

# DriveDreamer4D: World Models Are Effective Data Machines for 4D Driving Scene Representation

Guosheng Zhao<sup>\*1,2</sup> Chaojun Ni<sup>\*1,4</sup> Xiaofeng Wang<sup>\*1,2</sup> Zheng Zhu<sup>\*1✉</sup>  
 Xueyang Zhang<sup>3</sup> Yida Wang<sup>3</sup> Guan Huang<sup>1</sup> Xinze Chen<sup>1</sup> Boyuan Wang<sup>1,2</sup>  
 Youyi Zhang<sup>5</sup> Wenjun Mei<sup>4</sup> Xingang Wang<sup>2✉</sup>  
<sup>1</sup>GigaAI <sup>2</sup>Institute of Automation, Chinese Academy of Sciences  
<sup>3</sup> Li Auto Inc. <sup>4</sup>Peking University <sup>5</sup> Technical University o f Munich

Project Page: <https://drivedreamer4d.github.io>



Figure 1. Previous 4D Gaussian Splatting methods (e.g., PVG [9], S<sup>3</sup>Gaussian [28], Deformable-GS [74]) face challenges in rendering novel trajectories, such as lane change. *DriveDreamer4D* addresses this by enhancing 4D driving scene representation via integrating priors from world models, significantly improving rendering quality under complex scenarios and novel trajectory viewpoints.

## Abstract

*Closed-loop simulation is essential for advancing end-to-end autonomous driving systems. Contemporary sensor simulation methods, such as NeRF and 3DGS, rely predominantly on conditions closely aligned with training data distributions, which are largely confined to forward-driving scenarios. Consequently, these methods face limitations when rendering complex maneuvers (e.g., lane change, acceleration, deceleration). Recent advancements in autonomous-driving world models have demonstrated the potential to generate diverse driving videos. However, these approaches remain constrained to 2D video generation, inherently lacking the spatiotemporal coherence required to capture intricacies of dynamic driving environments. In this paper, we introduce DriveDreamer4D, which enhances 4D driving scene representation leveraging world*

*model priors. Specifically, we utilize the world model as a data machine to synthesize novel trajectory videos, where structured conditions are explicitly leveraged to control the spatial-temporal consistency of traffic elements. Besides, the cousin data training strategy is proposed to facilitate merging real and synthetic data for optimizing 4DGS. To our knowledge, DriveDreamer4D is the first to utilize video generation models for improving 4D reconstruction in driving scenarios. Experimental results reveal that DriveDreamer4D significantly enhances generation quality under novel trajectory views, achieving a relative improvement in FID by 32.1%, 46.4%, and 16.3% compared to PVG, S<sup>3</sup>Gaussian, and Deformable-GS. Moreover, DriveDreamer4D markedly enhances the spatiotemporal coherence of driving agents, which is verified by a comprehensive user study and the relative increases of 22.6%, 43.5%, and 15.6% in the NTA-IoU metric.*

<sup>\*</sup>These authors contributed equally to this work. <sup>✉</sup>Corresponding authors: Zheng Zhu, zhengzhu@ieee.org, Xingang Wang, xingang.wang@ia.ac.cn.

## 1. Introduction

End-to-end planning [26, 27, 30], which directly maps sensor inputs to control signals, is among the most critical and promising tasks in autonomous driving. However, current open-loop evaluations are inadequate for accurately assessing end-to-end planning algorithms, highlighting an urgent need for enhanced evaluation methods [37, 39, 80]. A compelling solution lies in closed-loop evaluations within real-world scenarios, which require retrieving sensor data from arbitrarily specified viewpoints. This necessitates constructing a 4D driving scene representation capable of reconstructing complex, dynamic driving environments.

Closed-loop simulation in driving environments predominantly relies on scene reconstruction techniques such as Neural Radiance Fields (NeRF) [18, 45, 71, 73] and 3D Gaussian Splatting (3DGS) [11, 28, 32, 70], which are inherently limited by the density of input data. Specifically, these methods can render scenes effectively only under conditions closely aligned with their training data distributions—primarily forward-driving scenarios—and struggle to perform accurately during complex maneuvers (see Fig. 1). To mitigate these limitations, methods like SGD [78] and GGS [20] leverage generative models to extend the range of training viewpoints. However, these approaches primarily supplement sparse image data or static background elements, falling short of modeling the intricacies of dynamic, interactive driving scenes. Recently, advancements in autonomous driving world models [16, 25, 61, 63, 64, 81] have introduced the capability to generate diverse, command-aligned video viewpoints, offering renewed promise for closed-loop simulation in autonomous driving. Nonetheless, these models remain constrained to 2D videos, lacking the spatial-temporal coherence essential for accurately modeling complex driving scenarios.

In this paper, we introduce *DriveDreamer4D*, which improves 4D driving scene representation by integrating priors from autonomous driving world models. Our approach utilizes an autonomous driving world model [81] as a generative engine, synthesizing novel trajectory video data that densifies real-world driving datasets for enhanced training. Notably, we propose the Novel Trajectory Generation Module (NTGM) to generate diverse structured traffic conditions, and *DriveDreamer4D* applies these conditions to independently regulate the motion dynamics of foreground and background elements in complex driving environments. These conditions undergo view projection synchronized with vehicle maneuvers, ensuring that the synthesized data adheres to the spatiotemporal constraints. Subsequently, the Cousin Data Training Strategy (CDTS) is proposed to merge temporal-aligned real and synthetic data for training 4DGS. In CDTS, a regularization loss is further incorporated to ensure perceptual coherence. To the best of our knowledge, *DriveDreamer4D* is the first

framework to harness video generation models for elevating 4D scene reconstruction quality in autonomous driving, providing richly varied viewpoint data for scenarios including lane change, acceleration, and deceleration. As shown in Fig. 1, experiment results demonstrate that *DriveDreamer4D* significantly enhances generation fidelity for novel trajectory viewpoints, achieving a relative improvement in FID by 32.1%, 46.4%, and 16.3% compared to PVG [9], S<sup>3</sup>Gaussian [28], and Deformable-GS [74]. Besides, *DriveDreamer4D* fortifies the spatiotemporal coherence between foreground and background elements, with respective increases of 22.6%, 43.5%, and 15.6% in the NTA-IoU metric. Furthermore, a comprehensive user study confirms that the average win rate of *DriveDreamer4D* exceeds 80%, compared to three baselines.

The primary contributions of this work are as follows: (1) We present *DriveDreamer4D*, the first framework to leverage world model priors for advancing 4D scene reconstruction in autonomous driving. (2) The NTGM is proposed to automate the generation of structured conditions, allowing *DriveDreamer4D* to create novel trajectory videos with complex maneuvers while ensuring spatial-temporal consistency. Additionally, the CDTS is introduced to merge temporal-aligned real and synthetic data for training 4DGS, using a regularization loss to maintain perceptual coherence. (3) We perform comprehensive experiments to validate that *DriveDreamer4D* notably enhances generation quality across novel trajectory viewpoints, as well as the spatiotemporal coherence of driving scene elements.

## 2. Related Work

### 2.1. Driving Scene Representation

NeRF and 3DGS have emerged as leading approaches for 3D scene representation. NeRF models [2, 3, 45, 46] continuous volumetric scenes using multi-layer perceptron (MLP) networks, enabling highly detailed scene reconstructions with remarkable rendering quality. More recently, 3DGS [32, 77] introduces an innovative method by defining a set of anisotropic Gaussians in 3D space, leveraging adaptive density control to achieve high-quality renderings from sparse point cloud inputs. Several works have extended NeRF [12, 18, 29, 43, 52, 58, 71, 73] or 3DGS [9, 11, 28, 40, 70, 78, 82] to autonomous driving scenarios. Given the dynamic nature of driving environments, there has also been significant effort in modeling 4D driving scene representations. Some approaches encode time as an additional input to parameterize 4D scenes [1, 13, 28, 38, 42, 48, 56], while others represent scenes as a composition of moving object models alongside a static background model [35, 47, 59, 66, 68, 73]. Despite these advancements, methods based on NeRF and 3DGS face limitations tied to the density of input data. These tech-

niques can only render scenes effectively when sensor data closely matches the training data distribution, which is typically confined to forward-driving scenarios.

## 2.2. World Models

The world model module predicts possible future world states as a function of imagined action sequences proposed by the actor [36, 83]. Approaches such as [4, 5, 17, 19, 22–24, 34, 44, 62, 63, 67, 69, 75, 79] simulate environments through video generation controlled by free-text actions. At the forefront of this evolution is Sora [6], which leverages advanced generative techniques to produce intricate visual sequences that respect the fundamental laws of physics. This ability to deeply understand and simulate the environment not only improves video generation quality but also has substantial implications for real-world driving scenarios. Autonomous driving world models [16, 25, 61, 64, 72, 81] employ predictive methodologies to interpret driving environments, thereby generating realistic driving scenarios and learning key driving elements and policies from video data. Although these models successfully produce diverse driving video data conditioned on complex driving actions, they remain limited to 2D outputs and lack the spatial-temporal coherence needed to accurately capture the complexities of dynamic driving environments.

## 2.3. Diffusion Prior for 3D Representation

Constructing comprehensive 3D scenes from limited observations demands generative prior, particularly for unseen areas. Earlier studies distill the knowledge from text-to-image diffusion models [49, 51, 53, 54] into a 3D representation model. Specifically, the Score Distillation Sampling (SDS) [41, 50, 65] is adopted to synthesize a 3D object from the text prompt. Furthermore, to enhance 3D consistency, several approaches extend the multi-view diffusion models [15, 55] and video diffusion models [4, 10, 60] to 3D scene generation. To extend the diffusion prior to complex, dynamic, large-scale driving scenes for 3D reconstruction, methods such as SGD [78], GGS [20] and MagicDrive3D [14] employ generative models to broaden the range of training viewpoints. Nonetheless, these approaches mainly address sparse image data or static background elements, lacking the capacity to fully capture the complexities inherent in the 4D driving environments.

## 3. Method

In this section, we first elaborate on the preliminaries of 4D driving scene representation and world models for driving video generation. Then we present the details of *Drive-Dreamer4D*, which enhances 4D driving scene representation leveraging priors from driving world models.

## 3.1. Preliminary

### 3.1.1. 4D Driving Scene Representation

4DGS models the driving scene with a collection of 3DGS and a temporal field module. Each 3DGS [32] is parameterized by its center position  $\mathbf{x}$ , opacity  $\gamma$ , covariance  $\Sigma$ , and view-dependent RGB color  $\mathbf{c}$ , controlled via spherical harmonics. For stability, each covariance matrix  $\Sigma$  is decomposed by:

$$\Sigma = \mathbf{R} \mathbf{S} \mathbf{S}^T \mathbf{R}^T, \quad (1)$$

where scaling matrix  $\mathbf{S}$  and a rotation matrix  $\mathbf{R}$  are learnable parameters, represented by scaling  $s$  and quaternion  $r$ . All trainable parameters of a single 3D Gaussian are collectively denoted as  $\phi = \{\mathbf{x}, \gamma, s, r, \mathbf{c}\}$ . The temporal field  $\mathcal{F}$  takes  $\phi$  and a time step  $t_{gs}$  as input, outputting the offset  $\delta\phi = \{\delta\mathbf{x}, \delta\gamma, \delta\mathbf{s}, \delta\mathbf{r}, \delta\mathbf{c}\}$  for each Gaussian relative to canonical space. The 4D Gaussian  $\phi' = \{\mathbf{x}', \gamma', \mathbf{s}', \mathbf{r}', \mathbf{c}'\}$  is then computed by:

$$\phi' = \phi + \delta\phi = \phi + \mathcal{F}(\phi, t_{gs}). \quad (2)$$

Following [76], a differentiable Gaussian Splatting renderer is employed to project 4D Gaussians  $\phi$  into camera coordinates, yielding the covariance matrix  $\Sigma' = \mathbf{J} \mathbf{V} \Sigma \mathbf{V}^T \mathbf{J}^T$ , where  $\mathbf{J}$  is the Jacobian matrix of the perspective projection, and  $\mathbf{V}$  is the transform matrix. The color of each pixel is calculated by  $N$  ordered points using  $\alpha$ -blending:

$$C = \sum_{i \in N} T_i \mathbf{c}_i' \alpha_i, \quad (3)$$

where  $T_i$  is the transmittance defined by  $\prod_{j=1}^{i-1} (1 - \alpha_j)$ ,  $\mathbf{c}_i'$  denotes the color of each point,  $\alpha_i$  is given by evaluating a 2D Gaussian with covariance  $\Sigma'$  multiplied with a learned per-point opacity  $\gamma'_i$ . The trainable parameters  $\phi'$  can be optimized by a combination of RGB loss, depth loss and SSIM loss:

$$\begin{aligned} \mathcal{L}_{\text{ori}}(\phi') = & \lambda_1 \|\hat{I}_{\text{ori}} - I_{\text{ori}}\|_1 + \lambda_2 \|\hat{D}_{\text{ori}} - D_{\text{ori}}\|_1 \\ & + \lambda_3 \text{SSIM}(\hat{I}_{\text{ori}}, I_{\text{ori}}), \end{aligned} \quad (4)$$

where  $\hat{I}_{\text{ori}}$  and  $I_{\text{ori}}$  represent the rendered image and the ground truth image.  $\hat{D}_{\text{ori}}$  and  $D_{\text{ori}}$  are the rendered depth and the ground truth LiDAR depth map.  $\text{SSIM}(\cdot)$  refers to the operation of the Structural Similarity Index Measure, and  $\lambda_1, \lambda_2, \lambda_3$  are the loss weights.

### 3.1.2. World Models for Driving Video Generation

The world model module predicts possible future world states based on imagined action sequences [36]. Autonomous-driving world models [16, 61, 64, 81], typically based on diffusion models, leverage structured driving information or action controls to guide future video prediction. During training, these models first encode

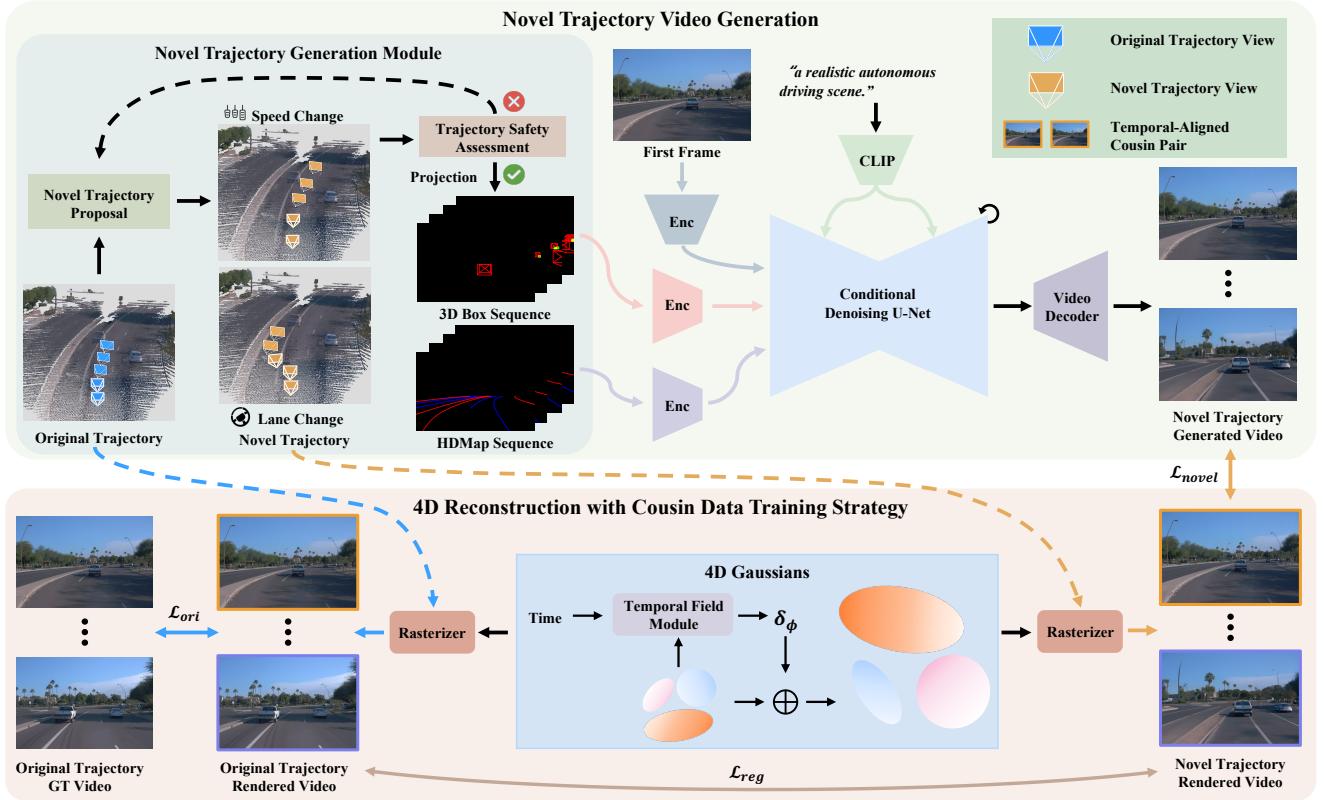


Figure 2. The overall framework of *DriveDreamer4D*. Initially, by altering the actions of the original trajectory (e.g., steering angle, speed), new trajectories can be obtained. Conditioned on the first frame and the structured information (3D bounding boxes, HDMap) from the new trajectory, the novel trajectory videos are generated. Subsequently, the temporal-aligned cousin pair (original and novel trajectory videos) are merged to optimize the 4D Gaussian Splatting model, where a regularization loss is calculated to ensure perceptual coherence.

videos  $\mathbf{v}$  into a lower-dimensional latent space  $\mathbf{z} = \mathcal{E}(\mathbf{v})$  using a variational encoder  $\mathcal{E}$ . After adding noise  $\epsilon_t$  to the latent, the diffusion model learns a denoising process. This diffusion process is optimized by:

$$\mathcal{L}_{diff} = \mathbb{E}_{\mathbf{z}, \epsilon \sim \mathcal{N}(0, 1), t} \left[ \|\epsilon_t - \epsilon_\theta(\mathbf{z}_t, t, \mathbf{f})\|_2^2 \right], \quad (5)$$

where  $\epsilon_\theta$  is a parameterized denoising network,  $t$  denotes the time step, representing the level of noise added or removed at each stage. Additionally, to improve the controllability of the generated data, conditional features  $\mathbf{f}$  (e.g., reference images, speed, steering angle, scene layouts, camera poses and textual information) can be introduced into the reverse diffusion process, ensuring that the generated outputs adhere to the input control signals. During inference, the world models can be conditioned on a reference image to control the style of the output scene, while predicting the future world states contingent upon the other input actions.

### 3.2. DriveDreamer4D

The overall pipeline of *DriveDreamer4D* is depicted in Fig. 2. In the upper part, the Novel Trajectory Generation Module (NTGM) is proposed to adjust driving actions (e.g.,

steering angle, speed) to generate new trajectories. These novel trajectories provide new perspectives for extracting structured information like 3D boxes and HDMap. Subsequently, a controllable video diffusion model synthesizes videos from these updated viewpoints, incorporating specific priors associated with the modified trajectories. In the lower part, the Cousin Data Training Strategy (CDTS) is introduced to combine the temporal-aligned original and generated data for optimizing the 4DGS model, where a regularization loss is calculated to impose perceptual coherence. In the following sections, we delve into the details of novel trajectories video generation and then introduce the CDTS for 4D reconstruction.

#### 3.2.1. Novel Trajectory Video Generation

As previously mentioned, traditional 4DGS methods are limited in rendering complex maneuvers, largely due to the training data being dominated by straightforward driving scenarios. To overcome this, *DriveDreamer4D* leverages world model priors to generate diverse viewpoint data, enhancing the 4D scene representation. To achieve this, we propose the NTGM, which is designed to create new trajectories that serve as input for the world model, en-

abling the automated generation of complex maneuver data. NTGM comprises two main components: (1) novel trajectory proposal, (2) trajectory safety assessment. In the novel trajectory proposal stage, *text-to-trajectory* [81] can be adopted to automatically generate diverse complex trajectories. Additionally, trajectories can be custom-designed to meet specific requirements, allowing for tailored data generation based on precise needs. The overview of the custom-designed trajectory proposal (e.g., lane change) and trajectory safety assessment is shown in the Algo. 1. In a specific driving scenario, the original trajectory in the world coordinate system can be readily acquired as  $\mathcal{T}_{\text{ori}}^{\text{world}} = \{p_i^{\text{world}}\}_{i=0}^K$ , where  $K$  denotes the number of frames and  $p_i^{\text{world}} \in \mathbb{R}^3$  refers to the position of the ego-vehicle at the  $i$ -th frame. To propose novel trajectories, the original trajectory  $\mathcal{T}_{\text{ori}}^{\text{world}}$  is transformed into the ego-vehicle coordinate system of the first frame, denoted as  $\mathcal{T}_{\text{ori}}^{\text{EgoStart}}$  and computed as:

$$[p_i^{\text{EgoStart}}, 1]^T = M_0^{-1} \times [p_i^{\text{world}}, 1]^T, \quad (6)$$

where  $M_0 \in \mathbb{R}^{4 \times 4}$  represents the transformation matrix from the ego-vehicle coordinate system of the first frame to the world coordinate system,  $[.]$  denotes the operation of the concat. In the ego-vehicle coordinate system, the vehicle's heading is aligned with the positive  $x$ -axis, the  $y$ -axis points to the left side of the vehicle, and the  $z$ -axis is oriented vertically upwards, perpendicular to the plane of the vehicle. Consequently, changes in the vehicle's velocity and direction can be respectively represented by adjusting the value along the  $x$ -axis and  $y$ -axis. A final safety assessment is conducted for the newly generated trajectory points, which includes verifying whether the vehicle trajectories  $p$  remain within drivable areas  $\mathcal{B}_{\text{road}}$  and ensuring that no collisions occur with pedestrians or other vehicles  $\{o_j\}_{j=1}^M$ .

$$\begin{aligned} p &\in \mathcal{B}_{\text{road}}, \\ \|p - o_j\| &\geq d_{\min}, \quad \forall j \in \{1, \dots, M\}, \end{aligned} \quad (7)$$

where  $d_{\min}$  is the minimal distance between different agents. Once a novel trajectory that complies with traffic regulations is generated, the road structure and 3D bounding boxes can be projected onto the camera view from the perspective of the new trajectory, thereby generating structured information relative to the updated trajectory. This structured information, along with the initial frame and text, is fed into a world model [81] to produce the videos that follow the novel trajectories.

### 3.2.2. Cousin Data Training Strategy

To better integrate generated data for training 4DGS, we propose the CDTs. Specifically, we construct temporally aligned cousin pair data as a minimal training batch:

$$\text{BatchStack}(\{\hat{I}_{\text{ori},t}\}_{t=0}^T, \{\hat{I}_{\text{novel},t}\}_{t=0}^T), \quad (8)$$

---

### Algorithm 1 Novel Trajectory Generation Module

---

**Input:** Trajectory  $\mathcal{T}_{\text{ori}}^{\text{world}}$ , Transformation matrix  $M_0$

**Output:** Novel trajectory  $\mathcal{T}_{\text{novel}}^{\text{ego}}$

$$\mathcal{T}_{\text{novel}}^{\text{ego}} \leftarrow [[0, 0, 0]]$$

$$\text{Offset} \leftarrow 0$$

**for** each  $p_{\text{ori}}^{\text{world}}$  in  $\mathcal{T}_{\text{ori}}^{\text{world}}[1:]$  **do**

$$p^{\text{EgoStart}} \leftarrow \text{RelativeCoord}(p_{\text{ori}}^{\text{world}}, M_0)$$

$$\text{MaxOffset} \leftarrow 0.1$$

**while** True **do**

$$\text{NewOffset} \leftarrow \text{Offset} + \text{RandOffset}(0, \text{MaxOffset})$$

$$p^{\text{EgoStart}'} \leftarrow p^{\text{EgoStart}} + [0, \text{NewOffset}, 0]$$

**if** SafeCheck( $p^{\text{EgoStart}'}$ ) **then**

$$\text{AddElement}(\mathcal{T}_{\text{novel}}^{\text{ego}}, p^{\text{EgoStart}'})$$

$$\text{Offset} \leftarrow \text{NewOffset}$$

**break**

**else**

$$\text{MaxOffset} \leftarrow \text{MaxOffset}/2$$

**end if**

**end while**

**end for**

---

where  $\text{BatchStack}(\cdot)$  is the data processor to stack temporal-aligned  $\{\hat{I}_{\text{ori},t}\}_{t=0}^T$  and  $\{\hat{I}_{\text{novel},t}\}_{t=0}^T$  into a training batch. By leveraging real and synthetic data aligned at each timestep, CDTs mitigates the data gap in 4DGS training, enhancing the model's ability to learn consistent representations across real and synthetic data. To optimize 4DGS, the temporal-aligned cousin pair is input per step before gradient optimization. The loss function  $\mathcal{L}_{\text{ori}}$  for the original data is defined in Eq. 4. And the loss function  $\mathcal{L}_{\text{novel}}$  for the generated data is akin to [9, 28], defined as follows:

$$\begin{aligned} \mathcal{L}_{\text{novel}}(\phi') &= \lambda_1 \|\hat{I}_{\text{novel}} - I_{\text{novel}}\|_1 \\ &\quad + \lambda_3 \text{SSIM}(\hat{I}_{\text{novel}}, I_{\text{novel}}), \end{aligned} \quad (9)$$

where  $I_{\text{novel}}$  represents the generated images corresponding to the novel trajectories as described in Sec. 3.2.1, and  $\hat{I}_{\text{novel}}$  denotes the rendered images under the novel trajectories via differentiable splatting [76]. Notably, different from [9, 28], depth maps are not employed as constraints in the optimization of 4DGS when using the generated dataset  $D_{\text{novel}}$ . The limitation arises from the fact that LiDAR point cloud data is exclusively collected for the original trajectory. When these LiDAR points are projected onto a new trajectory, it cannot produce a complete depth map for the new perspective, as something visible in the novel trajectory may have been occluded in the original view. Consequently, the incorporation of such depth maps does not facilitate the optimization of the 4DGS model. More details are described in Sec. 4.3. Additionally, we propose a regularization loss to enhance the perceptual coherence, defined as follows:

$$\mathcal{L}_{\text{reg}}(\phi') = \|\mathcal{F}_p(\hat{I}_{\text{ori}}) - \mathcal{F}_p(\hat{I}_{\text{novel}})\|_1, \quad (10)$$

Method	Lane Change		Acceleration		Deceleration		Average	
	NTA-IoU ↑	NTL-IoU ↑						
PVG [9]	0.256	50.70	0.396	53.08	0.394	53.65	0.349	52.48
<i>DriveDreamer4D</i> with PVG	<b>0.438</b>	<b>53.06</b>	<b>0.421</b>	<b>53.35</b>	<b>0.424</b>	<b>53.89</b>	<b>0.428</b>	<b>53.43</b>
S <sup>3</sup> Gaussian [28]	0.175	49.05	0.434	51.93	0.384	52.14	0.331	51.04
<i>DriveDreamer4D</i> with S <sup>3</sup> Gaussian	<b>0.495</b>	<b>53.42</b>	<b>0.484</b>	<b>52.63</b>	<b>0.445</b>	<b>52.69</b>	<b>0.475</b>	<b>52.91</b>
Deformable-GS [74]	0.240	51.62	0.346	52.17	0.377	53.21	0.321	52.33
<i>DriveDreamer4D</i> with Deformable-GS	<b>0.335</b>	<b>52.93</b>	<b>0.371</b>	<b>52.77</b>	<b>0.406</b>	<b>53.79</b>	<b>0.371</b>	<b>53.16</b>

Table 1. Comparison of NTA-IoU and NTL-IoU scores across different novel trajectory views (lane change, acceleration, deceleration).

where  $\mathcal{F}_p(\cdot)$  denotes the perception feature extraction model [21]. The overall loss function for mixed training is defined as follows:

$$\mathcal{L}(\phi') = \mathcal{L}_{\text{ori}} + \lambda_{\text{novel}} \mathcal{L}_{\text{novel}} + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}}. \quad (11)$$

## 4. Experiments

In this section, we first outline the experimental setup, including the dataset, implementation details, and evaluation metrics. Subsequently, both quantitative and qualitative evidence are provided to demonstrate that the proposed *DriveDreamer4D* significantly enhances rendering quality for novel trajectory viewpoints and improves the spatiotemporal coherence of foreground and background components. Finally, we conduct an ablation study on the hyperparameter settings, as well as the effects of depth loss and the proposed CDTs, including temporal-aligned pairs, and regularization loss.

### 4.1. Experiment Setup

**Dataset.** We conduct experiments using the Waymo dataset [57], known for its comprehensive real-world driving logs. However, most logs capture scenes with relatively straightforward dynamics, lacking focus on scenarios with dense, complex vehicle interactions. To address this gap, we specifically select eight scenes characterized by highly dynamic interactions, featuring numerous vehicles with diverse relative positions and intricate driving trajectories. Each selected segment contains approximately 40 frames, with segment IDs detailed in the supplement.

**Implementation Details.** To demonstrate the versatility and robustness of *DriveDreamer4D*, we incorporate various 4DGS baselines into our pipeline, including Deformable-GS [74], S<sup>3</sup>Gaussian [28], and PVG [9]. For a fair comparison, LiDAR supervision is introduced to Deformable-GS. During training, scenes are segmented into multiple clips, each containing 40 frames, aligned with the generative model’s output length. We use only forward-facing camera data and standardize the resolution across methods to  $640 \times 960$ . Our models are trained using the Adam optimizer [33], following the learning rate schedule used for 3D Gaussian Splatting [32]. Hyperparameter settings are

Method	FID ↓
PVG[9]	105.29
<i>DriveDreamer4D</i> with PVG	<b>71.52</b>
S <sup>3</sup> Gaussian[28]	124.90
<i>DriveDreamer4D</i> with S <sup>3</sup> Gaussian	<b>66.93</b>
Deformable-GS[74]	92.34
<i>DriveDreamer4D</i> with Deformable-GS	<b>77.32</b>

Table 2. Comparison of FID scores in novel trajectory view synthesis (lane change) on the Waymo dataset.

aligned with each baseline [9, 28, 74], and the training strategy remains the same, with the exception of the incorporation of CDTs.

**Metrics.** Traditional 3D reconstruction tasks typically employ PSNR and SSIM metrics for evaluation, with validation sets that closely match the training data distribution (i.e., uniformly sampling frames from video sequences for validation, with the remainder used for training). However, in closed-loop driving simulation, the focus shifts to evaluating model rendering performance under novel trajectories, where corresponding sensor data are unavailable, making metrics like PSNR and SSIM inapplicable for evaluation. Therefore, we propose Novel Trajectory Agent IoU (NTA-IoU) and Novel Trajectory Lane IoU (NTL-IoU), which assess the spatiotemporal coherence of foreground and background traffic components in novel trajectory viewpoints.

For NTA-IoU, we use YOLO11 [31] to identify vehicles in images rendered from novel trajectory views, yielding 2D bounding boxes. Simultaneously, geometric transformations are applied to the original 3D bounding boxes, projecting them onto the new viewpoints to generate corresponding 2D bounding boxes. For each projected 2D box, we then identify the closest detector-generated 2D box and compute their Intersection over Union (IoU). To ensure accurate matching, a distance threshold  $d_{\text{thresh}}$  is introduced: when the center-to-center distance  $\|c(B^{\text{proj}}) - c(B^{\text{det}})\|$  between the nearest detected box  $B^{\text{det}}$  and the correctly projected box  $B^{\text{proj}}$  surpasses this threshold, their NTA-IoU is



Figure 3. Qualitative comparisons of novel trajectory renderings during lane change scenarios. The orange boxes highlight that *DriveDreamer4D* significantly enhances the rendering quality across various baselines (PVG [9],  $S^3$ Gaussian [28], Deformable-GS [74]).

assigned a value of zero:

$$\text{NTA-IoU} = \begin{cases} 0 & \text{if } \|c(B^{\text{proj}}) - c(B^{\text{det}})\| \geq d_{\text{thresh}} \\ \text{IoU}(B^{\text{proj}}, B^{\text{det}}) & \text{otherwise.} \end{cases} \quad (12)$$

For NTL-IoU, we employ TwinLiteNet [8] to extract 2D lanes from rendered images. Ground truth lanes are also projected onto the 2D image plane. We then compute the mean Intersection-over-Union (mIoU) between the rendered and ground truth lanes  $L^{\text{det}}$  and  $L^{\text{proj}}$ :

$$\text{NTL-IoU} = \text{mIoU}(L^{\text{proj}}, L^{\text{det}}). \quad (13)$$

Additionally, in lane change scenarios, we observe inaccuracies in relative positioning, as well as frequent occurrences of artifacts such as flying points and ghosting, which notably degrade image quality. To assess this, we employ the FID metric [21], which quantifies differences in feature distribution between rendered novel trajectory images and original trajectory images. This metric effectively reflects visual quality and is particularly sensitive to artifacts like flying points and ghosting, providing a robust measure of image fidelity in these complex scenes. Finally, a user study is conducted to evaluate the quality of the ren-

Counterpart Method	<i>DriveDreamer4D</i> Win Rate			
	Lane Change	Acceleration	Deceleration	Average
PVG [9]	100.0%	90.5%	89.1%	<b>93.2%</b>
S <sup>3</sup> Gaussian [28]	100.0%	97.9%	92.2%	<b>96.7%</b>
Deformable-GS [74]	95.8%	83.5%	72.9%	<b>84.1%</b>

Table 3. User study comparison of *DriveDreamer4D* win rates across various novel trajectory view synthesis.

derings, where participants compare the rendering results of each baseline with its *DriveDreamer4D* enhanced version across three novel trajectories. The evaluation criteria focus on overall video quality, with particular attention to foreground objects like vehicles. For each comparison, participants were asked to select the option they found most favorable. Further details are provided in the supplement.

## 4.2. Comparison with Different 4DGS Baselines

**Quantitative Results.** As demonstrated in Tab. 1, integrating *DriveDreamer4D* with different 4DGS algorithms consistently yields superior NTA-IoU and NTL-IoU scores across diverse, complex maneuvers (e.g., lane changes, acceleration, and deceleration), significantly outperforming the baseline methods. Specifically, with *DriveDreamer4D*, the average NTA-IoU scores across three baselines (PVG [9], S<sup>3</sup>Gaussian [28], Deformable-GS [74]) are relatively enhanced by 22.6%, 43.5%, and 15.6%, underscoring *DriveDreamer4D*'s capability to improve the spatiotemporal coherence of foreground agents. Moreover *DriveDreamer4D* facilitates a relative improvement in average NTL-IoU for these baselines by 1.8%, 3.7%, and 1.6%, thereby markedly enhancing the spatiotemporal coherence of background lanes in 4D rendering of driving scenarios.

In addition to verifying the spatiotemporal consistency of rendered novel trajectory views, we leverage the FID metric to assess rendering quality under novel trajectories. Given that acceleration and deceleration scenarios yield rendered views with distributional similarities to ground truth, limiting FID's discriminative capability across algorithms, our FID comparisons focus specifically on lane change scenarios. Experiment results, as presented in Tab. 2, indicate that our method substantially outperforms the baseline methods (PVG [9], S<sup>3</sup>Gaussian [28], Deformable-GS [74]), with FID relative improvements of 32.1%, 46.4%, and 16.3%. These results highlight *DriveDreamer4D*'s capability to enhance generation quality for novel trajectory viewpoints.

Finally, we conduct a user study to evaluate the rendering quality of different methods on novel trajectories, with a specific focus on foreground agents. For each method, we generate three novel trajectory views—lane change, acceleration, and deceleration—across eight scenes from the Waymo dataset [57]. Participants are then asked to select the renderings they found most visually favorable in each comparison. The *DriveDreamer4D* win rates from this

$\lambda_{\text{novel}}$	NTA-IoU $\uparrow$	FID $\downarrow$	$\lambda_{\text{reg}}$	NTA-IoU $\uparrow$	FID $\downarrow$
0	0.349	105.29	0	0.420	79.54
0.5	0.405	82.84	1e-2	0.411	119.39
1	<b>0.420</b>	<b>79.54</b>	1e-3	<b>0.428</b>	<b>71.52</b>
1.5	0.417	82.10	1e-4	0.422	75.31

Table 4. Ablation study on the training loss weight  $\lambda_{\text{novel}}$  for novel trajectory data.

Table 5. Ablation study on the regularization loss weight  $\lambda_{\text{reg}}$ .

Depth Loss	Temporal-aligned Cousin Pair	Regularization Loss	NTA-IoU $\uparrow$	FID $\downarrow$
✓	✗	✗	0.401	82.63
✗	✗	✗	0.420	79.54
✗	✓	✗	0.423	76.20
✗	✓	✓	<b>0.428</b>	<b>71.52</b>

Table 6. Ablation study on the depth loss and CDTs.

study, shown in Tab. 3, reveal a significant user preference for our method's renderings.

**Qualitative Results.** In addition to quantitative comparisons, we provide a qualitative analysis of novel trajectory view renderings. As shown in Fig. 3, we present the novel trajectory view synthesis during lane change. Images rendered by the baseline algorithms exhibit issues where foreground vehicles incorrectly change lanes in sync with the camera's motion, and some vehicles are incompletely rendered. Additionally, the background is filled with speckles and ghosting. Especially shown in the rightmost column of Fig. 3, baseline algorithms often produce blurred, ghosted foreground vehicles and background speckles in the sky, alongside blurred lane markings. Our method, however, significantly improves rendering quality, as highlighted by the orange boxes. Vehicle contours are sharper, and background artifacts such as speckles and ghosting are substantially reduced.

## 4.3. Ablation Studies

We conduct an ablation study based on PVG [9], analyzing the effects of hyperparameters settings for  $\lambda_{\text{novel}}$  and  $\lambda_{\text{reg}}$ , as well as the impact of depth loss, temporal-aligned cousin pairs, and regularization loss. Following the experiments shown in Tab. 4 and 5, we set  $\lambda_{\text{novel}}$  to 1 and  $\lambda_{\text{reg}}$  to  $1 \times 10^{-3}$ , which yield the best performance. As shown in Tab. 6, the experiment confirms that the depth loss should be excluded when optimizing novel trajectory views, since LiDAR depth maps are incomplete due to occlusions. Furthermore, employing temporal-aligned cousin pairs achieves an FID of 76.20 and an NTA-IoU of 0.423, with the perception regularization loss further improving the FID to 71.52 and NTA-IoU to 0.428. These results highlight the effectiveness of CDTs, in achieving  $\sim 10\%$  improvement in FID, along with a  $\sim 2\%$  increase in NTA-IoU.

## 5. Discussion and Conclusion

In this paper, we presented *DriveDreamer4D*, a novel framework designed to advance 4D driving scene representations by harnessing priors from world models. Addressing key limitations of current sensor simulation methods—namely, their dependence on forward-driving training data distributions and inability to model complex maneuvers—*DriveDreamer4D* leverages a world model to generate novel trajectory videos that complement real-world driving data. By explicitly employing structured conditions, our framework maintains spatial-temporal consistency across traffic elements, ensuring that generated data adheres closely to the dynamics of real-world traffic scenarios. Our experiments demonstrate that *DriveDreamer4D* achieves superior quality in generating diverse simulation viewpoints, with significant improvements in both the rendering fidelity and spatiotemporal coherence of scene components. Notably, these results highlight *DriveDreamer4D*’s potential as a foundation for closed-loop simulations that require high-fidelity reconstructions of dynamic driving scenes.

## References

- [1] Benjamin Attal, Jia-Bin Huang, Christian Richardt, Michael Zollhoefer, Johannes Kopf, Matthew O’Toole, and Changil Kim. Hyperreal: High-fidelity 6-dof video with ray-conditioned sampling. In *CVPR*, 2023. 2
- [2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022. 2
- [3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *ICCV*, 2023. 2
- [4] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 3, 14
- [5] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, 2023. 3
- [6] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. 3
- [7] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liang, Qiang Xu, Anush Krishnan, Yu Pan, Giacarlo Baldan, and Oscar Beijbom. nuscenes: A multi-modal dataset for autonomous driving. In *CVPR*, 2020. 13
- [8] Quang-Huy Che, Dinh-Phuc Nguyen, Minh-Quan Pham, and Duc-Khai Lam. Twinlitenet: An efficient and lightweight model for driveable area and lane segmentation in self-driving cars. In *MAPR*, 2023. 7
- [9] Yurui Chen, Chun Gu, Junzhe Jiang, Xiatian Zhu, and Li Zhang. Periodic vibration gaussian: Dynamic urban scene reconstruction and real-time rendering. *arXiv preprint arXiv:2311.18561*, 2023. 1, 2, 5, 6, 7, 8, 13, 14, 15
- [10] Zilong Chen, Yikai Wang, Feng Wang, Zhengyi Wang, and Huaping Liu. V3d: Video diffusion models are effective 3d generators. *arXiv preprint arXiv:2403.06738*, 2024. 3
- [11] Ziyu Chen, Jiawei Yang, Jiahui Huang, Riccardo de Lutio, Janick Martinez Esturo, Boris Ivanovic, Or Litany, Zan Gocic, Sanja Fidler, Marco Pavone, Li Song, and Yue Wang. Omnidre: Omni urban scene reconstruction. *arXiv preprint arXiv:2408.16760*, 2024. 2
- [12] Kai Cheng, Xiaoxiao Long, Wei Yin, Jin Wang, Zhiqiang Wu, Yuexin Ma, Kaixuan Wang, Xiaozhi Chen, and Xuejin Chen. Uc-nerf: Neural radiance field for under-calibrated multi-view cameras in autonomous driving. *arXiv preprint arXiv:2311.16945*, 2023. 2
- [13] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *CVPR*, 2023. 2
- [14] Ruiyuan Gao, Kai Chen, Zihao Li, Lanqing Hong, Zhenguo Li, and Qiang Xu. Magicdrive3d: Controllable 3d generation for any-view rendering in street scenes. *arXiv preprint arXiv:2405.14475*, 2024. 3
- [15] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv preprint arXiv:2405.10314*, 2024. 3
- [16] Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. *arXiv preprint arXiv:2405.17398*, 2024. 2, 3
- [17] Rohit Girdhar, Mannat Singh, Andrew Brown, Quentin Duval, Samaneh Azadi, Sai Saketh Rambhatla, Akbar Shah, Xi Yin, Devi Parikh, and Ishan Misra. Emu video: Factorizing text-to-video generation by explicit image conditioning. *arXiv preprint arXiv:2311.10709*, 2023. 3
- [18] Jianfei Guo, Nianchen Deng, Xinyang Li, Yeqi Bai, Botian Shi, Chiyu Wang, Chenjing Ding, Dongliang Wang, and Yikang Li. Streetsurf: Extending multi-view implicit surface reconstruction to street views. *arXiv preprint arXiv:2306.04988*, 2023. 2
- [19] Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Li Fei-Fei, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic video generation with diffusion models. *arXiv preprint arXiv:2312.06662*, 2023. 3
- [20] Huasong Han, Kaixuan Zhou, Xiaoxiao Long, Yusen Wang, and Chunxia Xiao. Ggs: Generalizable gaussian splatting for lane switching in autonomous driving. *arXiv preprint arXiv:2409.02382*, 2024. 2, 3
- [21] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 2017. 6, 7

- [22] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 3
- [23] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *NeurIPS*, 2022.
- [24] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022. 3
- [25] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023. 2, 3
- [26] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *ECCV*, 2022. 2
- [27] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *CVPR*, 2023. 2
- [28] Nan Huang, Xiaobao Wei, Wenzhao Zheng, Pengju An, Ming Lu, Wei Zhan, Masayoshi Tomizuka, Kurt Keutzer, and Shanghang Zhang.  $s^3$ gaussian: Self-supervised street gaussians for autonomous driving. *arXiv preprint arXiv:2405.20323*, 2024. 1, 2, 5, 6, 7, 8, 13, 14, 15
- [29] Muhammad Zubair Irshad, Sergey Zakharov, Katherine Liu, Vitor Guizilini, Thomas Kollar, Adrien Gaidon, Zsolt Kira, and Rares Ambrus. Neo 360: Neural fields for sparse view synthesis of outdoor scenes. In *ICCV*, 2023. 2
- [30] Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *ICCV*, 2023. 2
- [31] Glenn Jocher and Jing Qiu. Ultralytics yolo11, 2024. 6
- [32] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM ToG*, 2023. 2, 3, 6
- [33] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6, 14
- [34] Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Rachel Hornung, Hartwig Adam, Hassan Akbari, Yair Alon, Vighnesh Birodkar, et al. Videopoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125*, 2023. 3
- [35] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *CVPR*, 2022. 2
- [36] Yann LeCun and Courant. A path towards autonomous machine intelligence version 0.9.2, 2022-06-27. 2022. 3
- [37] Hao Li, Ming Yuan, Yan Zhang, Chenming Wu, Chen Zhao, Chunyu Song, Haocheng Feng, Errui Ding, Dingwen Zhang, and Jingdong Wang. Xld: A cross-lane dataset for benchmarking novel driving view synthesis. *arXiv preprint arXiv:2406.18360*, 2024. 2
- [38] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 2021. 2
- [39] Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahua Li, Jan Kautz, Tong Lu, and Jose M Alvarez. Is ego status all you need for open-loop end-to-end autonomous driving? In *CVPR*, 2024. 2
- [40] Zhuopeng Li, Yilin Zhang, Cheming Wu, Jianke Zhu, and Liangjun Zhang. Ho-gaussian: Hybrid optimization of 3d gaussian splatting for urban scenes. *arXiv preprint arXiv:2403.20032*, 2024. 2
- [41] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *CVPR*, 2023. 3
- [42] Haotong Lin, Sida Peng, Zhen Xu, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Efficient neural radiance fields for interactive free-viewpoint video. In *SIGGRAPH Asia*, 2022. 2
- [43] Fan Lu, Yan Xu, Guang Chen, Hongsheng Li, Kwan-Yee Lin, and Changjun Jiang. Urban radiance field representation with deformable neural mesh primitives. In *ICCV*, 2023. 2
- [44] Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024. 3
- [45] 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. *CACM*, 2021. 2
- [46] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM ToG*, 2022. 2
- [47] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knott, and Felix Heide. Neural scene graphs for dynamic scenes. In *CVPR*, 2021. 2
- [48] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021. 2
- [49] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 3
- [50] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 3
- [51] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 3

- [52] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *CVPR*, 2022. 2
- [53] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 3
- [54] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, 2022. 3
- [55] Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, et al. Zeronvs: Zero-shot 360-degree view synthesis from a single real image. *arXiv preprint arXiv:2310.17994*, 2023. 3
- [56] Liangchen Song, Anpei Chen, Zhong Li, Zhang Chen, Lele Chen, Junsong Yuan, Yi Xu, and Andreas Geiger. Nerf-player: A streamable dynamic scene representation with decomposed neural radiance fields. *IEEE TVCG*, 2023. 2
- [57] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020. 6, 8, 13, 14, 15
- [58] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *CVPR*, 2022. 2
- [59] Adam Tonderski, Carl Lindström, Georg Hess, William Ljungbergh, Lennart Svensson, and Christoffer Petersson. Neurad: Neural rendering for autonomous driving. In *CVPR*, 2024. 2
- [60] Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitry Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion. *arXiv preprint arXiv:2403.12008*, 2024. 3
- [61] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-driven world models for autonomous driving. *arXiv preprint arXiv:2309.09777*, 2023. 2, 3
- [62] Xiaofeng Wang, Kang Zhao, Feng Liu, Jiayu Wang, Guosheng Zhao, Xiaoyi Bao, Zheng Zhu, Yingya Zhang, and Xingang Wang. Egovid-5m: A large-scale video-action dataset for egocentric video generation. *arXiv preprint arXiv:2411.08380*, 2024. 3
- [63] Xiaofeng Wang, Zheng Zhu, Guan Huang, Boyuan Wang, Xinze Chen, and Jiwen Lu. Worlddreamer: Towards general world models for video generation via predicting masked tokens. *arXiv preprint arXiv:2401.09985*, 2024. 2, 3
- [64] Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *CVPR*, 2024. 2, 3
- [65] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. Reconfusion: 3d reconstruction with diffusion priors. In *CVPR*, 2024. 3
- [66] Zirui Wu, Tianyu Liu, Liyi Luo, Zhide Zhong, Jianteng Chen, Hongmin Xiao, Chao Hou, Haozhe Lou, Yuntao Chen, Runyi Yang, et al. Mars: An instance-aware, modular and realistic simulator for autonomous driving. In *ICAI*, 2023. 2
- [67] Jiannan Xiang, Guangyi Liu, Yi Gu, Qiyue Gao, Yuting Ning, Yuheng Zha, Zeyu Feng, Tianhua Tao, Shibo Hao, Yemin Shi, et al. Pandora: Towards general world model with natural language actions and video states. *arXiv preprint arXiv:2406.09455*, 2024. 3
- [68] Ziyang Xie, Junge Zhang, Wenye Li, Feihu Zhang, and Li Zhang. S-nerf: Neural radiance fields for street views. *arXiv preprint arXiv:2303.00749*, 2023. 2
- [69] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021. 3
- [70] Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians for modeling dynamic urban scenes. *arXiv preprint arXiv:2401.01339*, 2024. 2
- [71] Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, et al. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. *arXiv preprint arXiv:2311.02077*, 2023. 2
- [72] Xuemeng Yang, Licheng Wen, Yukai Ma, Jianbiao Mei, Xin Li, Tiantian Wei, Wenjie Lei, Daocheng Fu, Pinlong Cai, Min Dou, Botian Shi, Liang He, Yong Liu, and Yu Qiao. Drivearena: A closed-loop generative simulation platform for autonomous driving. *arXiv preprint arXiv:2408.00415*, 2024. 3
- [73] Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *CVPR*, 2023. 2
- [74] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *CVPR*, 2024. 1, 2, 6, 7, 8, 13, 14, 15
- [75] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 3
- [76] Wang Yifan, Felice Serena, Shihao Wu, Cengiz Öztireli, and Olga Sorkine-Hornung. Differentiable surface splatting for point-based geometry processing. *ACM TOG*, 2019. 3, 5
- [77] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *CVPR*, 2024. 2
- [78] Zhongrui Yu, Haoran Wang, Jinze Yang, Hanzhang Wang, Zeke Xie, Yunfeng Cai, Jiale Cao, Zhong Ji, and Mingming

- Sun. Sgd: Street view synthesis with gaussian splatting and diffusion prior. *arXiv preprint arXiv:2403.20079*, 2024. 2, 3
- [79] Yan Zeng, Guoqiang Wei, Jiani Zheng, Jiaxin Zou, Yang Wei, Yuchen Zhang, and Hang Li. Make pixels dance: High-dynamic video generation. *arXiv preprint arXiv:2311.10982*, 2023. 3
- [80] Jiang-Tian Zhai, Ze Feng, Jinhao Du, Yongqiang Mao, Jiang-Jiang Liu, Zichang Tan, Yifu Zhang, Xiaoqing Ye, and Jingdong Wang. Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes. *arXiv preprint arXiv:2305.10430*, 2023. 2
- [81] 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. 2, 3, 5, 13
- [82] Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *CVPR*, pages 21634–21643, 2024. 2
- [83] Zheng Zhu, Xiaofeng Wang, Wangbo Zhao, Chen Min, Nianchen Deng, Min Dou, Yuqi Wang, Botian Shi, Kai Wang, Chi Zhang, et al. Is sora a world simulator? a comprehensive survey on general world models and beyond. *arXiv preprint arXiv:2405.03520*, 2024. 3



Figure 4. Visual comparisons in the novel trajectories for the Cousin Data Training Strategy (CDTS) ablation study. The orange boxes emphasize the superior performance of *DriveDreamer4D* and the further improvements in detail rendering brought by CDTS.

In the supplementary material, we begin by introducing the three baseline methods employed in our work. Next, we elaborate on the implementation of *DriveDreamer4D*, covering the training for novel trajectory video generation, the selection of scenes, and the setup of the user study. Finally, additional visualizations are presented to illustrate the improved rendering quality achieved through Cousin Data Training Strategy (CDTS) and showcase the performance of *DriveDreamer4D* in speed change scenarios.

## 6. Baselines

To demonstrate the effectiveness and generalizability of our method, three different 4D Gaussian Splatting (4DGS) baselines are selected for the experiments. In this section, we briefly introduce the three baselines employed in this paper: PVG [9], S<sup>3</sup>Gaussian [28], and Deformable-GS [74].

**PVG** [9] introduces a unified representation model known as Periodic Vibration Gaussians (PVGs), which vibrate over time with optimizable parameters, including vibration directions, lifespan, and life peak (the moment of highest opacity), to effectively represent dynamic scenes. The model employs a self-supervised approach to optimize these Gaussians and achieves static-dynamic decomposition by classifying them based on their lifespans. This method allows PVG to effectively represent the characteristics of various objects and elements in dynamic urban scenes.

**S<sup>3</sup>Gaussian** [28] proposes a self-supervised street Gaussian method to model complex 4D dynamic scenes. Each scene

is represented using 3D Gaussians to preserve explicitness, and a spatial-temporal field network is employed to compactly model the 4D dynamics. To facilitate efficient scene reconstruction without costly annotations, it utilizes an self-supervised approach to decompose dynamic and static 3D Gaussians.

**Deformable-GS** [74] represents scenes using a canonical space defined by Gaussian distributions. It models scene dynamic by employing a deformation network to predict offsets for the Gaussian parameters. These offsets adjust the Gaussians to align with the dynamic elements of the scene. Additionally, Deformable-GS has demonstrated strong performance in both synthetic and indoor datasets.

## 7. Implementation Details

In Sec. 7, we primarily introduce the training for novel trajectory video generation, the selection of scenes, and the details of the user study.

**Training for Novel Trajectory Video Generation.** As depicted in the upper part of Fig. 2 (in the main text), a controllable driving video generation model is crucial for producing novel trajectory videos. Specifically, we follow the approach outlined in [81] to train such a model on the Waymo dataset [57]. Unlike [81], which focuses on multi-view video generation using the nuScenes dataset [7], our work concentrates solely on front-view video generation. This focus allows us to increase the number of frames to 40 and the resolution to 960 × 640, a significant improve-

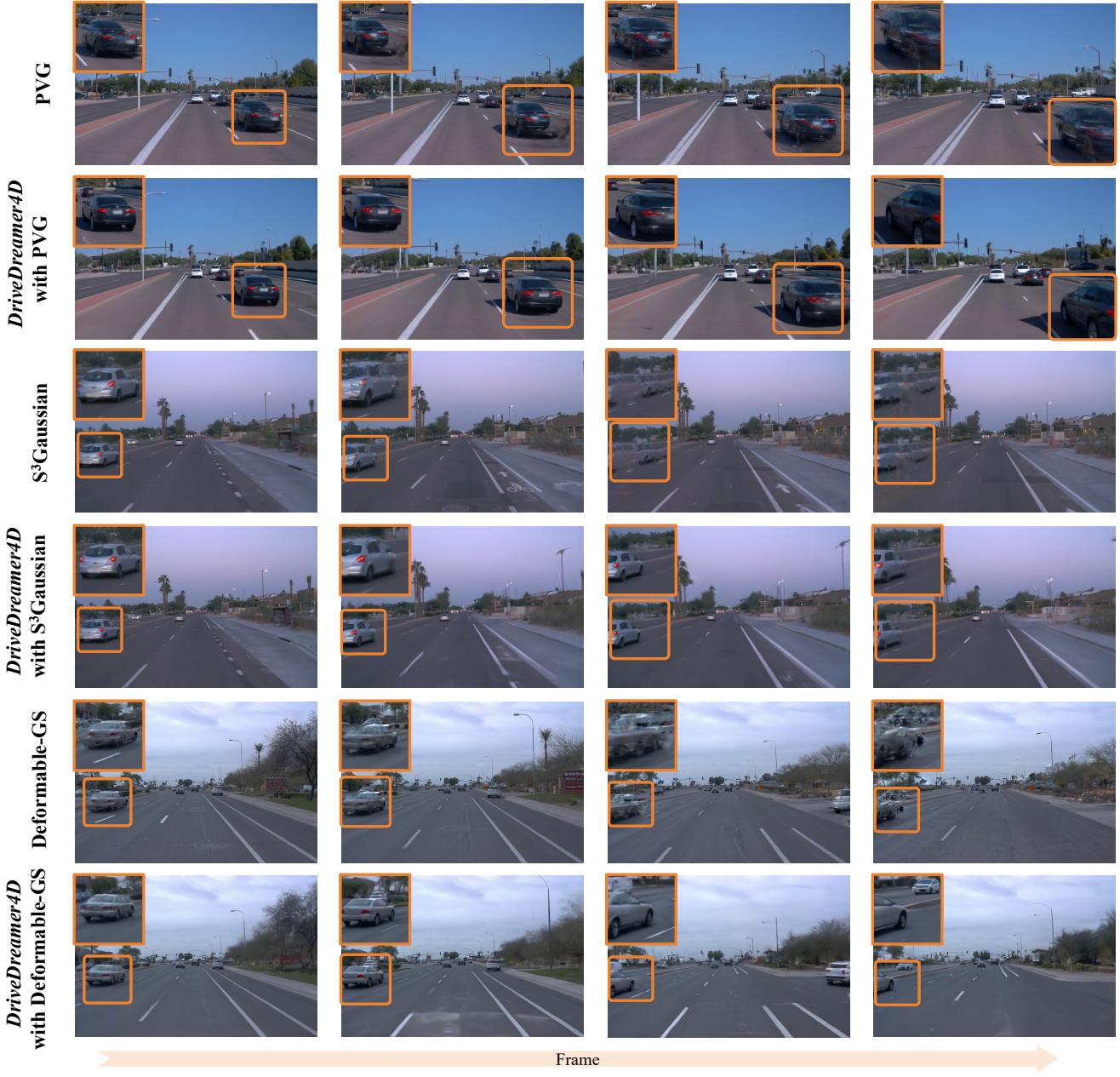


Figure 5. Qualitative comparisons of novel trajectory renderings during speed change scenarios. The orange boxes highlight that *DriveDreamer4D* significantly enhances the rendering quality across various baseline methods (PVG [9], S<sup>3</sup>Gaussian [28], Deformable-GS [74]).

ment compared to the previous 8 frames at a resolution of  $448 \times 256$ . The increase in both frame count and resolution contributes to an enhanced performance of the reconstruction model, particularly for novel trajectory generation. As for the training data, it comprises the entire Waymo training split, consisting of 798 videos. To enhance the dataset, we further divide these videos into 40-frame clips, resulting in approximately 64K clips. Additionally, the training pro-

cess is initialized with parameters from SVD [4], with 3D bounding boxes, HDMaps, and text incorporated as control conditions. And, the AdamW optimizer [33] is employed for parameter optimization, with a learning rate of  $5 \times 10^{-5}$ , a batch size of 8, and a total of 50K iterations. All experiments are conducted on an NVIDIA H20 (96GB) GPUs.

**Scene Selection.** All selected scenes are sourced from the validation set of the Waymo dataset [57] and are carefully

Scene	Start Frame	End Frame
segment-10359308928573410754_720_000_740_000_with_camera_labels.tfrecord	120	159
segment-12820461091157089924_5202_916_5222_916_with_camera_labels.tfrecord	0	39
segment-15021599536622641101_556_150_576_150_with_camera_labels.tfrecord	0	39
segment-16767575238225610271_5185_000_5205_000_with_camera_labels.tfrecord	0	39
segment-17152649515605309595_3440_000_3460_000_with_camera_labels.tfrecord	60	99
segment-17860546506509760757_6040_000_6060_000_with_camera_labels.tfrecord	90	129
segment-2506799708748258165_6455_000_6475_000_with_camera_labels.tfrecord	80	119
segment-3015436519694987712_1300_000_1320_000_with_camera_labels.tfrecord	40	79

Table 7. Selected scenes from the validation set of the Waymo dataset [57].

chosen based on their distinctive characteristics. Specifically, the selection prioritizes scenes that exhibit significant motion dynamics, such as large-scale maneuvers, as these scenarios pose greater challenges for both video reconstruction and trajectory generation tasks. Tab. 7 shows all 8 scenes selected for our experiments. The official file names of these scenes, as provided in [57], are listed along with their respective starting and ending frames.

**User Study.** For the eight different scenes mentioned above, we create 72 comparison videos for the user study, covering three novel trajectories (acceleration, deceleration, and lane change) under three different baselines. To ensure fairness, the baseline and our method were randomly assigned to the left or right side of each comparison video. For each comparison, the participants are asked to choose the result they deem the most accurate or realistic (either the left or right side).

## 8. Visualization

In this part, we present additional visualization results, including qualitative analyses from the Cousin Data Training Strategy (CDTS) ablation study and visual comparisons for speed change scenarios.

As mentioned in Sec. 4.3 of the main text, we perform an ablation study on the CDTs using PVG [9]. For clarity, *DriveDreamer4D* in this ablation study refers to *DriveDreamer4D* with PVG. As shown in Fig. 4, *DriveDreamer4D* demonstrates significant improvement over the baseline methods, regardless of whether CDTs is applied. Notably, the baseline methods struggle to accurately reconstruct the positions of vehicles in novel trajectories, resulting in severe ghosting artifacts. In contrast, *DriveDreamer4D* excels at rendering the vehicle positions with high precision, significantly enhancing rendering performance. Moreover, with the introduction of CDTs, *DriveDreamer4D* further enhances the reconstruction quality of dynamic vehicles, particularly at the edges, providing more detailed and accurate representations.

In Sec. 4.2 of the main text, we analyze the im-

proved visualization effects of *DriveDreamer4D* in lane change scenarios. For more details, please refer to the file videos/lane\_change\_comparison.mp4. More qualitative analysis of novel trajectory view renderings are shown in Fig. 5, focusing on speed change scenarios. Our method significantly enhances the positional accuracy of foreground vehicles and background elements under speed change scenarios. Specifically, baseline results (PVG [9], S<sup>3</sup>Gaussian [28], Deformable-GS [74]) are displayed in rows 1, 3, and 5. It is evident that the baseline methods face challenges with perspective synthesis in speed-change scenarios, resulting in inaccurate positional shifts (such as blurring or disappearance of foreground vehicles). In contrast, the integration of *DriveDreamer4D* enables the 4DGs algorithms to achieve superior spatial consistency and significantly improved rendering quality, as illustrated by the orange boxes in the Fig. 5. More details can be found in the file videos/speed\_change\_comparison.mp4.