XLD: A Cross-Lane Dataset for Benchmarking Novel Driving View Synthesis

Hao Li[1,2,*] Chenming Wu[2,*] Ming Yuan[2] Yan Zhang[2] Chen Zhao[2] Chunyu Song[2]
Haocheng Feng[2] Errui Ding[2] Dingwen Zhang[1,✉] Jingdong Wang[2]

[1] BRAIN Lab, NWPU [2] Baidu VIS

Abstract

Comprehensive testing of autonomous systems through simulation is essential to ensure the safety of autonomous driving vehicles. This requires the generation of safety-critical scenarios that extend beyond the limitations of real-world data collection, as many of these scenarios are rare or rarely encountered on public roads. However, evaluating most existing novel view synthesis (NVS) methods relies on sporadic sampling of image frames from the training data, comparing the rendered images with ground-truth images. Unfortunately, this evaluation protocol falls short of meeting the actual requirements in closed-loop simulations. Specifically, the true application demands the capability to render novel views that extend beyond the original trajectory (such as cross-lane views), which are challenging to capture in the real world. To address this, this paper presents a synthetic dataset for novel driving view synthesis evaluation, which is specifically designed for autonomous driving simulations. This unique dataset includes testing images captured by deviating from the training trajectory by $1-4$ meters. It comprises six sequences that cover various times and weather conditions. Each sequence contains 450 *training images,* 120 *testing images, and their corresponding camera poses and intrinsic parameters. Leveraging this novel dataset, we establish the first realistic benchmark for evaluating existing NVS approaches under front-only and multicamera settings. The experimental findings underscore the significant gap in current approaches, revealing their inadequate ability to fulfill the demanding prerequisites of cross-lane or closed-loop simulation. Our dataset and code are released publicly on the project page: https://3d-aigc.github.io/XLD.*

1. Introduction

Autonomous driving (AD) simulation, which bridges the gap between the real and virtual worlds, is essential for testing and developing autonomous driving software in vehicles [7]. Research indicates that employing effective simulation methods can significantly expedite the evaluation of safety tests for autonomous driving, achieving a speedup of approximately 10^3 to 10^5 times faster than real-world testing [21]. This compelling evidence underscores the importance of leveraging simulation to enhance the efficiency and effectiveness of autonomous driving development. However, the self-driving industry primarily conducts system testing using two approaches: log replay, which involves testing on pre-recorded real-world sensor data, and real-world driving, where new miles are driven to gather additional data for testing purposes [61]. In closed-loop simulation, the vehicle must be free to respond to control commands within the simulation environment rather than strictly following the original trajectory from logs. To promote the rapid advancement of end-to-end autonomous driving systems [24, 64], designing a neural simulator for AD simulation [55, 57, 61], which can render photo-realistic images on novel views for closed-loop simulation and algorithm training, is in high demands. The main scientific problems boil down to the 3D reconstruction [35], and novel view synthesis (NVS) [57, 59], which are also long-standing problems in computer vision and computer graphics. Traditional methods such as [43, 44] have dominated the major deployment of 3D scene reconstruction for a long time. However, those reconstructed scenes cannot be directly used to produce photo-realistic novel views, thus imposing large restrictions on sensor simulations. As a result, the industrial bridges the sim-to-real gap by parametric and procedural modeling technique [50] or human-involved creations. With the recent rapid development of 3D implicit fields, such as neural radiance field (NeRF [37]) and explicit primitive representations, i.e., 3D Gaussian Splatting (3DGS [26]), reconstructing a scene from a collection of images serves as the foundation of end-to-end autonomous driving simulation [23]. These techniques enable the rendering of high-quality and photorealistic images on novel views.

Presently, most approaches evaluate the performance of NVS results by splitting the dataset into training and testing sets. However, this strategy of splitting and sampling leads

*Equal contribution.

to an interpolation benchmark, which we argue is insufficient for evaluating whether the trained models can effectively render simulation-ready (*i.e.,* cross-lane) and high-fidelity data for closed-loop simulations. On the contrary, our proposed XLD dataset is a brand-new benchmark that evaluates the synthesis quality on a cross-lane view with additionally captured GT images. Our dataset and benchmark focus on **assessing the NVS capability specifically for cameras in cross-lane scenes**. The primary objective is to evaluate the performance of cameras in generating accurate and realistic novel views in scenarios involving multiple lanes. Specifically, we introduce the XLD dataset, which encompasses the generation of 150×3 rendered images for each scene. Furthermore, we evaluate novel cross-lane view synthesis by rendering 30 images with deviations of 0m, 1m, 2m, and 4m from the training trajectory. Using the XLD dataset, we conduct a benchmark of leading methods, which are based on either NeRF or 3DGS, using well-established NVS metrics. Our benchmark results demonstrate that the proposed dataset offers a comprehensive evaluation benchmark tailored specifically to the requirements of closed-loop simulation. Furthermore, our benchmarking results reveal intriguing findings, emphasizing the value of the proposed dataset.

2. Related Work

2.1. Autonomous Driving Simulation

In the past few years, there has been a surge in the use of autonomous driving simulations [31]. These simulators are instrumental in validating planning and control mechanisms, producing educational and evaluative datasets, and significantly cutting down the time needed to perform these functions. The current landscape is dominated by two predominant categories of simulation tools: model-based and data-driven. Model-based simulation platforms, such as PyBullet [18] MuJoCo [49], AirSim [45] and CARLA [19], utilize advanced computer graphics to replicate vehicles and their surroundings. However, the manual effort required to construct these models and program the vehicles' dynamics can be quite demanding and lengthy. Moreover, the visual output may sometimes fall short of the necessary realism, which can adversely affect the efficacy of perception systems when they are put into operation.

Previously, NVS heavily relied on conventional image processing techniques. For instance, Chaurasia et al.[12] propose using depth synthesis from over-segmented graph structures. At the same time, AADS[30] employs filtered and completed dense depth maps for warping novel view images through image stitching. A data-driven simulation platform VISTA [2, 3] leverages datasets from the real world to create comprehensively labeled and photorealistic simulations. Recently, a wave of innovations has employed the NeRF method to simulate driving perspectives superficially. These new approaches excel in creating photorealistic images and have been shown to surpass traditional view synthesis algorithms in the realm of autonomous driving simulation. Recent advances in neural novel view synthesis significantly accelerate the rapid development of the next-gen driving simulation, which exhibits superior expressiveness and flexibility compared to traditional methods. Our dataset and benchmark are specifically designed for those methods.

2.2. NeRF-based NVS for Driving Simulation

The introduction of neural radiance field (NeRF) revolutionized NVS by incorporating coordinate-based representation within multilayer perceptron (MLP) architectures, leading to significant performance improvements. Building upon NeRF, numerous subsequent works have further adapted these algorithms to fulfill requirements such as efficient training, anti-aliasing rendering, large-scale reconstruction, *etc.*. InstantNGP [38] proposes using a multi-resolution hash grid with a shallow MLP network to eliminate large MLP networks. Mip-NeRF [4, 5] uses anti-aliased conical frustums instead of rays to reduce objectionable aliasing artifacts, which enables NeRF to represent fine details. Zip-NeRF [6] borrows the ideas from rendering and signal processing that combine Mip-NeRF with InstantNGP. Nerfacto [48] integrates many advantages of existing methods to provide an all-in-one solution for NeRF training. Block-NeRF [47] tackles the reconstruction of large-scale urban scenes by division. To handle dynamics, NSG [41] decomposes dynamic scenes into scene graphs and learns a structured representation. SUDS [51] factorizes a large scene into three hash table data structures, encoding static, dynamic, and far-field radiance fields. MARS [57] is an instance-aware and modular simulator based on NeRF, which models dynamic foreground instances and static background environments separately. UniSim [61] transforms a recorded log into a realistic closed-loop multi-sensor simulation, which incorporates dynamic object priors and utilizes a convolutional network to handle and complete unseen regions. EmerNeRF [60] employs a self-bootstrapping approach to simultaneously capture scene geometry, appearance, motion, and semantics, which enables comprehensive and synchronized modeling of these elements by stratifying scenes into static and dynamic fields. UC-NeRF [15] addresses the challenge of under-calibrated multi-view novel view synthesis through layer-based color correction and virtual warping techniques.

2.3. 3DGS-based NVS for Driving Simulation

Motivated by the NeRF-based methods and point-based differentiable rendering [1, 32, 42, 62], 3D Gaussian Splatting (3DGS) [26] opens a new era with the leading advantages

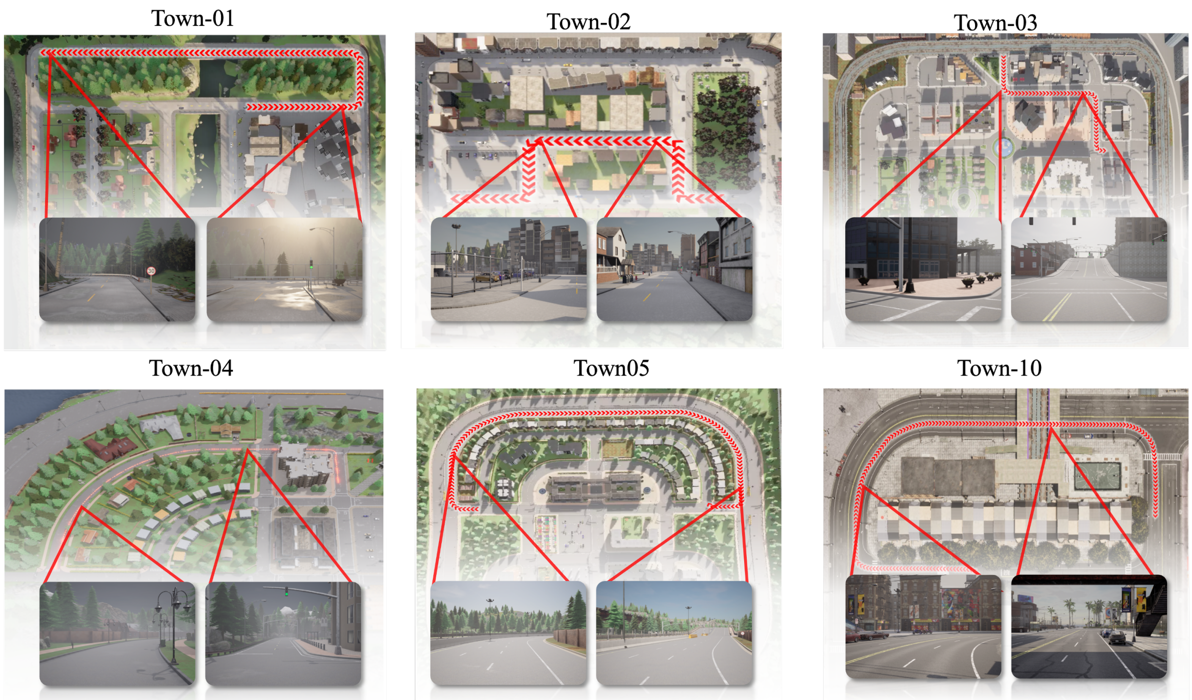

Figure 1. Our datasets encompass six distinct scenes, each involving the vehicle following an on-road trajectory. To generate training data for the cameras and LiDAR sensor, we sample 150 waypoints along each trajectory. The trajectory is visually emphasized using the color red.

in explicit representation and real-time rendering capability. Within a concise timeframe, numerous methods [13, 59] have emerged that focus on road scene reconstruction and NVS by leveraging the 3DGS representation. For instance, PVG [13, 29] introduces periodic vibration-based temporal dynamics to reconstruct dynamic urban scenes. StreetGaussian [59] models the dynamic urban street environment as a collection of point clouds with semantic logits and 3D Gaussians, each associated with either a foreground vehicle or the background. DrivingGaussian [65] uses incremental static 3D Gaussians to represent the scene's static background. It also employs a composite dynamic Gaussian graph to handle multiple moving objects with LiDAR data. HO-Gaussian [33] introduces a hybrid method that combines radiance fields with 3DGS representation, eliminating the requirement for point initialization in urban scene NVS. HGS-Mapping [56] proposes a hybrid Gaussian representation specifically designed for performing online dense mapping in unbounded large-scale scenes. GaussianPro [16] utilizes priors from reconstructed scene geometries and patch-matching techniques to generate precise Gaussians, leveraging the scene's existing structure. DC-Gaussian [53] introduces adaptive image decomposition for modeling reflections and occlusions. It incorporates illumination-aware obstruction modeling to handle reflections and occlusions under varying lighting conditions in urban scene novel view synthesis. Our dataset and benchmark specifically focus on evaluating the performance of neural-based driving simulation in NVS, particularly in cross-lane scenarios. A portion of the mentioned methods serve as our baselines, taking into account the code availability. Moreover, generalizable 3D-GS methods such as PixelSplat [11], Mvsplat [14], and GGRt [28] also attempts to synthesize novel images within a well-trained generalizable feed-forward Gaussian networks.

2.4. Datasets in Autonomous Driving

In autonomous driving training and benchmarking, there are many datasets available. For example, KITTI [22]. KITTI-360 [34], vKITTI [8], CityScapes [17], Mapillary [39], ApolloScape [25], Waymo Open Dataset [46]. nuScenes [9], Argoverse [10] and Argoverse 2 [54], BDD100K [63], OpenLane-V2 [52], LiDAR-CS [20], *etc.*. Those previous works have laid the groundwork for research and development in autonomous driving algorithms. A comprehensive survey of datasets related to autonomous driving refers to [36]. More recently, a newly proposed Hierarchical 3D-GS [27] showcases the scalability of 3D-GS followed by a short scene with parallel trajectories to train and render. However, none of these existing researches have

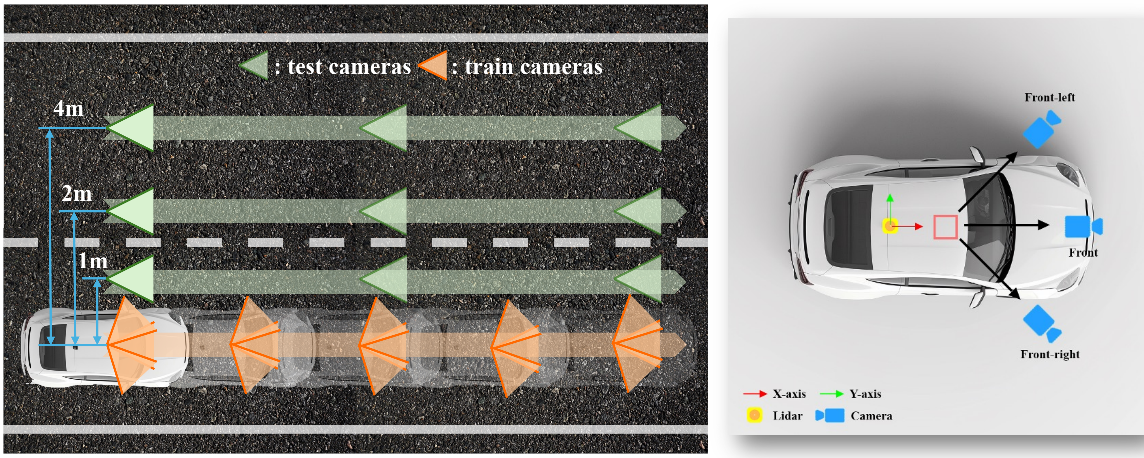

Figure 2. The composition of our training set and testing set. The training set consists of three RGB cameras ('front', 'left-front', and 'right-front') mounted on our vehicle lane. We sample image sequences along trajectories that run parallel to the vehicle's route for the test set. This encompasses four distinct test trajectories, each offset from the vehicle's trajectory by 0 meters, 1 meter, 2 meters, and 4 meters, respectively. The sampling interval for the test set is five times the sampling interval used for the training set.

specifically addressed the evaluation of novel view synthesis techniques tailored for autonomous driving simulation, particularly in terms of their ability to meet the high demands of cross-lane capability.

3. Dataset

Distinguishing our dataset from previous datasets (both real-world and synthetic) that typically capture only a single road trajectory, ours additionally captures multiple parallel trajectories. To this end, we need to generate cross-lane data in the created worlds; here, we utilize Carla [19], an autonomous driving simulator platform built on Unreal Engine.

3.1. Sensor Setup

To meet the needs of most NeRF and 3D-GS algorithms, the sensors used to capture data include three color cameras (*i.e.* 'left-front', 'front', 'right-front') and one 3D laser scanner. The spatial relationships between sensors and the vehicle are fixed as shown in Fig. 2. All three RGB cameras share identical intrinsic, lens parameters. Specifically, they have a resolution of 1920×1280, a field of view (FOV) of $49.5° \times 36.7°$, a sensitivity of ISO 100, and a shutter speed of 5ms. The 3D laser scanner features 64 laser beams, a scanning range of 60 meters, and laser beam angles ranging from $-30°$ to $30°$ in the vertical direction.

3.2. Data Generation

Our simulation environment consists of six scenes ("Town01," "Town02," "Town03," "Town04," "Town05," and "Town10", all of them are under CC-BY License provided by [19]) with various weather conditions, such as sunshine and rain, closely resembling real-world settings. An overview of all the scenes and the trajectories is shown in Fig. 1. For example, the training set samples 150 times in one scene to capture the images of three cameras, LiDAR points, and the vehicle's extrinsic information. The vehicle's forward distance between the two sample points is 2 meters. For the evaluation process, in contrast to datasets captured in real-world scenarios, our cameras are configured with identical parameters, ensuring uniform image quality across all cameras. Therefore, we only test the novel-view-synthesis in the view of *front* camera, which includes four groups. These images are aligned parallel to the training set, with each group exhibiting a progressive deviation of 0m, 1m, 2m, and 4m along the y-axis in vehicle coordinate, as shown in Fig. 2. Additionally, a few novel-view-synthesis for AD methods like Gaussian-Pro [16], MARS [57], and UC-NeRF [15] need annotated sky masks to split the scene into foreground-sky and model the color compensation separately. We employ a pre-trained SegFormer [58] to effectively infer semantic segmentation masks and extract sky masks from them.

4. Benchmark

4.1. Benchmarking Environment

To comprehensively evaluate the performance and computational efficiency of the assessment methods, we conducted a series of experiments using an NVIDIA Tesla V100 16GB GPU. We benchmarked the selected methods across five different cities that include various driving scenarios. Our findings are presented through both qualitative and quantitative

Table 1. Results on our proposed dataset with the different offsets using *front-only* camera. ↑: higher is better, ↓: lower is better. The red, orange, and yellow colors respectively denote the best, the second best, and the third best results.

Method	*w/o* Offset			Offset-1m			Offset-2m			Offset-4m		
	↑PSNR	↑SSIM	↓LPIPS	↑PSNR	↑SSIM	↓LPIPS	↑PSNR	↑SSIM	↓LPIPS	↑PSNR	↑SSIM	↓LPIPS
- *NeRF-based*												
Instant-NGP [38]	29.76	0.894	0.253	23.44	0.814	0.346	22.37	0.790	0.386	20.95	0.768	0.443
UC-NeRF [15]	35.95	0.936	0.311	30.07	0.896	0.355	25.17	0.863	0.367	22.89	0.797	0.420
MARS [57]	30.21	0.873	0.146	27.40	0.851	0.169	24.95	0.847	0.194	23.29	0.818	0.235
NeRFacto [48]	27.39	0.888	0.252	23.49	0.824	0.314	21.64	0.786	0.379	20.82	0.769	0.412
EmerNeRF [60]	31.76	0.907	0.126	28.66	0.878	0.150	26.05	0.852	0.182	24.80	0.837	0.203
- *Gaussian-based*												
3DGS [26]	30.87	0.916	0.274	23.26	0.873	0.334	22.01	0.829	0.396	19.17	0.768	0.460
PVG [13]	37.78	0.960	0.189	26.84	0.882	0.296	24.42	0.854	0.335	23.17	0.841	0.353
GaussianPro [16]	31.62	0.919	0.263	22.61	0.856	0.338	21.26	0.819	0.383	18.75	0.772	0.445
DC-Gaussian [53]	31.29	0.919	0.264	26.82	0.884	0.298	25.24	0.871	0.319	22.90	0.844	0.360

Table 2. Results on our proposed dataset with the different offsets using *left-front, front, right-front* cameras. ↑: higher is better, ↓: lower is better. The red, orange, and yellow colors respectively denote the best, the second best, and the third best results.

Method	*w/o* Offset			Offset-1m			Offset-2m			Offset-4m		
	↑PSNR	↑SSIM	↓LPIPS	↑PSNR	↑SSIM	↓LPIPS	↑PSNR	↑SSIM	↓LPIPS	↑PSNR	↑SSIM	↓LPIPS
- *NeRF-based*												
Instant-NGP [38]	29.24	0.888	0.262	23.52	0.847	0.344	22.39	0.815	0.382	21.05	0.783	0.428
UC-NeRF [15]	33.12	0.912	0.360	30.07	0.896	0.355	28.81	0.881	0.373	26.87	0.870	0.421
MARS [57]	31.37	0.887	0.151	29.28	0.874	0.157	28.54	0.869	0.165	26.49	0.847	0.193
NeRFacto [48]	29.39	0.890	0.246	22.91	0.850	0.307	22.26	0.809	0.348	21.03	0.779	0.393
EmerNeRF [60]	31.51	0.894	0.146	29.41	0.878	0.152	28.66	0.873	0.160	26.62	0.851	0.188
- *Gaussian-based*												
3DGS [26]	29.74	0.914	0.312	22.08	0.842	0.359	21.34	0.824	0.402	19.47	0.796	0.441
PVG [13]	33.33	0.933	0.256	26.62	0.878	0.318	25.50	0.870	0.332	23.35	0.849	0.360
GaussianPro [16]	28.13	0.889	0.321	21.90	0.839	0.379	20.85	0.822	0.404	19.36	0.795	0.443
DC-Gaussian [53]	30.23	0.912	0.271	26.74	0.883	0.315	25.53	0.872	0.329	23.53	0.860	0.357

analyses.

4.2. Benchmarking Methods

InstantNGP [38]: We employ the Adam optimizer and maintain similar parameter settings as the original Instant-NGP implementation: the learning rate is 1×10^{-4}, the number of feature dimensions per entry is $F = 8$, the number of levels is $L = 10$, and the hash tables is 2^4. We train the model with 30,000 steps.

Nerfacto [48]: We use the implementation in [48] without pose refinement to test our benchmark. We employ the Adam optimizer with 1×10^{-3} learning rate. We train the model with 30,000 steps.

MARS [57]: We inherit most of the parameter settings as the original MARS implementation. We employ the RAdam optimizer with 1×10^{-3} learning rate. Since our scenes are static without moving objects, we disable $\mathcal{L}_{sem}$, and the rest of the loss functions remain the same. We train the model with 50,000 steps.

UC-NeRF [15]: We inherit most parameter settings from the original UC-NeRF implementation. We employ the AdamW optimizer with 2.5×10^{-3} learning rate. The weight of sky loss is set to 2×10^{-3}. We train the model with 40,000 steps.

3DGS [26]: We use the implementation of NerfStudio to evaluate our dataset. We employ the AdamW optimizer with 1×10^{-3} learning rate. For stability, we "warm up" the computation in lower resolution. Specifically, we start the optimization using 4 times smaller image resolution and then upsample twice after 500 and 1000 iterations. We train the model with 30,000 steps.

PVG [13]: We employ the Adam optimizer and maintain a similar learning rate for most parameters as the original PVG implementation. At the same time, we adjust the learning rate of the amplitude $\mathbf{A}$, opacity decaying β and opacity $\mathbf{O}$ to 3×10^{-5}, 0.02 and 0.005 respectively. We train the model with 30,000 steps.

GaussianPro [16]: In alignment with the approach described in GaussianPro, our models are trained for 30,000 iterations across all scenes following GaussianPro's training schedule and hyper-parameters. The interval step of the progressive propagation strategy is set to 50 where propa-

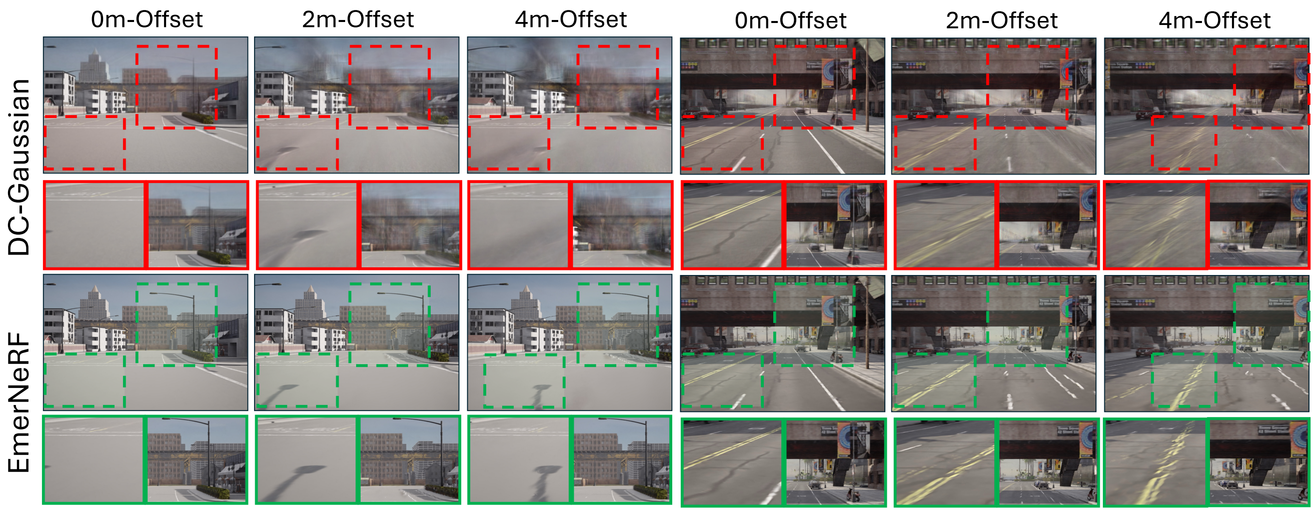

Figure 3. Visualization of the rendered images using DC-Gaussian and EmerNeRF with different offsets (*i.e.* 0m, 2m, and 4m) in two scenes. The discriminated areas are highlighted, and the areas with better results are marked as □, while worse results are marked as □.

gation is performed 3 times. The threshold σ of the absolute relative difference is set to 0.8. We set $\beta = 0.001$ and $\gamma = 0.001$ for the planar loss.

DC-Gaussian [53]: To align the performance described in DC-Gaussian, we set loss coefficient 0.001 for both photometric and sky losses. Sky loss is the same as UC-NeRF.

4.3. Used Metrics

We adopt the evaluation criteria employed by the methods mentioned above, which comprise Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS) as our evaluation metrics. Furthermore, we ensure transparency and clarity by describing the experimental framework employed to compare the methods.

4.4. Experimental Results

We evaluate the methods mentioned below on our dataset using novel-view-synthesis metrics. These methods are trained in two modes: (1) front-only mode (Tab. 1) utilizes images captured solely by the front camera; (2) multi-camera mode (Tab. 2) incorporates images from all three cameras. We conduct separate evaluations on trajectories with 0m, 1m, 2m, and 4m offsets. Our evaluation encompasses qualitative and quantitative experiments, ensuring a comprehensive analysis of the methods' performance. Furthermore, it is worth noting that EmerNeRF [60] demonstrates the highest performance in the cross-lane dataset, with an average PSNR of 26.50 dB. We attribute this superior performance to its inherent self-supervised scene decomposition and positional embedding decomposition capability. We place detailed experimental results in Tab. 1 and Tab. 2. For more detailed results, please refer to the supplementary material and our webpage.

5. Findings

NeRFs perform better than 3D-GS averagely

Lately, there has been a shift in research interest within the community, transitioning from NeRF towards 3D-GS. 3D-GS has achieved state-of-the-art results on datasets such as KITTI and Waymo. However, it is worth noting that these datasets primarily focus on evaluating interpolation images without considering significant offsets (*i.e.* cross-lane NVS). Consequently, a crucial aspect of our dataset is to assess and compare the novel-view synthesis performance between NeRF and 3D-GS methods under challenging conditions involving substantial offsets.

As shown in Tab. 2, in the scenario with no offset, compared with EmerNeRF [60], PVG [13] achieves better performance by 0.2dB in PSNR metrics with 3-cameras setting and 1.83dB in PSNR metrics with 1-camera setting. In the scenario with 4m offset, EmerNeRF [60] and UC-NeRF [15] show a much more powerful ability to synthesize images in cross-lane novel view. compared with the SOTA Gaussian method (*i.e.* DC-Gaussian [53]), they achieve leading performance by 1.90dB and 3.43dB in PSNR separately. The visualization results are shown in Fig. 3.

Self-decomposition handles background better

In this comparison, we examine the self-decomposition method (PVG [13]) alongside traditional approaches (3D-GS [26] and GaussianPro [16]), the quantitative results are shown in Tab. 3. The traditional method initializes 3D-GS

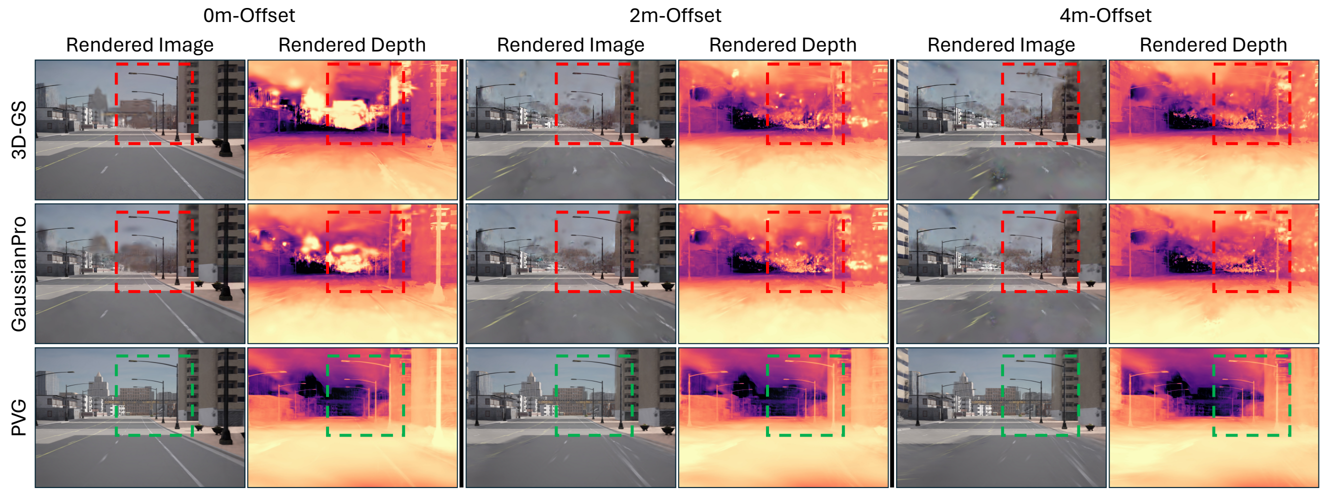

Figure 4. Visualization of the rendered images and depth maps using 3D-GS [26], GaussianPro [16], and PVG [13] with different offsets (*i.e.* 0m, 2m, and 4m) in two scenes. The discriminated areas are highlighted, and the areas with better results are marked as '□'.

primitives, denoted as $\mathcal{G} = \{(\sigma_i, \boldsymbol{\mu}_i, \Sigma_i, c_i)\}_{i=1}^{G}$, using LiDAR point clouds and extends them to areas lacking geometric features through cloning and splitting operations. However, these techniques lack a geometry prior and heavily rely on photometric loss, resulting in an issue of overfitting. This problem becomes evident in our benchmark, particularly in the context of cross-lane novel view synthesis, as illustrated in Fig. 4.

When not initialized by point clouds, the Gaussian points representing the background tend to overfit the training data to minimize rendering loss, often at the expense of positional accuracy. While they may appear satisfactory within the training trajectory, their performance degrades significantly when rendering images with offsets from the training trajectory. In Fig. 4, it can be observed that the street lamps are accurately modeled in the rendered images without any offset. However, when the offset is increased to 2m and 4m, these lamps become obscured by the background Gaussian points, resulting in a substantial decline in performance. Specifically, there is a decrease in performance of -7.99dB and -7.69dB for 3D-GS [26] and GaussianPro [16], respectively, as we move from 0 meters to 4 meters offset.

In contrast, the self-decomposition method, such as PVG [13], incorporates frequency information into 3D-GS primitives to create a unified representation of dynamic objects and the background. These primitives are denoted as $\mathcal{G}_t^k = \{\tilde{\mu}_t^k, \Sigma^k, \tilde{\alpha}_t^k, S^k\}_{k=1}^{G}$. Additionally, PVG employs a high-resolution environment cube map, denoted as $f_{sky}(d) = c_{sky}$, to effectively handle high-frequency details in the sky, where d represents the ray direction. These techniques enable the network to model the sky and avoid local minima accurately. As depicted in Fig. 4, PVG produces depth maps that better represent the background, leading to more realistic images than other Gaussian methods. It achieves an impressive improvement of 4.72dB in PSNR compared to the baseline 3D-GS [26], resulting in higher fidelity rendered images.

Table 3. Results on three 3D-GS methods with the different offsets with multi-cameras setting.

Offset	3D-GS			GaussianPro			PVG		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
0m	27.23	0.892	0.278	26.97	0.890	0.283	34.08	0.948	0.219
1m	22.33	0.846	0.331	21.89	0.840	0.335	28.11	0.907	0.270
2m	21.15	0.825	0.355	20.76	0.819	0.362	26.63	0.894	0.289
4m	19.24	0.797	0.393	19.01	0.792	0.399	23.96	0.873	0.317

Multi-camera benefits cross-lane NVS

Using EmerNeRF [60] as an example, we compare the performance differences between multiple-cameras and front-camera settings shown in Fig. 5. NeRF with only front camera training performs better than the network using three cameras by 0.25db in PSNR metrics when rendering images without offsets. However, when it comes to an offset, we observe a trend that NeRF using multiple-camera training performs better than using only front-camera training, up to 1.82db PSNR improvement in the scenario of 4m offset. These results indicate that using more cameras can significantly boost the performance of cross-lane novel view synthesis ability.

Geometric quality is key to cross-lane NVS

NeRF and 3D-GS methods have excellently performed on established autonomous driving datasets such as KITTI and Waymo. In these datasets, the primary function of

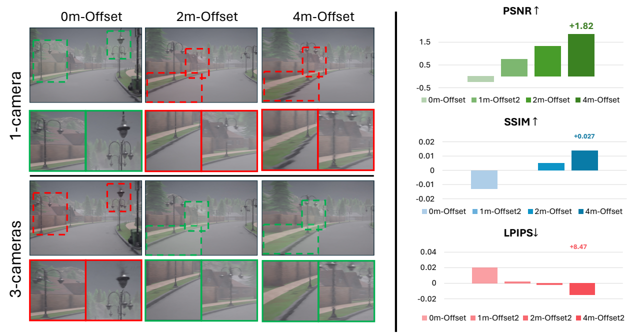

Figure 5. Use EmerNeRF [60] as example. In the left column: we visualize the novel-view-synthesis results with different offsets and camera numbers, the discriminate areas are highlighted. In the right column, we show the performance improvement between 3-cameras and 1-camera settings using PSNR, SSIM, and LPIPS.

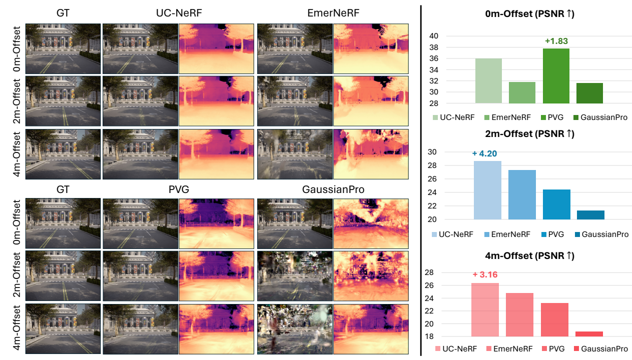

Figure 6. In the left column: we visualize the novel-view-synthesis results of four NeRF and Gaussian methods with different offsets. In the right column, we demonstrate the quantitative comparison between these four methods with different offsets using PSNR metrics.

the NVS methods is often to replay or interpolate existing views within the training trajectory rather than generating images in truly novel views. However, our cross-lane novel view synthesis benchmark offers a distinct evaluation scenario that differs significantly from previous efforts validated on Waymo or KITTI datasets. This benchmark introduces novel challenges and evaluation criteria specifically designed to assess the performance of NVS methods when synthesizing images from viewpoints across different lanes. As a result, the evaluation of NeRF and 3D-GS methods in this benchmark provides unique insights beyond their performance on traditional datasets.

As shown in the right column of Fig. 6, all the selected methods perform well in the scenarios without any offset, which is identical to the previous validation efforts, and PVG even achieves leading performance up to 37.78dB in PSNR. When increasing the offset to 4 meters, its performance drops significantly while UC-NeRF [15] and EmerNeRF [60] surpass PVG by 3.16dB and 1.63dB in PSNR metric.

To this end, we visualize the rendered RGB images and depth maps with different methods, as shown in the left column of Fig. 6. UC-NeRF and EmerNeRF perform reasonable depth maps, properly distributing background buildings and sky. PVG successfully detaches the sky but performs poorly on the background buildings. GaussianPro overfits the training images and fails to estimate depth. These seriously affect the render performance in novel views with large offsets, where UC-NeRF, EmerNeRF, and PVG can render the rough structure of the scene in novel views, but GaussianPro fails. These phenomena show that the NeRF / Gaussian methods need to achieve precise geometry reconstruction to obtain better performance in cross-lane NVS.

5.1. Discussion

The results show that current approaches exhibit a substantial gap, indicating their limited capability to meet the rigorous requirements of cross-lane or closed-loop simulation. Future research endeavors can leverage our proposed dataset and benchmark to gauge how novel methods can advance toward achieving closed-loop simulation.

Limitation. Currently, our dataset has certain limitations as it was generated using the Unreal-based Carla simulator. We see our work represents an initial stride towards the accurate evaluation of novel driving view synthesis. As a future endeavor, we plan to curate a real-world dataset, similar to [40], encompassing cross-lane ground truth data. This expansion will enhance the authenticity and applicability of our evaluation, advancing the field of novel driving view synthesis.

6. Conclusion

In conclusion, this paper addresses the challenge that existing evaluation methods for NVS fall short of meeting the requirements of closed-loop simulations, which demand the capability to render views beyond the original trajectory. We introduce a unique driving view synthesis dataset and benchmark for autonomous driving simulations. This dataset includes testing images captured by deviating from the training trajectory, enabling the realistic evaluation of NVS approaches. Our dataset establishes a much-needed benchmark for advancing NVS techniques in closed-loop autonomous driving simulation.

Acknowledgements

The authors would like to thank Yuanyuan Gao and Jingfeng Li for their assistance in preparing the experiments and the reviewers for their insightful feedback.

References

[1] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pages 696–712. Springer, 2020. 2

[2] Alexander Amini, Igor Gilitschenski, Jacob Phillips, Julia Moseyko, Rohan Banerjee, Sertac Karaman, and Daniela Rus. Learning robust control policies for end-to-end autonomous driving from data-driven simulation. *IEEE Robot. Automat. Lett. (RA-L)*, 2020. 2

[3] Alexander Amini, Tsun-Hsuan Wang, Igor Gilitschenski, Wilko Schwarting, Zhijian Liu, Song Han, Sertac Karaman, and Daniela Rus. Vista 2.0: An open, data-driven simulator for multimodal sensing and policy learning for autonomous vehicles. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2022. 2

[4] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, pages 5855–5864, 2021. 2

[5] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 5470–5479, 2022. 2

[6] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, pages 19697–19705, 2023. 2

[7] Christian Berger and Bernhard Rumpe. Engineering autonomous driving software. *arXiv preprint arXiv:1409.6579*, 2014. 1

[8] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020. 3

[9] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 11621–11631, 2020. 3

[10] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 8748–8757, 2019. 3

[11] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 19457–19467, 2024. 3

[12] Gaurav Chaurasia, Sylvain Duchene, Olga Sorkine-Hornung, and George Drettakis. Depth synthesis and local warps for plausible image-based navigation. *ACM Trans. on Graphics (TOG)*, 32(3):1–12, 2013. 2

[13] 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. 3, 5, 6, 7

[14] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2024. 3

[15] 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 *Proc. of the Int. Conf. on Learning Representations (ICLR)*, 2023. 2, 4, 5, 6, 8

[16] Kai Cheng, Xiaoxiao Long, Kaizhi Yang, Yao Yao, Wei Yin, Yuexin Ma, Wenping Wang, and Xuejin Chen. Gaussianpro: 3d gaussian splatting with progressive propagation. In *Proc. of the Int. Conf. on Machine Learning (ICML)*, 2024. 3, 4, 5, 6, 7

[17] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 3213–3223, 2016. 3

[18] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. 2016. 2

[19] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Proc. of the Conf. on Robot Learning (CoRL)*, 2017. 2, 4

[20] Jin Fang, Dingfu Zhou, Jingjing Zhao, Chenming Wu, Chulin Tang, Cheng-Zhong Xu, and Liangjun Zhang. Lidarcs dataset: Lidar point cloud dataset with cross-sensors for 3d object detection. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 14822–14829, 2024. 3

[21] Shuo Feng, Haowei Sun, Xintao Yan, Haojie Zhu, Zhengxia Zou, Shengyin Shen, and Henry X Liu. Dense reinforcement learning for safety validation of autonomous vehicles. *Nature*, 615(7953):620–627, 2023. 1

[22] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 3354–3361, 2012. 3

[23] Lei He, Leheng Li, Wenchao Sun, Zeyu Han, Yichen Liu, Sifa Zheng, Jianqiang Wang, and Keqiang Li. Neural radiance field in autonomous driving: A survey. *arXiv preprint arXiv:2404.13816*, 2024. 1

[24] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 17853–17862, 2023. 1

[25] Xinyu Huang, Xinjing Cheng, Qichuan Geng, Binbin Cao, Dingfu Zhou, Peng Wang, Yuanqing Lin, and Ruigang Yang.

The apolloscape dataset for autonomous driving. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops*, pages 954–960, 2018. 3

[26] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. on Graphics (TOG)*, 2023. 1, 2, 5, 6, 7

[27] Bernhard Kerbl, Andreas Meuleman, Georgios Kopanas, Michael Wimmer, Alexandre Lanvin, and George Drettakis. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Trans. on Graphics (TOG)*, 43(4):1–15, 2024. 3

[28] Hao Li, Yuanyuan Gao, Chenming Wu, Dingwen Zhang, Yalun Dai, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, and Junwei Han. Ggrt: Towards generalizable 3d gaussians without pose priors in real-time. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2024. 3

[29] Hao Li, Jingfeng Li, Dingwen Zhang, Chenming Wu, Jieqi Shi, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, and Junwei Han. Vdg: Vision-only dynamic gaussian for driving simulation. *arXiv preprint arXiv*, 2024. 3

[30] Wei Li, CW Pan, Rong Zhang, JP Ren, YX Ma, Jin Fang, FL Yan, QC Geng, XY Huang, HJ Gong, et al. Aads: Augmented autonomous driving simulation using data-driven algorithms. *Science robotics*, 2019. 2

[31] Yueyuan Li, Wei Yuan, Songan Zhang, Weihao Yan, Qiyuan Shen, Chunxiang Wang, and Ming Yang. Choose your simulator wisely: A review on open-source simulators for autonomous driving. *IEEE Trans. on Intelligent Vehicles*, 2024. 2

[32] Zhuopeng Li, Chenming Wu, Liangjun Zhang, and Jianke Zhu. Dgnr: Density-guided neural point rendering of large driving scenes. *IEEE Transactions on Automation Science and Engineering*, 2024. 2

[33] Zhuopeng Li, Yilin Zhang, Chenming Wu, Jianke Zhu, and Liangjun Zhang. Ho-gaussian: Hybrid optimization of 3d gaussian splatting for urban scenes. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2024. 3

[34] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Trans. on Pattern Analalysis and Machine Intelligence (TPAMI)*, 45(3):3292–3310, 2022. 3

[35] Ligang Liu, Xi Xia, Han Sun, Qi Shen, Juzhan Xu, Bin Chen, Hui Huang, and Kai Xu. Object-aware guidance for autonomous scene reconstruction. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, 2018. 1

[36] Mingyu Liu, Ekim Yurtsever, Jonathan Fossaert, Xingcheng Zhou, Walter Zimmer, Yuning Cui, Bare Luka Zagar, and Alois C Knoll. A survey on autonomous driving datasets: Statistics, annotation quality, and a future outlook. *IEEE Trans. on Intelligent Vehicles*, 2024. 3

[37] 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. *Communications of the ACM*, 65(1):99–106, 2021. 1

[38] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. on Graphics (TOG)*, 41 (4):1–15, 2022. 2, 5

[39] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, pages 4990–4999, 2017. 3

[40] Ziqian Ni Ni, Sicong Du, Zhenghua Hou, Chenming Wu, and Sheng Yang. Para-lane: Multi-lane dataset registering parallel scans for benchmarking novel view synthesis. In *Proc. of the Intl. Conf. on 3D Vision (3DV)*, 2025. 8

[41] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 2856–2865, 2021. 2

[42] Darius Rückert, Linus Franke, and Marc Stamminger. Adop: Approximate differentiable one-pixel point rendering. *ACM Trans. on Graphics (TOG)*, 41(4):1–14, 2022. 2

[43] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[44] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016. 1

[45] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics: Results of the 11th International Conference*, pages 621–635. Springer, 2018. 2

[46] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 2446–2454, 2020. 3

[47] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 8248–8258, 2022. 2

[48] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–12, 2023. 2, 5

[49] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2012. 2

[50] Apostolia Tsirikoglou, Joel Kronander, Magnus Wrenninge, and Jonas Unger. Procedural modeling and physically based rendering for synthetic data generation in automotive applications. *arXiv preprint arXiv:1710.06270*, 2017. 1

[51] Haithem Turki, Jason Y Zhang, Francesco Ferroni, and Deva Ramanan. Suds: Scalable urban dynamic scenes. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 12375–12385, 2023. 2

[52] Huijie Wang, Tianyu Li, Yang Li, Li Chen, Chonghao Sima, Zhenbo Liu, Bangjun Wang, Peijin Jia, Yuting Wang, Shengyin Jiang, et al. Openlane-v2: A topology reasoning benchmark for unified 3d hd mapping. In *Proc. of the Conference on Neural Information Processing Systems (NeurIPS)*, 2024. 3

[53] Linhan Wang, Kai Cheng, Shuo Lei, Shengkun Wang, Wei Yin, Chenyang Lei, Xiaoxiao Long, and Chang-Tien Lu. Dc-gaussian: Improving 3d gaussian splatting for reflective dash cam videos. In *Proc. of the Conference on Neural Information Processing Systems (NeurIPS)*, 2024. 3, 5, 6

[54] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493*, 2023. 3

[55] Chenming Wu, Jiadai Sun, Zhelun Shen, and Liangjun Zhang. Mapnerf: Incorporating map priors into neural radiance fields for driving view simulation. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 7082–7088, 2023. 1

[56] Ke Wu, Kaizhao Zhang, Zhiwei Zhang, Shanshuai Yuan, Muer Tie, Julong Wei, Zijun Xu, Jieru Zhao, Zhongxue Gan, and Wenchao Ding. Hgs-mapping: Online dense mapping using hybrid gaussian representation in urban scenes. *arXiv preprint arXiv:2403.20159*, 2024. 3

[57] Zirui Wu, Tianyu Liu, Liyi Luo, Zhide Zhong, Jianteng Chen, Hongmin Xiao, Chao Hou, Haozhe Lou, Yuantao Chen, Runyi Yang, et al. Mars: An instance-aware, modular and realistic simulator for autonomous driving. In *CAAI International Conference on Artificial Intelligence*, pages 3–15. Springer, 2023. 1, 2, 4, 5

[58] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Proc. of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 12077–12090, 2021. 4

[59] 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. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2024. 1, 3

[60] 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. In *Proc. of the Int. Conf. on Learning Representations (ICLR)*, 2024. 2, 5, 6, 7, 8

[61] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 1389–1399, 2023. 1, 2

[62] Wang Yifan, Felice Serena, Shihao Wu, Cengiz Öztireli, and Olga Sorkine-Hornung. Differentiable surface splatting for point-based geometry processing. *ACM Trans. on Graphics (TOG)*, 38(6):1–14, 2019. 2

[63] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 2636–2645, 2020. 3

[64] Wenzhao Zheng, Ruiqi Song, Xianda Guo, and Long Chen. Genad: Generative end-to-end autonomous driving. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2024. 1

[65] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024. 3