mathcal{T}_{\mathbf{B}^i}\left(\boldsymbol{\theta}_k^j\right)\right), \tag{4}$$

where $\mathcal{T}_{\mathbf{B}^i}$ represents the transformation and resizing operation based on $\mathbf{B}^i$, $\mathcal{Z}$ denotes a zero convolution layer with weights and biases initialized to $\mathbf{0}$, and $k \in \{0, 1, \ldots, 2L-1\}$ refers to the $2L$ blocks in the U-Net, with the first L blocks in the encoder and the last L in the decoder.

Position Control

To precisely control object position and orientation, we introduce the Depth-aware Position Controller, comprising Depth-aware Projection and Position Controller.

In the Depth-aware Projection module, each face of the 3D bounding box $\mathbf{B}$ is processed individually. For each face, P points are interpolated from its four vertices. These face and edge points are projected onto the 2D image plane using camera intrinsics, with their corresponding depth values (z) assigned as pixel intensities. This process generates a six-channel depth-aware pose image $\mathbf{I}_P$, preserving depth information without inter-face occlusion.

$\mathbf{I}_P$ is fed into a ResNet-like Position Controller to extract multi-scale positional control features. This Position Controller contains L blocks, aligning with the hierarchical structure of the video U-Net's encoder and decoder. These features are injected into the spatial and temporal ResBlocks of the video U-Net using adapters. Let $\mathbf{f}_k$ denote the feature at the k-th block of the Object Position Controller, and $\mathbf{g}_k \in \mathbb{R}^{N \times C_k \times H_k \times W_k}$ and $\mathbf{v}_k \in \mathbb{R}^{C_k \times N \times H_k \times W_k}$ represent the spatial and temporal ResBlocks of the k-th block in video U-Net, respectively. The injection of position control features can be represented by:

$$\begin{aligned} \mathbf{g}_k &= \mathbf{g}_k + \text{Adapter2d}_k(\mathbf{f}_l), \\ \mathbf{v}_k &= \mathbf{v}_k + \text{Adapter3d}_k(\text{Permute}(\mathbf{f}_l)), \end{aligned} \tag{5}$$

where $l = k$ if $l < L$ else $2L - k - 1$, AdapterNd$(\cdot)$ represents the k-th adapter composed of SiLU, LayerNorm, and ConvNd operations. Permute$(\cdot)$ denotes the permutation of frame and channel dimensions. By aligning spatially with video model features and ensuring temporal coherence via temporal injection, it facilitating precise position control.

To prevent object location information leakage through the 3D-IFM, we adopt a two-stage training strategy. In the first stage, by training the video model and Position Controller without 3D information fusion, we restrict the model to learning object positions solely from the input, which effectively facilitates the training of the Position Controller. In the second stage, we reintroduce the 3D-IFM and the frozen SV3D model, allowing for the training of the video model alongside all proposed modules.

Experiments

Experimental Setups

Baselines. While our work focuses on constructing a unified framework for versatile editing tasks, there is limited related work available, and even fewer methods allow for fair comparison. For *replacement*, we compare our method to Tune-A-Video (T2V) (Wu et al. 2023) and Text2Video-Zero (T2V) (Khachatryan et al. 2023), which excel at video context editing via text prompts. For *deletion*, we compare against ProPainter (Zhou et al. 2023) and Stable Diffusion (Rombach et al. 2022) Inpainting (SD). To the best of our knowledge, existing video editing methods lack the capability for fine-grained 3D object position manipulation, as they are primarily designed for broader editing tasks. We thus provide quantitative and qualitative results of DriveEditor on *insertion* and *replacement* tasks.

Quality Metrics. We utilize frame-wise FID (Heusel et al. 2017) and FVD (Unterthiner et al. 2019) metrics to evaluate both image quality and temporal consistency. Additionally, we apply CLIP-I metrics to assess the semantic alignment between the reference image and the object region in the edited video. Given the availability of ground truth (GT) videos in the reconstruction task, we further employ PSNR and LPIPS (Zhang et al. 2018) to measure pixel-level fidelity, and perceptual differences, respectively.

Position Controllability Metrics. We assess the alignment between the edited object and the ground truth 3D bounding box using a pretrained StreamPETR (Wang et al. 2023a) model, which is a state-of-the-art multi-view 3D object detector. We use mean Recall (mRecall), mean Average Translation Error (mATE), and mean Average Orientation Error (mAOE) to quantify positional accuracy of editing results. To ensure a fair evaluation of object editing capabilities, our assessment focuses exclusively on the edited objects.

Model Setup. DriveEditor generates high-resolution videos of 576×1024 pixels with a length of 10 frames. Additional details can be found in the supplementary materials.

Main Results

Editing Quality. We conduct object editing on the validation set and present both quantitative metrics of editing quality in Table 1 and qualitative visualizations in Figure 4. For deletion, ProPainter exhibits severe artifacts, resulting in a lack of realism as indicated by an FID of 36.13. While SD achieves higher image realism with an FID of 30.89, it suffers from inconsistent frame-to-frame results, leading to a higher FVD of 457. In contrast, DriveEditor generates temporally coherent high-quality results, with FID and FVD scores of 30.17 and 228, respectively. For replacement, T2V achieves modifications to object colors but loses detailed realism. TAV alters the overall style and introduces object deformations, resulting in significantly higher FID and FVD scores. In contrast, DriveEditor preserves high-fidelity object details from the reference image, achieving much lower

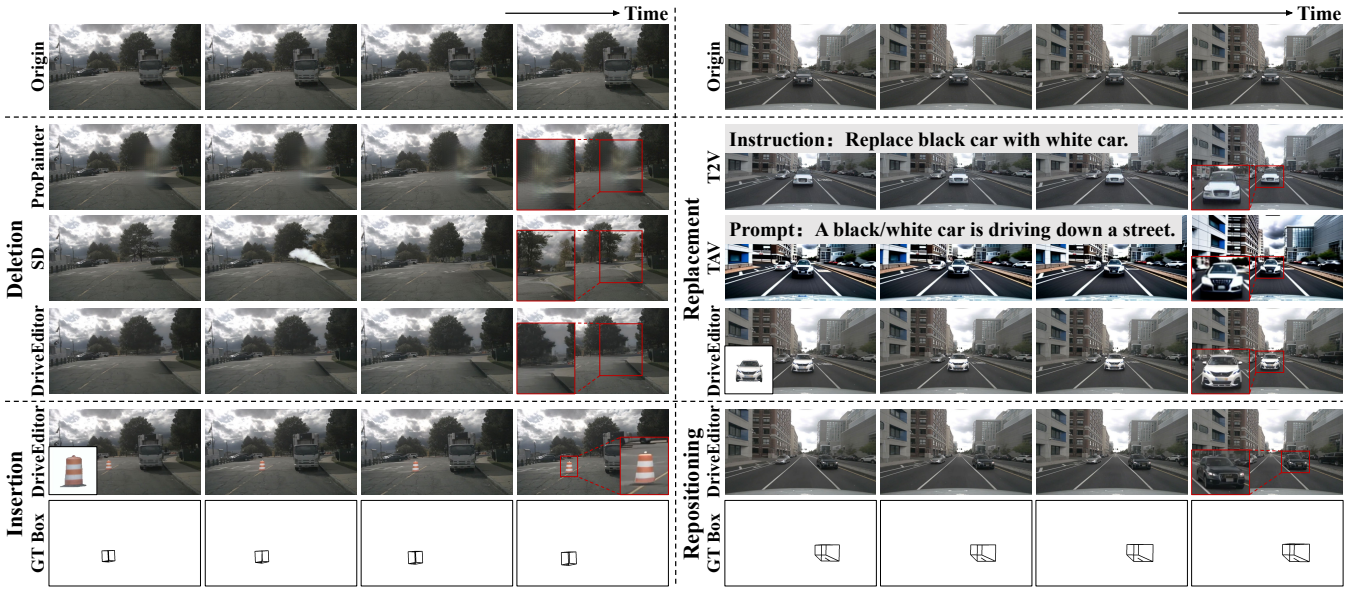

Figure 4: **Top row:** Original videos. **Middle left:** Qualitative comparison on the *deletion* task. ProPainter suffers from artifacts, while SD lacks temporal consistency. DriveEditor effectively generates plausible occluded regions. **Middle Right:** Qualitative comparison on the *replacement* task. T2V loses realism, for instance, the roof color remains unchanged. TAV alters the overall style of the video and leads to object deformations. In contrast, DriveEditor maintains high-fidelity object details from the reference image. **Bottom Left:** Visualization of object *insertion* using DriveEditor. It enables precise control over object insertion position while maintaining appearance from the reference image. **Bottom Right:** Visualization of object *repositioning* using DriveEditor. The object is accurately repositioned to align with the GT bounding box while preserving its original appearance.

Tasks	Methods	FID ↓	FVD ↓
Deletion	ProPainter	36.13	365
	SD	30.89	457
	DriveEditor	**30.17**	**228**
Replacement	T2V	14.22	151
	TAV	58.25	783
	DriveEditor	**11.83**	**39**

Table 1: Comparison of generation fidelity on replacement and deletion tasks. The best results are in **bold**. DriveEditor outperforms all baselines in terms of both single-frame generation quality and temporal consistency by a large margin.

Tasks	mRecall ↑	mATE ↓	mAOE ↓	CLIP-I ↑
Repositioning	0.93	0.68	0.044	77.73
Insertion	0.94	0.66	0.043	77.91
Replacement	0.94	0.74	0.044	77.86
Oracle	0.99	0.42	0.037	78.06

Table 2: Evaluation of DriveEditor's positional control via 3D object detection and the semantic alignment between the reference image and the edited object. The performance on the edited results shows only slight degradation compared to the unedited data *oracle*, indicating robust positional control and semantic alignment across all tasks.

FID and FVD scores of 11.83 and 39, respectively. Visualization results also demonstrate that DriveEditor can produce visually harmonious editing results, while highly preserving the object's appearance attributes.

Position Control Ability. Given that the target object is absent in the deletion task, we employ 3D object detection on the remaining three tasks. Table 2 reveals a slight performance drop in the edited data compared to the unedited oracle data. In terms of translation error, repositioning, insertion, and replacement incur errors of 0.26 meters, 0.24 meters, and 0.32 meters, respectively, with an average translation error of only 0.27 meters. Replacement exhibits a more pronounced performance degradation compared to the other two tasks, primarily attributed to size discrepancies between the replaced and original objects. All three tasks demonstrate comparable orientation errors, approximately 0.007 radians higher than the oracle. The visualization results in Figure 4 also corroborate the accurate alignment of object positions with GT bounding boxes. These results validate DriveEditor's proficiency in controlling both position and orientation.

Ablation Study

We conduct a comprehensive ablation study on the reconstruction task, evaluating both editing quality and position control ability, as presented in Table 3.

As a *baseline*, we ablate the 3D information fusion and position control modules, forcing the model to rely solely on the single cut-and-paste frame and mask positions to infer

Settings			Quality				Position		
PC	DP	3D	FID ↓	FVD ↓	PSNR ↑	LPIPS ↓	mRecall ↑	mATE ↓	mAOE ↓
-	-	-	6.56	24.95	29.83	0.052	0.91	0.74	0.197
✓	-	-	6.59	25.29	29.82	0.052	0.92	0.71	0.148
✓	✓	-	6.49	22.11	29.98	0.049	0.92	0.64	0.061
✓	✓	✓	6.36	18.82	30.10	0.047	0.95	0.59	0.043

Table 3: Ablation study on the reconstruction task verifies the effectiveness of our proposed modules.

Figure 5: Effectiveness of our proposed modules in controlling the position and orientation of objects. GT bounding boxes are outlined in black within the images.

Figure 6: Effectiveness of the 3D information fusion module in preserving object appearance and preventing distortion.

object position. This results in a significant translation error of 0.74 meters and an orientation error of 0.197 radians.

To access our Pose Controller (*PC*), we project 3D bounding box edges onto image plane and use it to encode non-depth positional features. While this approach yields improvements in positional metrics, it also introduces subtle degradations in quality compared to the baseline, potentially due to interference from the injected positional information.

We then introduce Depth-aware Projection (*DP*), which yields a significant improvement of 0.087 radians in mAOE and 0.06 meters in mATE. The remarkable improvement in quality metrics compared to the baseline further underscores the importance of depth information.

Finally, we incorporate the 3D priors through 3D Information Fusion Module (*3D*), to develop DriveEditor. This integration significantly enhances video quality and consistency, as evidenced by FID, FVD, and PSNR scores of 6.36, 18.82, and 30.10, respectively. Notably, mRecall improves substantially from 0.92 to 0.95, indicating a reduction in unexpected object appearance distortion, as visually confirmed in Figure 6. Concurrently, position control is improved, with mATE and mAOE decreasing by 0.05 meters and 0.018 radians, respectively, as visualized in Figure 5.

Real	Gen 50% Repo.	Gen 50% Repl.	mAP ↑	mATE ↓	mAOE ↓	NDS ↑
✓	-	-	0.480	0.615	0.378	0.569
✓	✓	-	0.482	0.600	0.340	0.577
✓	-	✓	0.487	0.588	0.375	0.576
✓	✓	✓	0.488	0.582	0.338	0.581

Table 4: Comparison of augmented datasets generated via repositioning (*Repo.*) and replacement (*Repl.*), each derive from the same 50% subset of the nuScenes training set. The augmented data significantly benefits downstream tasks.

Training Support for 3D Object Detection

Since not all data contains suitable objects for editing, we select 50% of the nuScenes training data to generate two augmented datasets through repositioning and replacement, respectively, to enhance StreamPETR model training. As shown in Table 4, repositioning expands the object viewpoint distribution, resulting in a 0.038 radian reduction in orientation error compared to non-augmented data. Replacement introduces diverse objects at identical positions, leading to a decrease in translation error. Combining both augmented datasets, despite being derived from the same original data, significantly reduces translation and orientation errors, achieving a NuScenes Detection Score (NDS) of 0.581. This demonstrates that DriveEditor can significantly enrich the diversity of data, thereby facilitating downstream tasks.

Conclusion

This paper introduces DriveEditor, a diffusion-based unified framework that allows users to easily reposition, insert, replace, and delete objects within driving scenario videos. Additionally, it supports iterative editing by conditioning on the last frame of the previous video, allowing for the editing of long videos. With our proposed position control modules, DriveEditor achieves alignment of edited results with 3D bounding boxes, enabling highly controllable position manipulation. Besides, DriveEditor preserves appearance through different feature levels, enabling high-fidelity object appearance control based solely on a single reference image. Through extensive experiments, DriveEditor has demonstrated exceptional fidelity and controllability in object editing within driving scenarios.

Acknowledgements

This work was supported by grants from the National Natural Science Foundation of China (62176100, 62301228, 62376011). The computation is completed in the HPC Platform of Huazhong University of Science and Technology.

References

Blattmann, A.; Dockhorn, T.; Kulal, S.; Mendelevitch, D.; Kilian, M.; Lorenz, D.; Levi, Y.; English, Z.; Voleti, V.; Letts, A.; Jampani, V.; and Rombach, R. 2023. Stable Video Diffusion: Scaling Latent Video Diffusion Models to Large Datasets. arXiv:2311.15127.

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuScenes: A Multimodal Dataset for Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11621–11631.

Chen, Y.; Dong, X.; Gan, T.; Zhou, C.; Yang, M.; and Guo, Q. 2023a. EVE: Efficient zero-shot text-based Video Editing with Depth Map Guidance and Temporal Consistency Constraints. arXiv:2308.10648.

Chen, Y.; Yu, Z.; Chen, Y.; Lan, S.; Anandkumar, A.; Jia, J.; and Alvarez, J. M. 2023b. FocalFormer3D: Focusing on Hard Instance for 3D Object Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 8394–8405.

Couairon, P.; Rambour, C.; Haugeard, J.-E.; and Thome, N. 2024. VidEdit: Zero-Shot and Spatially Aware Text-Driven Video Editing. arXiv:2306.08707.

Gao, R.; Chen, K.; Xie, E.; Hong, L.; Li, Z.; Yeung, D.-Y.; and Xu, Q. 2024. MagicDrive: Street View Generation with Diverse 3D Geometry Control. arXiv:2310.02601.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, 6629–6640.

Hu, M.; Jiang, K.; Nie, Z.; Zhou, J.; and Wang, Z. 2023. Store and fetch immediately: Everything is all you need for space-time video super-resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 863–871.

Huang, B.; Wen, Y.; Zhao, Y.; Hu, Y.; Liu, Y.; Jia, F.; Mao, W.; Wang, T.; Zhang, C.; Chen, C. W.; Chen, Z.; and Zhang, X. 2024. SubjectDrive: Scaling Generative Data in Autonomous Driving via Subject Control. arXiv:2403.19438.

Jin, S.; Wang, R.; and Pokorny, F. T. 2024. RealCraft: Attention Control as A Tool for Zero-Shot Consistent Video Editing. arXiv:2312.12635.

Karras, T.; Aittala, M.; Aila, T.; and Laine, S. 2022. Elucidating the Design Space of Diffusion-Based Generative Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 26565–26577.

Khachatryan, L.; Movsisyan, A.; Tadevosyan, V.; Henschel, R.; Wang, Z.; Navasardyan, S.; and Shi, H. 2023. Text2Video-Zero: Text-to-Image Diffusion Models are Zero-Shot Video Generators. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 15954–15964.

Khandelwal, A. 2023. InFusion: Inject and Attention Fusion for Multi Concept Zero-Shot Text-Based Video Editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 3017–3026.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; Dollar, P.; and Girshick, R. 2023. Segment Anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 4015–4026.

Li, X.; Zhang, Y.; and Ye, X. 2023. DrivingDiffusion: Layout-Guided multi-view driving scene video generation with latent diffusion model. arXiv:2310.07771.

Liu, Z.; Tang, H.; Amini, A.; Yang, X.; Mao, H.; Rus, D. L.; and Han, S. 2023. BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2774–2781.

Ma, E.; Zhou, L.; Tang, T.; Zhang, Z.; Han, D.; Jiang, J.; Zhan, K.; Jia, P.; Lang, X.; Sun, H.; Lin, D.; and Yu, K. 2024. Unleashing Generalization of End-to-End Autonomous Driving with Controllable Long Video Generation. arXiv:2406.01349.

Mao, J.; Niu, M.; Jiang, C.; Liang, H.; Chen, J.; Liang, X.; Li, Y.; Ye, C.; Zhang, W.; Li, Z.; Yu, J.; Xu, H.; and Xu, C. 2021. One Million Scenes for Autonomous Driving: ONCE Dataset. arXiv:2106.11037.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. NeRF: representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1): 99–106.

QI, C.; Cun, X.; Zhang, Y.; Lei, C.; Wang, X.; Shan, Y.; and Chen, Q. 2023. FateZero: Fusing Attentions for Zero-shot Text-based Video Editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 15932–15942.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 8748–8763.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-Resolution Image Synthesis With Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10684–10695.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 234–241.

Shin, C.; Kim, H.; Lee, C. H.; Lee, S.-g.; and Yoon, S. 2024. Edit-A-Video: Single Video Editing with Object-

Aware Consistency. In *Proceedings of the 15th Asian Conference on Machine Learning (ACML)*, 1215–1230.

Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; Vasudevan, V.; Han, W.; Ngiam, J.; Zhao, H.; Timofeev, A.; Ettinger, S.; Krivokon, M.; Gao, A.; Joshi, A.; Zhang, Y.; Shlens, J.; Chen, Z.; and Anguelov, D. 2020. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2446–2454.

Unterthiner, T.; van Steenkiste, S.; Kurach, K.; Marinier, R.; Michalski, M.; and Gelly, S. 2019. Towards Accurate Generative Models of Video: A New Metric & Challenges. arXiv:1812.01717.

Voleti, V.; Yao, C.-H.; Boss, M.; Letts, A.; Pankratz, D.; Tochilkin, D.; Laforte, C.; Rombach, R.; and Jampani, V. 2024. SV3D: Novel Multi-view Synthesis and 3D Generation from a Single Image using Latent Video Diffusion. arXiv:2403.12008.

Wang, S.; Liu, Y.; Wang, T.; Li, Y.; and Zhang, X. 2023a. Exploring Object-Centric Temporal Modeling for Efficient Multi-View 3D Object Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 3621–3631.

Wang, W.; Jiang, Y.; Xie, K.; Liu, Z.; Chen, H.; Cao, Y.; Wang, X.; and Shen, C. 2024. Zero-Shot Video Editing Using Off-The-Shelf Image Diffusion Models. arXiv:2303.17599.

Wang, X.; Zhu, Z.; Huang, G.; Chen, X.; Zhu, J.; and Lu, J. 2023b. DriveDreamer: Towards Real-world-driven World Models for Autonomous Driving. arXiv:2309.09777.

Wei, Y.; Wang, Z.; Lu, Y.; Xu, C.; Liu, C.; Zhao, H.; Chen, S.; and Wang, Y. 2024. Editable Scene Simulation for Autonomous Driving via Collaborative LLM-Agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15077–15087.

Wen, Y.; Zhao, Y.; Liu, Y.; Jia, F.; Wang, Y.; Luo, C.; Zhang, C.; Wang, T.; Sun, X.; and Zhang, X. 2024. Panacea: Panoramic and Controllable Video Generation for Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6902–6912.

Wu, J. Z.; Ge, Y.; Wang, X.; Lei, S. W.; Gu, Y.; Shi, Y.; Hsu, W.; Shan, Y.; Qie, X.; and Shou, M. Z. 2023. Tune-A-Video: One-Shot Tuning of Image Diffusion Models for Text-to-Video Generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 7623–7633.

Yang, B.; Gu, S.; Zhang, B.; Zhang, T.; Chen, X.; Sun, X.; Chen, D.; and Wen, F. 2023a. Paint by Example: Exemplar-Based Image Editing With Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 18381–18391.

Yang, K.; Ma, E.; Peng, J.; Guo, Q.; Lin, D.; and Yu, K. 2023b. BEVControl: Accurately Controlling Street-view Elements with Multi-perspective Consistency via BEV Sketch Layout. arXiv:2308.01661.

Yang, S.; Zhou, Y.; Liu, Z.; and Loy, C. C. 2023c. Rerender A Video: Zero-Shot Text-Guided Video-to-Video Translation. In *SIGGRAPH Asia 2023 Conference Papers (SA '23)*, 95.

Yang, Z.; Chen, Y.; Wang, J.; Manivasagam, S.; Ma, W.-C.; Yang, A. J.; and Urtasun, R. 2023d. UniSim: A Neural Closed-Loop Sensor Simulator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1389–1399.

Zhang, D. J.; Li, D.; Le, H.; Shou, M. Z.; Xiong, C.; and Sahoo, D. 2024. Moonshot: Towards Controllable Video Generation and Editing with Multimodal Conditions. arXiv:2401.01827.

Zhang, L.; Rao, A.; and Agrawala, M. 2023. Adding Conditional Control to Text-to-Image Diffusion Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 3836–3847.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 586–595.

Zhou, B.; and Krähenbühl, P. 2022. Cross-View Transformers for Real-Time Map-View Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13760–13769.

Zhou, S.; Li, C.; Chan, K. C.; and Loy, C. C. 2023. ProPainter: Improving Propagation and Transformer for Video Inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 10477–10486.