ReCamDriving: LiDAR-Free Camera-Controlled Video Synthesis for Novel Trajectories

Li Yaokun[1⋆], Shuaixian Wang[1,3], Mantang Guo[2], Jiehui Huang[4], Taojun Ding[2], Mu Hu[4], Kaixuan Wang[2], Shaojie Shen[4], and Guang Tan[1†]

[1] Sun Yat-sen University
[2] ZhuoYu Technology
[3] Shenzhen Polytechnic University
[4] The Hong Kong University of Science and Technology

Abstract. Synthesizing multi-pass videos is important for autonomous driving. While current repair-based methods often struggle with out-of-distribution artifacts, camera-controlled methods often produce 3D-inconsistent results due to sparse LiDAR cues. We propose ReCamDriving, a purely vision-based framework that achieves camera-controlled generation by leveraging dense, structurally complete 3DGS renderings as geometric guidance. Specifically, to prevent the model from overfitting to a trivial repair solution when conditioning on 3DGS renderings, we adopt a two-stage progressive training paradigm: the first stage uses camera poses for coarse control, while the second stage incorporates 3DGS renderings for fine-grained viewpoint and geometric guidance. Furthermore, to align training and inference camera transformation patterns, we propose a 3DGS-based cross-trajectory data curation strategy, enabling consistent lateral-trajectory supervision from single-pass videos. Based on this strategy, we construct the ParaDrive dataset, containing approximately 110K parallel-trajectory video pairs. Extensive experiments demonstrate that ReCamDriving achieves state-of-the-art camera controllability and structural consistency. Project website: https://recamdriving.github.io/.

Keywords: Video generation · Autonomous driving · Novel trajectory synthesis

1 Introduction

High-quality multi-pass videos is pivotal for autonomous driving, as it provides diverse viewpoints for tasks such as 3D reconstruction [57, 66] and world-model training [9, 65]. However, collecting real-world multi-pass data is costly, requiring multiple synchronized vehicles to capture videos from different trajectories.

† Corresponding author.
⋆ This work was done during Yaokun Li's internship at ZhuoYu Technology.

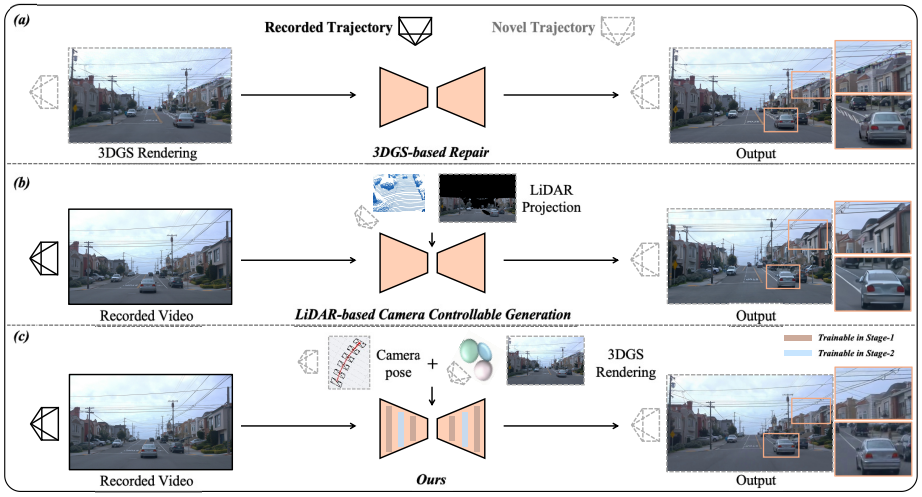

Fig. 1: Comparison of novel-trajectory video synthesis. Repair-based methods (e.g., Difix3D+ [54]) suffer from severe artifacts under large-extrapolation novel viewpoints, while LiDAR-based camera-controlled methods (e.g., StreetCrafter [58]) exhibit inconsistencies in occluded or distant regions due to incomplete geometric cues. In contrast, ReCamDriving leverages dense scene-structure information of novel-trajectory 3DGS renderings to achieve precise camera control and structurally consistent generation.

Consequently, 3D-consistent, high-fidelity novel-trajectory video synthesis from single-pass ego-trajectories offers an attractive, scalable alternative.

A feasible strategy for novel-trajectory synthesis is the reconstruction-then-repair pipeline [13, 29, 54, 60]. It first reconstructs scenes using Neural Radiance Fields (NeRF) [33] or 3D Gaussian Splatting (3DGS) [22], renders novel trajectories, and then trains diffusion-based repair models to restore rendering artifacts. Although effective at artifact removal, this pipeline often fails under complex rendering artifacts (Fig. 1a). Its core limitation is that repair models learn local degraded-to-clean mappings based on training-time artifact patterns, whereas highly varying inference-time artifacts often fall out-of-distribution (OOD), leading to failed restoration and 3D inconsistency.

Another paradigm is camera-controlled video generation [2,64], which synthesizes novel-trajectory videos based on source video and camera conditions. Early works [1, 14, 53] simply condition on camera poses, often leading to imprecise viewpoint control. To enhance accuracy, FreeVS [47] and StreetCrafter [58] incorporate novel-trajectory LiDAR projections to provide explicit geometric cues, yet LiDAR projections are sparse and incomplete in background or occluded regions, often resulting in 3D-inconsistent results (Fig. 1b). Moreover, training camera-transformation models requires ground-truth novel-trajectory supervision, which is unavailable in autonomous driving datasets. Prior works [47, 58] circumvent this by constructing pseudo training pairs from different segments of the single-pass recorded trajectory (Fig. 2a). However, this compromise only cap-

tures longitudinal motion patterns, leading to a train-test gap when generalizing to lateral novel trajectory synthesis at inference time.

To address these limitations, we propose *ReCamDriving*, a purely vision-based framework for 3D-consistent novel-trajectory video synthesis. Our approach redefines camera-controlled generation by replacing sparse LiDAR projections with novel-trajectory 3DGS renderings as the primary control signal. The key insight is that the structural completeness of 3DGS renderings, despite potential rendering artifacts, is more critical for precise camera control and structurally consistent generation.

However, directly conditioning the model on 3DGS renderings introduces a significant risk of shortcut learning. Since the conditioning renderings and the ground-truth videos share the same viewpoint, the model tends to prioritize local artifact restoration over complex novel-trajectory geometric reasoning, thereby collapsing into a trivial artifact-repair solution. To circumvent this, we propose a two-stage progressive training strategy. We first train the model using only camera poses to establish coarse control capability. Subsequently, we freeze these modules and introduce auxiliary attention layers to incorporate 3DGS renderings. This decoupling simplifies novel-trajectory reasoning by interacting rendering tokens with latent features pre-aligned to the target pose. This encourages the model to treat 3DGS renderings as a structural scaffold for guidance rather than a template for restoration.

Furthermore, to bridge the train-test gap of camera transformation patterns, we introduce a 3DGS-based cross-trajectory data curation strategy (Fig. 2). By utilizing novel-trajectory 3DGS renderings as source views and recorded videos as supervision during training, we align the camera-transformation patterns encountered at inference. This strategy enables large-scale lateral-trajectory supervision from monocular videos, allowing us to construct the *ParaDrive* dataset, comprising 110K parallel-trajectory video pairs across 1.6K 3DGS scenes from Waymo Open Dataset (WOD) [40] and NuScenes [7]. This approach facilitates scalable data construction and has the potential to extend camera-controlled models to web-scale video data without requiring LiDAR annotations. Overall, our main contributions are:

- We propose *ReCamDriving*, a vision-based framework leveraging 3DGS renderings for precise camera control and structurally consistent novel-trajectory synthesis.
- We introduce a novel 3DGS-based cross-trajectory data curation strategy for scalable lateral-trajectory supervision and construct the *ParaDrive* dataset with 110K pairs.
- Extensive experiments demonstrate that ReCamDriving achieves the state-of-the-art performance in both camera controllability and 3D consistency.

2 Related Work

2.1 Diffusion Priors for Repairing 3D Renderings.

Recent advances in NeRF [33] and 3DGS [22] have greatly advanced 3D reconstruction [3, 4, 21, 25, 32, 48, 49, 52, 63]. However, under limited viewpoints, these methods still produce noticeable artifacts during novel-view synthesis, especially in extrapolated regions [24, 43, 67]. To mitigate this issue, recent works explore learning diffusion priors to repair degraded 3D renderings. 3DGS-Enhancer [29] fine-tunes a video diffusion model for rendering restoration, Difix3D+ [54] enables real-time neural enhancement, and GSFixer [60] conditions video restoration on both semantic and geometric cues. Freesim [13] introduces a framework for novel-trajectory restoration through a progressive reconstruction strategy. While these methods can improve the visual fidelity of 3D renderings, they essentially address a restoration problem, focusing on local artifact correction rather than consistent scene-level geometry modeling. Consequently, their performance heavily depends on the training data distribution, and since rendering artifacts vary significantly across scenes and viewpoints, they often fail when facing out-of-distribution degradations.

2.2 Camera-Controlled Video Generation.

Camera-controlled generation has attracted increasing attention for synthesizing immersive scenes and novel viewpoints [2, 17, 31, 37, 46, 55, 56, 64]. The key challenge lies in how to represent camera motion and inject it into generative models for precise control. Early works [1,2,14,53,59] condition diffusion models on camera extrinsics or Plücker embeddings, enabling controllable generation but often resulting in inconsistent target-view geometry. From a representation-learning perspective [5, 35, 41, 42], such pose-conditioned models primarily capture *statistical correlations* between camera motion and appearance variation rather than *geometric causality*, leading to spatial misalignment and geometric instability.

To address these issues, recent approaches [61, 62] incorporate 3D point-map priors [18, 23, 50] to inject explicit geometric structure. However, their performance remains limited by the quality of reconstructed point maps in large-scale driving scenes. Alternatively, FreeVS [47] and StreetCrafter [58] leverage LiDAR point clouds for camera control, yet LiDAR data are costly and spatially incomplete, frequently leading to geometrically inconsistent generation results. In contrast, our approach adopts a purely visual formulation for novel-trajectory video generation. By replacing LiDAR with 3DGS renderings as dense structural camera conditions, we achieve fine-grained, geometrically consistent control.

3 Cross-trajectory Data Curation and ParaDrive Dataset

Training a camera-controlled video regeneration model requires ground-truth videos from novel trajectories for supervision. While general datasets [27, 36]

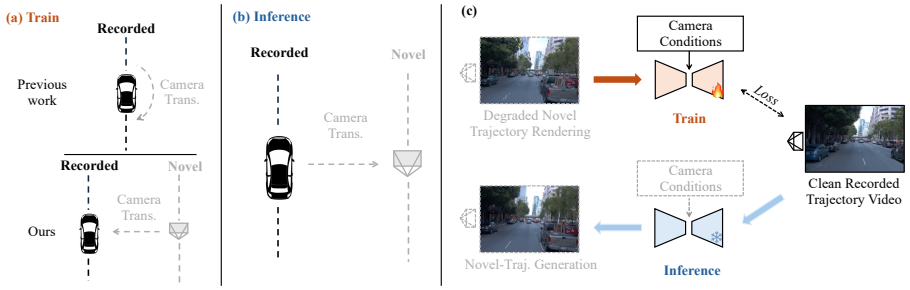

Fig. 2: (a–b) Comparison of training and inference camera-transformation patterns. (c) Our training and inference data strategy. (Trans.: Transformation)

provide such supervision, autonomous driving datasets typically contain only single-pass trajectories. Prior works [47, 58] mitigate this limitation by splitting a single trajectory into source and target segments (Fig. 2a), but this setting models only longitudinal motion, resulting in degraded performance for lateral-view generation during inference.

To address this challenge, we propose a 3DGS-based cross-trajectory data curation strategy that enables supervised learning of lateral camera transformations. Specifically, given an original trajectory video, we first reconstruct a 3DGS representation of the scene and laterally shift the camera trajectory by a fixed offset (e.g., +3 m) to render a novel trajectory. Since the rendered novel-trajectory video contains artifacts and cannot serve as ground truth, we instead use it as the source input during training, while the clean recorded-trajectory video provides the ground-truth supervision (Fig. 2c). At inference, we use the clean recorded video as the source input to generate laterally shifted novel-trajectory views . Although the network is trained with rendered videos as source inputs but tested with clean recorded videos, experiments show that this train–inference source mismatch does not degrade performance. On the contrary, using clean source videos during inference further improves visual quality. Please refer to Sec. 5.3 for more details.

To construct the training pairs, we reconstruct approximately 1.6K 3DGS scenes from WOD [40] and NuScenes [7] using the DriveStudio framework [10]. To enhance the model's robustness to novel-trajectory rendering artifacts during inference, we save intermediate 3DGS checkpoints at 100, 500, and 1,000 iterations. These underfitted models are used to generate recorded-trajectory renderings with varying artifact levels, which serve as structural camera conditions during training. Meanwhile, the fully-optimized 30K-iteration models are used to render eight laterally shifted trajectories (± 1 m, ± 2 m, ± 3 m, and ± 4 m) to serve as source videos. Each trajectory contains three 121-frame clips (front, middle, and rear). In total, the 3DGS reconstruction process consumed 8,240 L20 GPU hours and produced approximately 110K dual-trajectory video pairs. We name this dataset ***ParaDrive***, which will be released to facilitate research on camera-controlled video generation.

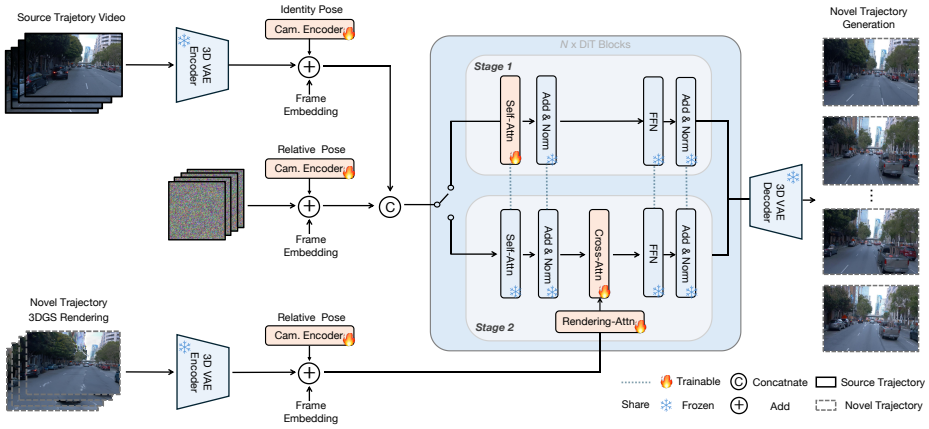

Fig. 3: Overview of our framework. We adopt a two-stage training scheme for precise and structurally consistent novel-trajectory video generation. In *Stage 1*, *ReCamDriving* trains DiT blocks conditioned on the source trajectory video and relative camera (cam.) pose. When switching to *Stage 2*, the original DiT parameters are frozen, and additional attention modules are introduced to integrate 3DGS renderings for fine-grained view control and structural guidance. Shared modules between stages are connected with blue dashed lines.

4 *ReCamDriving*

Given a source trajectory video V_s, our goal is to synthesize a novel trajectory video V_t with lateral offsets. To this end, we adopt a two-stage training strategy that progressively guides the model to perform viewpoint transformation, as illustrated in Fig. 3. In Stage 1, we aim for the network to understand and learn the physical process of viewpoint transformation by using the relative camera pose $\Delta T = T_{t \leftarrow s} \in SE(3)$ between trajectories, enabling it to warp information from the source video to the target trajectory. In Stage 2, we introduce novel-trajectory 3DGS renderings as fine-grained guidance, freeze the first-stage parameters, and add two additional attention modules to incorporate 3DGS renderings for accurate viewpoint and structural generation.

4.1 Preliminary: Latent Diffusion Models with Flow Matching

Latent Diffusion Models (LDMs) [38] perform diffusion in a compact latent space learned by a pretrained autoencoder, achieving high-fidelity generation with reduced computational cost. This balance between efficiency and quality has made LDMs the foundation of many state-of-the-art image and video generators [6, 38, 45]. Given a video $V \in \mathbb{R}^{F \times 3 \times H \times W}$, a 3D VAE encoder projects it into latents $x \in \mathbb{R}^{f \times c \times h \times w}$ with compressed spatial and temporal dimensions. Diffusion and generation are then performed in this latent space, and the decoded outputs reconstruct the final video frames.

Traditional diffusion models [16, 20, 39] formulate generation as a stochastic differential equation (SDE), reversing a predefined noise process but suffering

from training instability and slow inference. Flow Matching [12,28] replaces this stochastic formulation with a deterministic ordinary differential equation (ODE) that learns a time-dependent velocity field to transport samples from noise to data, yielding more stable training and faster sampling. The forward process is typically defined as a straight path between noise x_0 and data x_1:

$$x_t = tx_1 + (1-t)x_0, \quad t \in [0,1]. \tag{1}$$

Differentiating with respect to t gives the corresponding velocity field:

$$v_t = \frac{dx_t}{dt} = x_1 - x_0. \tag{2}$$

In the context of camera-controlled video generation, given a camera condition c_{cam}, the training objective of flow matching can be formulated as:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{x_0,\, x_1,\, c_{\text{cam}},\, t\sim U(0,1)} \left\| \epsilon_\theta(x_t; c_{\text{cam}}, t) - v_t \right\|^2, \tag{3}$$

where ϵ_θ denotes the neural network predicting the velocity field.

4.2 Training Stage 1: Relative Pose-Guided Coarse Camera Control

In this stage, the relative camera pose ΔT is used as a condition to guide the model in re-generating the source-trajectory video. Instead of using absolute poses, relative poses are easier to specify during inference and are more robust to calibration errors. ReCamMaster [2] adopts a similar idea but concatenates the relative pose with both the source latent x_s and the noisy latent x_t, which can introduce ambiguity regarding which latent corresponds to the source or the target view.

To resolve this, we separately encode the relative pose ΔT and the identity pose $T_I = I_4 \in SE(3)$ (representing zero motion) using a camera encoder $\mathcal{E}_{\text{cam}}$, producing $c_r = \mathcal{E}_{\text{cam}}(\Delta T) \in \mathbb{R}^{f\times d}$ and $c_I = \mathcal{E}_{\text{cam}}(T_I) \in \mathbb{R}^{f\times d}$, where d denotes the feature dimension. We further introduce a learnable frame embedding $E_f \in \mathbb{R}^{f\times d}$ that provides frame-wise correspondence cues, enhancing temporal alignment in self-attention operations.

After encoding the source and noisy videos using the 3D VAE encoder, we obtain latent representations $x_s, x_t \in \mathbb{R}^{f\times c\times h\times w}$. A patchify operation unfolds each spatial feature map into a sequence of local tokens, resulting in $x_s, x_t \in \mathbb{R}^{f\times l\times d}$, where $l = h \times w$. The camera and frame embeddings are then added to each latent via broadcasting along the l-dimension:

$$x_i = \text{Cat}(x_t + c_r + E_f,\ x_s + c_I + E_f), \tag{4}$$

where Cat denotes concatenation along the frame dimension f. The resulting sequence x_i is processed by N layers of Diffusion Transformer (DiT) [34] blocks to predict the velocity field. Each DiT block in this stage (see Fig. 3) consists of a self-attention layer, a feed-forward network (FFN), and normalization layers, with only the self-attention parameters unfrozen for adaptation.

We adopt the flow-matching framework to schedule noise levels and optimize the diffusion process. We adopt the flow-matching framework to schedule noise levels and optimize the diffusion process. The training objective follows Eq. 3, where c_{cam} denotes the relative camera-pose embedding and the target data x_1 corresponds to the clean recorded-trajectory videos.

4.3 Training Stage 2: Fine-grained Camera Control via 3DGS Renderings

In this stage, to improve camera control accuracy and structural guidance, besides the relative camera pose, we utilize DriveStudio [10] to reconstruct 3DGS representations and render novel-trajectory 3DGS renderings as additional conditions. The renderings V_{gs} are encoded by the same 3D VAE encoder and patchified to obtain $x_{gs} \in \mathbb{R}^{f \times l \times d}$. We then augment x_{gs} with the relative camera-pose embedding and frame embedding:

$$\bar{x}_{gs} = x_{gs} + c_r + E_f. \tag{5}$$

To integrate 3DGS features effectively, each DiT block introduces two additional components: a *Rendering Attention* and a *Cross Attention*. The *Rendering Attention* serves as an auxiliary self-attention layer that refines spatio–temporal representations within the 3DGS rendering latent space. It shares the same structure as the self-attention used in stage 1 but is named differently to avoid confusion. The *Cross Attention* then fuses the rendering latent features $\bar{x}_{gs}$ with the diffusion latent $\bar{x}_i = \text{SelfAttn}(x_i)$ to achieve geometric alignment between the generated and target trajectories. The self-attention modules from stage 1 remain frozen to ensure that $\bar{x}_i$ has already undergone coarse camera transformations when interacting with the 3DGS rendering features.

The primary advantage of this two-stage design over a single-stage alternative lies in the reduced complexity of geometric alignment. By the time the diffusion latents interact with the 3DGS rendering features, they have already undergone a coarse viewpoint transformation. This significantly reduces the difficulty for the model to align the geometric features of both inputs to infer novel-trajectory geometry, thereby effectively mitigating the tendency to collapse into a trivial restoration of 3DGS rendering artifacts. Training in this stage also follows Eq. 3, but with the condition c_{cam} extended to include both the relative pose embedding and the 3DGS rendering latent. Finally, during inference, the 3D VAE decoder reconstructs the novel-trajectory video V_t from the stage 2 latent representation, achieving precise camera control and structural consistency.

5 Experiments

5.1 Experimental setup

Implementation. We train *ReCamDriving* on the proposed *ParaDrive* dataset in two stages. Each stage is conducted on 64 NVIDIA A100 GPUs for 6,000

Table 1: Quantitative comparison results on WOD [40], where bold indicates the best performance, and underline denotes the second best.

Method	Lateral Offset ±1m					Lateral Offset ±2m				
	Visual Quality		View Consistency			Visual Quality		View Consistency		
	IQ↑	TCE↓	FID↓	FVD↓	CLIP-V↑	IQ↑	TCE↓	FID↓	FVD↓	CLIP-V↑
DriveStudio	52.13	7.93	83.32	25.37	94.78	47.32	8.36	104.24	39.79	94.23
Difix3D+	64.24	7.81	56.35	27.80	95.32	63.11	9.39	57.73	31.88	92.85
FreeVS	62.74	11.27	63.06	37.06	88.99	60.16	12.30	67.87	43.59	88.41
StreetCrafter	63.57	9.72	28.18	20.51	96.01	63.78	10.61	46.78	22.81	94.74
Ours	**65.18**	**3.38**	**13.76**	**13.27**	**97.96**	**65.34**	**3.69**	**25.01**	**14.08**	**97.18**
Method	**Lateral Offset ±3m**					**Lateral Offset ±4m**				
DriveStudio	43.83	9.06	116.12	63.21	90.37	41.47	9.49	144.05	72.50	88.76
Difix3D+	60.88	9.79	66.39	45.23	91.76	58.81	10.49	78.08	65.37	90.12
FreeVS	57.71	13.32	84.87	55.76	86.33	56.15	14.35	107.04	58.39	85.17
StreetCrafter	59.34	11.49	50.75	30.26	92.13	59.89	12.38	68.73	36.67	91.17
Ours	**62.68**	**3.99**	**28.38**	**22.59**	**96.50**	**61.32**	**4.30**	**32.36**	**26.76**	**94.91**

Table 2: Quantitative comparison of camera accuracy on WOD [40]. RErr.: Rotation Error; TErr.: Translation Error.

Method	Offset ±1m		Offset ±2m		Offset ±3m		Offset ±4m	
	RErr.↓	TErr.↓	RErr.↓	TErr.↓	RErr.↓	TErr.↓	RErr.↓	TErr.↓
Difix3D+	1.36	2.42	1.64	2.66	2.01	2.97	2.68	3.12
FreeVS	1.71	2.88	2.12	2.93	3.17	3.78	3.02	3.39
StreetCrafter	1.52	2.53	1.79	2.77	1.91	3.13	2.87	3.03
Ours	**1.32**	**2.37**	**1.45**	**2.43**	**1.63**	**2.65**	**1.57**	**2.73**

steps, with a batch size of 1 and a learning rate of 1e-4. The total training takes approximately 3.5 days. The training resolution is set to 480 × 832, and each video consists of 121 frames. We initialize our model from the Wan2.1 text-to-video foundation model [45], with the text prompt set to an empty string by default. Specifically, the 3D VAE, rendering attention, cross attention, and FFN modules are initialized from Wan2.1, while the camera encoder and self-attention layers are initialized from ReCamMaster [2].

Evaluation Dataset and Metrics. We select 20 scenes each from WOD [40] and NuScenes [7] for validation. We evaluate our method from three perspectives: (1) *Visual Quality*: We use Imaging Quality (IQ) [19] to assess fidelity. For temporal consistency, we follow [11] to measure the Temporal Consistency Error (TCE, lower is better) by computing the Average End-Point Error between consecutive frames via optical flow [51]. Please refer to the supplementary material for further details. (2) *Camera Accuracy*: Following CamCloneMaster [30], we adopt the state-of-the-art (SOTA) camera estimation method MegaSaM [26] to recover camera parameters from the generated videos, and evaluate accuracy

Fig. 4: Qualitative comparisons on WOD [40]. Our method and Difix3D+ [54] use renderings of DriveStudio [10] for camera control and restoration, respectively. Note that the officially released FreeVS [47] model is trained and tested on a cropped resolution that excludes sky regions to reduce computation and avoid LiDAR-sparse areas.

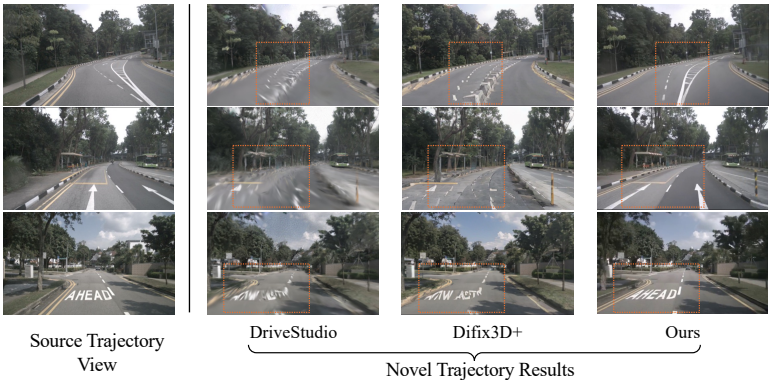

Fig. 5: Qualitative comparison results on NuScenes [7].

using rotation and translation error. (3) *View Consistency*: We compute the Fréchet Video Distance (FVD) [44] and Fréchet Image Distance (FID) [15] between the source and generated video distributions, and further compute the frame-wise CLIP similarity (CLIP-V) to assess cross-view semantic consistency.

Baselines. We compare *ReCamDriving* against state-of-the-art novel trajectory synthesis methods [10, 47, 54, 58]. DriveStudio [10] reconstructs 3DGS using a dynamic neural scene graph for novel-trajectory rendering. Difix3D+ [54] follows the reconstruction–repair paradigm, using a single-step diffusion model to restore 3DGS renderings. In addition, FreeVS [47] and StreetCrafter [58] adopt a camera-controll video generation approach, where the target trajectory is conditioned on colored LiDAR point projections to synthesize novel-trajectory videos.

5.2 Comparison Results

Qualitative Results. FreeVS [47] and StreetCrafter [58] provide pretrained weights only on WOD. For a fair comparison, we train a version of our model on the WOD subset of *ParaDrive*, with qualitative results shown in Fig. 4. We further compare our full model trained on the entire *ParaDrive* with Difix3D+ [54] dataset and DriveStudio [10] on NuScenes [7] (Fig. 5).

ReCamDriving consistently outperforms all baselines. Difix3D+ often produces view-inconsistent corrections on fine structures near the ego vehicle (e.g., lane markings, road text), whereas *ReCamDriving* maintains structural continuity. FreeVS and StreetCrafter exhibit severe inconsistencies under large viewpoint shifts due to sparse LiDAR conditioning. As shown in Fig. 4, StreetCrafter fails to reconstruct complete vehicle geometry in the first scene, while both baselines yield blurred or 3D-inconsistent distant backgrounds caused by occluded or sparsely sampled LiDAR regions. In contrast, *ReCamDriving* leverages dense structural cues from 3DGS renderings for robust view and geometric guidance, enabling more reliable generation of distant content and occluded objects.

Quantitative Results. Quantitative comparisons on WOD and NuScenes are reported in Table 1, Table 2 and Table 3. Since DriveStudio renders results using ground-truth poses, we do not evaluate its camera accuracy. As shown, *ReCamDriving* consistently outperforms all baselines across all metrics. Notably, as the lateral offset increases, the view consistency of FreeVS [47], StreetCrafter [58], and Difix3D+ [54] degrades sharply, whereas *ReCamDriving* remains stable, demonstrating superior robustness under large viewpoint shifts.

Table 3: Average quantitative results of novel-trajectory generation on NuScenes [7].

Method	Visual Quality		View Consistency		
	IQ↑	TCE↓	FID↓	FVD↓	CLIP-V↑
DriveStudio	45.39	<u>8.63</u>	103.25	52.93	89.19
Difix3D+	<u>61.76</u>	9.14	<u>65.14</u>	<u>41.37</u>	<u>90.48</u>
Ours	**62.38**	**3.79**	**25.68**	**18.98**	**96.14**

5.3 Ablation Studies

Effectiveness of 3DGS Rendering Condition. In Stage 2, 3DGS renderings of the target trajectory are used as conditions for fine-grained camera control. We ablate it with three variants: (1) without Stage 2 (*Pose* only); (2) Stage 2 with LiDAR projection (*Pose + LiDAR*); (3) Stage 2 with both LiDAR projection and 3DGS renderings (*Pose + LiDAR + GS*). LiDAR projections are generated following StreetCrafter [58] and injected into the Rendering Attention branch. Qualitative and quantitative results are presented in Fig. 6 and Table 4. The results demonstrate that using only camera poses often leads to inaccurate camera control. While incorporating LiDAR improves pose accuracy, it still suffers

from geometric inconsistencies under large offsets due to the sparsity of LiDAR data. In contrast, our 3DGS-conditioned model effectively mitigates these issues. Furthermore, introducing LiDAR conditions on top of our method does not yield significant improvements in overall metrics, and it incurs high costs for LiDAR data acquisition.

Fig. 6: Qualitative ablation of camera conditions on WOD [40].

Table 4: Quantitative ablation of camera conditions on WOD [40].

Camera Condition	IQ↑	FID↓	FVD↓	RErr.↓	TErr.↓
Pose	60.13	34.86	32.31	3.01	4.23
Pose + LiDAR	61.32	31.23	27.78	1.53	2.69
Pose + LiDAR + GS	<u>63.42</u>	**24.75**	<u>19.27</u>	**1.41**	**2.47**
Pose + GS (Ours)	**63.63**	<u>24.88</u>	**19.18**	<u>1.49</u>	<u>2.55</u>

Comparison with the repair baseline. We further compare our adopted camera-controlled video generation paradigm with a repair-based baseline. This baseline shares an identical architecture with our framework (Fig. 3) but excludes the Rendering Attention branch and camera pose injection. Specifically, it takes rendered videos as input and produces clean videos as output. For training, we construct pairs on recorded trajectories using blurred 3DGS renderings as inputs and clean videos as supervision. Both methods are evaluated on WOD [40], with results in Table 5 and Fig. 7. As shown, the repair baseline often fails to correct artifacts, as these artifacts are not well covered by its training distribution.

Ablation on the Training Strategies. We employ a two-stage strategy to ensure that 3DGS renderings enhance camera control rather than artifact repair. To evaluate its effectiveness, we compare it with a single-stage scheme where all modules are trained jointly without freezing on the full *ParaDrive* dataset. Quantitative and qualitative results are shown in Table 6 and Fig. 8. The single-stage model tends to produce artifacts and exhibits degraded visual

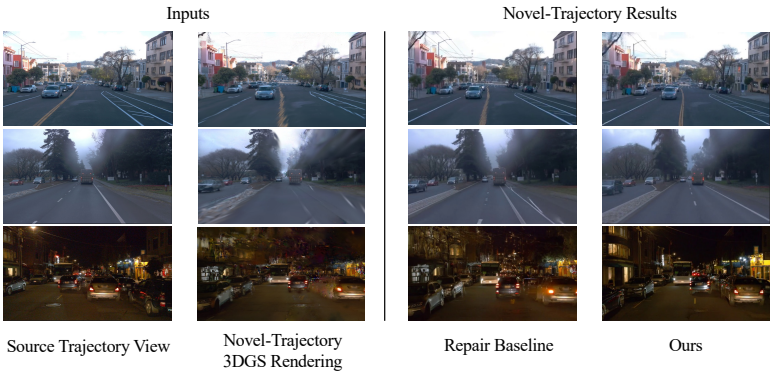

Fig. 7: Qualitative ablation of training paradigms on WOD [40]. Our method utilizes the two types of information shown on the left as input. The repair baseline uses novel-trajectory 3DGS renderings as input.

Table 5: Qualitative ablation of training paradigms on WOD [40].

Method	IQ↑	TCE↓	FID↓	FVD↓	CLIP-V↑
Repair Baseline	62.16	4.57	40.87	31.44	91.32
Ours	**63.63**	**3.84**	**24.88**	**19.18**	**96.64**

performance similar to repair-based methods, whereas the two-stage strategy yields clearer and more 3D-consistent results, confirming its effectiveness.

Ablation on Cross-Trajectory Data Curation. We propose a cross-trajectory data curation strategy to align camera-transformation modes between training and inference. To evaluate its effectiveness, we train a baseline using longitudinal transformations following FreeVS [47], where the first and last 121 frames of each recorded trajectory are used as source and supervision, respectively. Table 7 presents the comparison results, showing that our cross-trajectory strategy significantly improves both camera accuracy and view consistency for novel-trajectory prediction, confirming its effectiveness.

Impact of Train–Test Source-Video Mismatch. During training, we use blurred 3DGS renderings as source videos, while at test time, the source is the clean recorded videos. To examine the impact of this mismatch, we compare inference using either clean recorded videos or their degraded 3DGS-rendered counterparts as sources. As shown in Fig. 9, clean sources produce noticeably sharper backgrounds than blurred ones, indicating that clean source videos at test time do not cause degradation but instead improve visual quality. Furthermore, the high sensitivity of outputs to source inputs confirms that our model performs camera-control video generation rather than merely restoring 3DGS rendering conditions, further distinguishing it from repair-based methods.

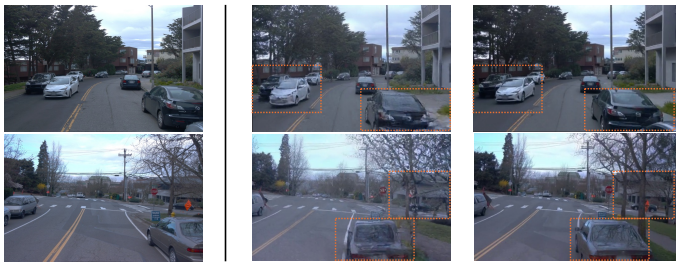

Fig. 8: Qualitative ablation on our two-stage training strategy.

Table 6: Quantitative ablation on our two-stage training strategy on WOD [40] and NuScenes [7].

Training strategy	IQ↑	FID↓	FVD↓	CLIP-V↑
One-stage	59.97	32.64	25.16	94.78
Two-stage (Ours)	**63.42**	**25.13**	**18.32**	**96.32**

Table 7: Ablation of our data curation strategy on WOD [40].

Camera Transformation	RErr.↓	TErr.↓	FID↓	FVD↓
Longitudinal	1.97	3.02	34.17	28.54
Lateral (Ours)	**1.49**	**2.55**	**24.88**	**19.18**

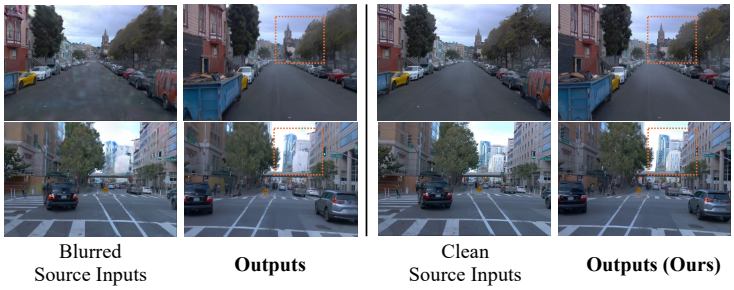

Fig. 9: Ablation on different source inputs at inference.

6 Discussion and Limitation

While our method achieves superior performance and eliminates expensive LiDAR acquisition costs, it requires an offline 3DGS pre-processing step. However, this reconstruction is a one-time overhead that supports multiple trajectory generations. Importantly, our framework remains agnostic to the specific reconstruction methodology, meaning that as feed-forward 3D reconstruction [8] techniques continue to evolve, the pre-processing cost can be reduced from hours to seconds.

Beyond this overhead, the primary significance of our paradigm lies in offering a scalable pathway to leverage vast, internet-scale datasets that lack expensive LiDAR annotations.

7 Conclusion

In this work, we introduced *ReCamDriving*, a pure vision-based framework for controllable novel-trajectory video generation. By leveraging 3DGS renderings as structural camera conditions and adopting a two-stage training paradigm, our model achieves precise camera control and consistent geometry. We also construct the large-scale *ParaDrive* dataset using the proposed cross-trajectory data curation strategy, which enables scalable lateral-trajectory supervision from single-pass driving videos. Extensive experiments show that *ReCamDriving* achieves higher camera accuracy and visual fidelity than existing baselines.

Acknowledgements

The study was supported by the Shenzhen Basic Research Fund under grant JCYJ20241202130025030.

References

1. Bahmani, S., Skorokhodov, I., Qian, G., Siarohin, A., Menapace, W., Tagliasacchi, A., Lindell, D.B., Tulyakov, S.: Ac3d: Analyzing and improving 3d camera control in video diffusion transformers. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 22875–22889 (2025)
2. Bai, J., Xia, M., Fu, X., Wang, X., Mu, L., Cao, J., Liu, Z., Hu, H., Bai, X., Wan, P., et al.: Recammaster: Camera-controlled generative rendering from a single video. arXiv preprint arXiv:2503.11647 (2025)
3. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5855–5864 (2021)
4. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5470–5479 (2022)
5. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence **35**(8), 1798–1828 (2013)
6. Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al.: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023)
7. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020)

8. Chen, X., Xiong, Z., Chen, Y., Li, G., Wang, N., Luo, H., Chen, L., Sun, H., Wang, B., Chen, G., et al.: Dggt: Feedforward 4d reconstruction of dynamic driving scenes using unposed images. arXiv preprint arXiv:2512.03004 (2025)
9. Chen, Y., Wang, Y., Zhang, Z.: Drivinggpt: Unifying driving world modeling and planning with multi-modal autoregressive transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 26890–26900 (2025)
10. Chen, Z., Yang, J., Huang, J., de Lutio, R., Esturo, J.M., Ivanovic, B., Litany, O., Gojcic, Z., Fidler, S., Pavone, M., Song, L., Wang, Y.: Omnire: Omni urban scene reconstruction. In: The Thirteenth International Conference on Learning Representations (2025)
11. Duan, H., Yu, H.X., Chen, S., Fei-Fei, L., Wu, J.: Worldscore: A unified evaluation benchmark for world generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 27713–27724 (2025)
12. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: Forty-first international conference on machine learning (2024)
13. Fan, L., Zhang, H., Wang, Q., Li, H., Zhang, Z.: Freesim: Toward free-viewpoint camera simulation in driving scenes. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 12004–12014 (2025)
14. He, H., Xu, Y., Guo, Y., Wetzstein, G., Dai, B., Li, H., Yang, C.: Cameractrl: Enabling camera control for text-to-video generation. arXiv preprint arXiv:2404.02101 (2024)
15. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
16. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
17. Hu, T., Zhang, J., Yi, R., Wang, Y., Huang, H., Weng, J., Wang, Y., Ma, L.: Motionmaster: Training-free camera motion transfer for video generation. arXiv preprint arXiv:2404.15789 (2024)
18. Hu, W., Gao, X., Li, X., Zhao, S., Cun, X., Zhang, Y., Quan, L., Shan, Y.: Depthcrafter: Generating consistent long depth sequences for open-world videos. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 2005–2015 (2025)
19. Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., et al.: Vbench: Comprehensive benchmark suite for video generative models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21807–21818 (2024)
20. Hyvärinen, A., Dayan, P.: Estimation of non-normalized statistical models by score matching. Journal of Machine Learning Research **6**(4) (2005)
21. Jiang, Y., Hedman, P., Mildenhall, B., Xu, D., Barron, J.T., Wang, Z., Xue, T.: Alignerf: High-fidelity neural radiance fields via alignment-aware training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 46–55 (2023)
22. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023)
23. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with mast3r. In: European Conference on Computer Vision. pp. 71–91. Springer (2024)
24. Li, Y., Gou, C., Tan, G.: Taming uncertainty in sparse-view generalizable nerf via indirect diffusion guidance. arXiv preprint arXiv:2402.01217 **3** (2024)

25. Li, Y., Wang, S., Tan, G.: Id-nerf: Indirect diffusion-guided neural radiance fields for generalizable view synthesis. Expert Systems with Applications **266**, 126068 (2025)
26. Li, Z., Tucker, R., Cole, F., Wang, Q., Jin, L., Ye, V., Kanazawa, A., Holynski, A., Snavely, N.: Megasam: Accurate, fast and robust structure and motion from casual dynamic videos. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 10486–10496 (2025)
27. Ling, L., Sheng, Y., Tu, Z., Zhao, W., Xin, C., Wan, K., Yu, L., Guo, Q., Yu, Z., Lu, Y., et al.: Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22160–22169 (2024)
28. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2022)
29. Liu, X., Zhou, C., Huang, S.: 3dgs-enhancer: Enhancing unbounded 3d gaussian splatting with view-consistent 2d diffusion priors. Advances in Neural Information Processing Systems **37**, 133305–133327 (2024)
30. Luo, Y., Bai, J., Shi, X., Xia, M., Wang, X., Wan, P., Zhang, D., Gai, K., Xue, T.: Camclonemaster: Enabling reference-based camera control for video generation. arXiv preprint arXiv:2506.03140 (2025)
31. Ma, Y., Feng, K., Hu, Z., Wang, X., Wang, Y., Zheng, M., He, X., Zhu, C., Liu, H., He, Y., et al.: Controllable video generation: A survey. arXiv preprint arXiv:2507.16869 (2025)
32. Martin-Brualla, R., Radwan, N., Sajjadi, M.S., Barron, J.T., Dosovitskiy, A., Duckworth, D.: Nerf in the wild: Neural radiance fields for unconstrained photo collections. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7210–7219 (2021)
33. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
34. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4195–4205 (2023)
35. Qian, R., Meng, T., Gong, B., Yang, M.H., Wang, H., Belongie, S., Cui, Y.: Spatiotemporal contrastive video representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6964–6974 (2021)
36. Reizenstein, J., Shapovalov, R., Henzler, P., Sbordone, L., Labatut, P., Novotny, D.: Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10901–10911 (2021)
37. Ren, X., Shen, T., Huang, J., Ling, H., Lu, Y., Nimier-David, M., Müller, T., Keller, A., Fidler, S., Gao, J.: Gen3c: 3d-informed world-consistent video generation with precise camera control. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 6121–6132 (2025)
38. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
39. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020)

40. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2446–2454 (2020)
41. Tang, Z., Chen, G., Chen, S., Yao, J., You, L., Chen, C.Y.C.: Modal-nexus auto-encoder for multi-modality cellular data integration and imputation. Nature Communications **15**(1), 9021 (2024)
42. Tang, Z., Yang, H., Chen, C.Y.C.: Weakly supervised posture mining for fine-grained classification. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 23735–23744 (2023)
43. Truong, P., Rakotosaona, M.J., Manhardt, F., Tombari, F.: Sparf: Neural radiance fields from sparse and noisy poses. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4190–4200 (2023)
44. Unterthiner, T., Van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., Gelly, S.: Towards accurate generative models of video: A new metric & challenges. arXiv preprint arXiv:1812.01717 (2018)
45. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., et al.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)
46. Wang, Q., Luo, Y., Shi, X., Jia, X., Lu, H., Xue, T., Wang, X., Wan, P., Zhang, D., Gai, K.: Cinemaster: A 3d-aware and controllable framework for cinematic text-to-video generation. In: Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers. pp. 1–10 (2025)
47. Wang, Q., Fan, L., Wang, Y., Chen, Y., Zhang, Z.: Freevs: Generative view synthesis on free driving trajectory. arXiv preprint arXiv:2410.18079 (2024)
48. Wang, S., Li, Y., Guo, C., Tan, G.: Learning hierarchical uncertainty from hybrid representations for neural active reconstruction. Pattern Recognition p. 112493 (2025)
49. Wang, S., Xu, H., Li, Y., Chen, J., Tan, G.: Ie-nerf: Inpainting enhanced neural radiance fields in the wild. arXiv preprint arXiv:2407.10695 (2024)
50. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20697–20709 (2024)
51. Wang, Y., Lipson, L., Deng, J.: Sea-raft: Simple, efficient, accurate raft for optical flow. In: European Conference on Computer Vision. pp. 36–54. Springer (2024)
52. Wang, Y., Wang, C., Gong, B., Xue, T.: Bilateral guided radiance field processing. ACM Transactions on Graphics (TOG) **43**(4), 1–13 (2024)
53. Wang, Z., Yuan, Z., Wang, X., Li, Y., Chen, T., Xia, M., Luo, P., Shan, Y.: Motionctrl: A unified and flexible motion controller for video generation. In: ACM SIGGRAPH 2024 Conference Papers. pp. 1–11 (2024)
54. Wu, J.Z., Zhang, Y., Turki, H., Ren, X., Gao, J., Shou, M.Z., Fidler, S., Gojcic, Z., Ling, H.: Difix3d+: Improving 3d reconstructions with single-step diffusion models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 26024–26035 (2025)
55. Xiao, F., Liu, X., Wang, X., Peng, S., Xia, M., Shi, X., Yuan, Z., Wan, P., Zhang, D., Lin, D.: 3dtrajmaster: Mastering 3d trajectory for multi-entity motion in video generation. In: The Thirteenth International Conference on Learning Representations (2024)

56. Xiao, Z., Ouyang, W., Zhou, Y., Yang, S., Yang, L., Si, J., Pan, X.: Trajectory attention for fine-grained video motion control. arXiv preprint arXiv:2411.19324 (2024)
57. Yan, Y., Lin, H., Zhou, C., Wang, W., Sun, H., Zhan, K., Lang, X., Zhou, X., Peng, S.: Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In: European Conference on Computer Vision. pp. 156–173. Springer (2024)
58. Yan, Y., Xu, Z., Lin, H., Jin, H., Guo, H., Wang, Y., Zhan, K., Lang, X., Bao, H., Zhou, X., et al.: Streetcrafter: Street view synthesis with controllable video diffusion models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 822–832 (2025)
59. Yang, S., Hou, L., Huang, H., Ma, C., Wan, P., Zhang, D., Chen, X., Liao, J.: Direct-a-video: Customized video generation with user-directed camera movement and object motion. In: ACM SIGGRAPH 2024 Conference Papers. pp. 1–12 (2024)
60. Yin, X., Zhang, Q., Chang, J., Feng, Y., Fan, Q., Yang, X., Pun, C.M., Zhang, H., Cun, X.: Gsfixer: Improving 3d gaussian splatting with reference-guided video diffusion priors. arXiv preprint arXiv:2508.09667 (2025)
61. YU, M., Hu, W., Xing, J., Shan, Y.: Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models. arXiv preprint arXiv:2503.05638 (2025)
62. Yu, W., Xing, J., Yuan, L., Hu, W., Li, X., Huang, Z., Gao, X., Wong, T.T., Shan, Y., Tian, Y.: Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. arXiv preprint arXiv:2409.02048 (2024)
63. Yu, Z., Chen, A., Huang, B., Sattler, T., Geiger, A.: Mip-splatting: Alias-free 3d gaussian splatting. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 19447–19456 (2024)
64. Zhang, D.J., Paiss, R., Zada, S., Karnad, N., Jacobs, D.E., Pritch, Y., Mosseri, I., Shou, M.Z., Wadhwa, N., Ruiz, N.: Recapture: Generative video camera controls for user-provided videos using masked video fine-tuning. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 2050–2062 (2025)
65. Zhao, G., Ni, C., Wang, X., Zhu, Z., Zhang, X., Wang, Y., Huang, G., Chen, X., Wang, B., Zhang, Y., et al.: Drivedreamer4d: World models are effective data machines for 4d driving scene representation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 12015–12026 (2025)
66. Zhou, J., Fan, L., Huang, L., Shi, X., Liu, S., Zhang, Z., Li, H.: Flexdrive: Toward trajectory flexibility in driving scene gaussian splatting reconstruction and rendering. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 1549–1558 (2025)
67. Zou, Z., Cheng, W., Cao, Y.P., Huang, S.S., Shan, Y., Zhang, S.H.: Sparse3d: Distilling multiview-consistent diffusion for object reconstruction from sparse views. In: Proceedings of the AAAI conference on artificial intelligence. vol. 38, pp. 7900–7908 (2024)