DrivingGaussian: Composite Gaussian Splatting for Surrounding Dynamic Autonomous Driving Scenes

Xiaoyu Zhou[1] Zhiwei Lin[1] Xiaojun Shan[1] Yongtao Wang[1 *]
Deqing Sun[2 †] Ming-Hsuan Yang[2,3 †]
[1]Wangxuan Institute of Computer Technology, Peking University
[2]Google Research [3]University of California, Merced

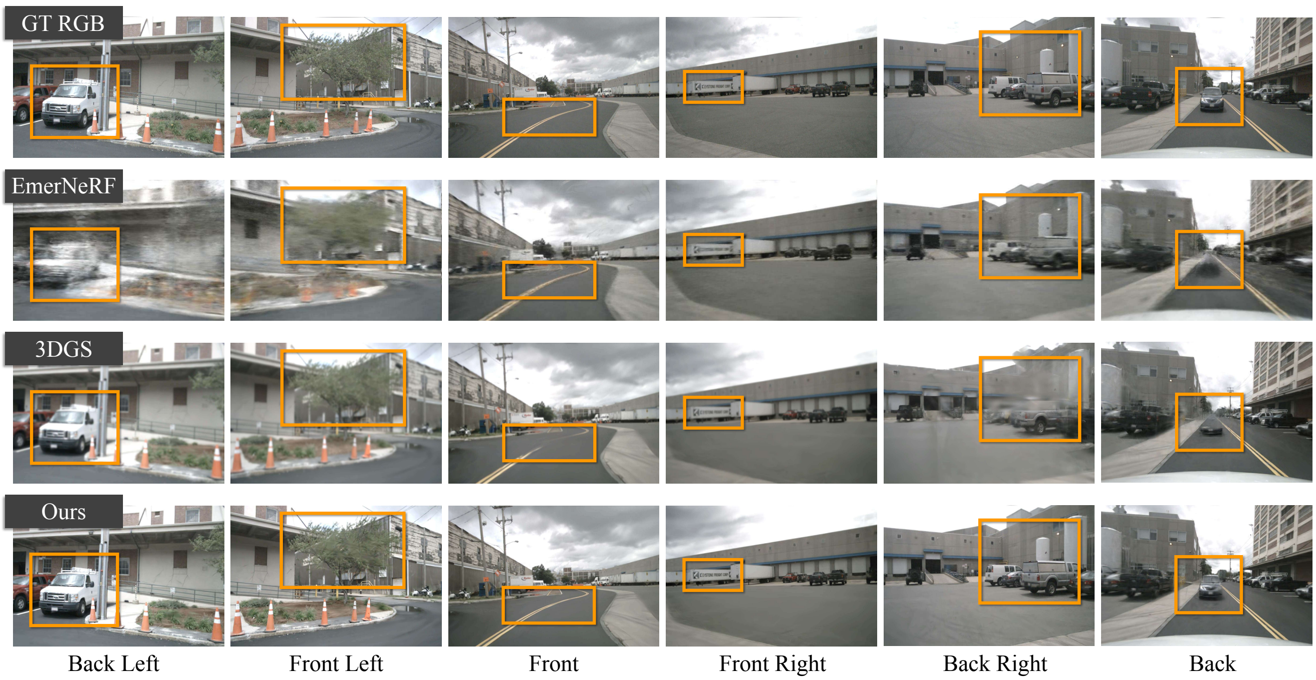

Figure 1. DrivingGaussian achieves photorealistic reconstruction and rendering performance for surrounding dynamic autonomous driving scenes. Naive approaches [17, 53] either produce unpleasant artifacts and blurring in the large-scale background or struggle with reconstructing dynamic objects and detailed scene geometry. DrivingGaussian first introduces Composite Gaussian Splatting to efficiently represent static backgrounds and multiple dynamic objects in complex surrounding driving scenes. DrivingGaussian enables the high-quality synthesis of surrounding views across multi-camera and facilitates long-term dynamic scene reconstruction.

Abstract

We present DrivingGaussian, an efficient and effective framework for surrounding dynamic autonomous driving scenes. For complex scenes with moving objects, we first sequentially and progressively model the static background of the entire scene with incremental static 3D Gaussians. We then leverage a composite dynamic Gaussian graph to handle multiple moving objects, individually reconstructing each object and restoring their accurate positions and occlusion relationships within the scene. We further use a LiDAR prior for Gaussian Splatting to reconstruct scenes with greater details and maintain panoramic consistency. DrivingGaussian outperforms existing methods in dynamic driving scene reconstruction and enables photorealistic surround-view synthesis with high-fidelity and multi-camera consistency. Our project page is at: https://github.com/VDIGPKU/DrivingGaussian.

*Corresponding author.
†Equal contribution.

1. Introduction

Representing and modeling large-scale dynamic scenes serves as the foundation for 3D scene understanding and contributes to a series of autonomous driving tasks, such as BEV perception [20, 21, 25], 3D detection [7, 9], and motion planning [6, 42]. View synthesis and controllable simulation for driving scenes also enable the generation of corner cases, and safety-critical situations aid in validating and enhancing the safety of autonomous driving systems.

Unfortunately, reconstructing such complex 3D scenes from sparse vehicle-mounted sensor data is challenging, especially when the ego vehicle moves at high speeds. Imagine a scene where a vehicle emerges at the edge of an unbounded scene captured by the left-front camera, swiftly moves to the center of the front camera's view, and in the subsequent frames, diminishes into a distant dot. For such driving scenes, both ego vehicles and dynamic objects are moving at relatively high speeds, posing significant challenges to the scene's construction. The static background and dynamic objects undergo rapid changes, depicted through limited views. Additionally, it becomes even more challenging in multi-camera settings due to their outward views, minimal overlaps, and variations in light from different directions. Complex geometry, diverse optical degradation, and spatiotemporal inconsistency also pose significant challenges to modeling such 360° large-scale dynamic driving scenes.

Neural radiance fields [30] (NeRF) has recently emerged as a promising neural reconstruction method for modeling object-level or room-level scenes. Some recent studies [41, 43, 46, 56] have extended NeRF to large-scale, unbounded static scenes, while some focus on modeling multiple dynamic objects within the scene [32, 40]. However, NeRF-based methods are computationally intensive and require densely overlapping views and consistent lighting. These limit their ability to construct driving scenes with outward multi-camera setups at high speeds. Furthermore, network capacity limitations make it challenging for these methods to model long-term, dynamic scenes with multiple objects, leading to visual artifacts and blurring.

In contrast to NeRF, the 3D Gaussian Splatting (3DGS) [17] represents scenes with more explicit 3D Gaussian representation and achieves impressive performance in novel view synthesis. However, the original 3DGS still encounters significant challenges in modeling large-scale dynamic driving scenes due to fixed Gaussians and constrained representation capacity. Some efforts [27, 47, 55] have extended 3DGS to dynamic scenes by constructing Gaussians at each timestamp. Unfortunately, they focus on individual dynamic objects and fail to handle complex driving scenes involving combined static-dynamic regions and multiple moving objects at high speeds.

In this paper, we introduce DrivingGaussian, a novel framework that represents surrounding dynamic autonomous driving scenes. Our key idea is to model the complex driving scene hierarchically using sequential data from multiple sensors. We adopt Composite Gaussian Splatting to decompose the whole scene into static background and dynamic objects, reconstructing each part separately. Specifically, we first use incremental static 3D Gaussians to construct backgrounds from surrounding multi-camera views sequentially. We then employ a composite dynamic Gaussian graph to individually reconstruct each moving object and dynamically integrate them into the entire scene based on the Gaussian graph. Based on these, global rendering via Gaussian Splatting captures occlusion relationships in the real world, encompassing static backgrounds and dynamic objects. Further, we incorporate the LiDAR prior to 3DGS, which is capable of recovering more precise geometry and maintaining better multi-view consistency.

Extensive experiments show that our method achieves state-of-the-art performance on public autonomous driving datasets. Even without LiDAR prior, our method still presents promising performance, demonstrating its versatility in reconstructing dynamic driving scenes. In addition, our framework enables controllable scene editing, such as corner case simulation, which facilitates the validation of the safety and robustness of autonomous driving systems.

The main contributions of this work are:

- To our knowledge, DrivingGaussian is the first representation and modeling framework for large-scale, dynamic driving scenes based on Composite Gaussian Splatting.
- Two novel modules are introduced, including Incremental Static 3D Gaussians and Composite Dynamic Gaussian Graph. The former reconstructs the static background incrementally, while the latter models multiple dynamic objects with a Gaussian graph. Assisted by LiDAR prior, the proposed method facilitates the recovery of complete geometry in surrounding dynamic driving scenes.
- Comprehensive experiments show that DrivingGaussian outperforms previous methods in challenging autonomous driving benchmarks and enables corner case simulation for various downstream tasks.

2. Related Work

NeRF for Bounded Scenes. The rapid progress in neural rendering has received significant attention. Neural Radiance Fields (NeRF), which utilizes multi-layer perceptrons (MLPs) and differentiable volume rendering, can reconstruct 3D scenes and synthesize novel views from a set of 2D images. However, original NeRF is limited to bounded scenes, requiring a consistent distance between the center object and cameras. It also struggles with scenes captured with slight overlaps and outward capture methods. Numerous advancements have expanded the capabilities of NeRF, leading to notable improvements in training speed [10, 12,

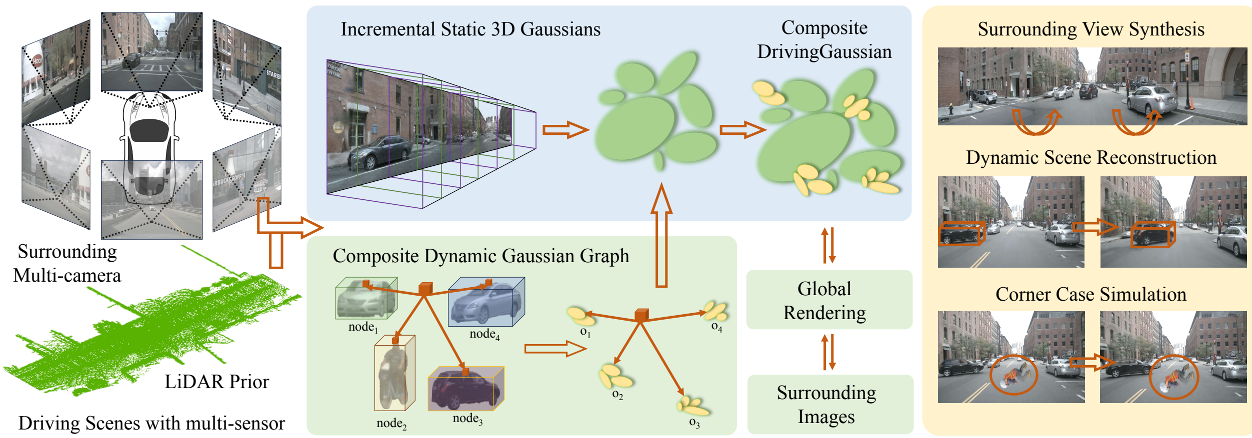

Figure 2. **Overall pipeline of our method. Left:** DrivingGaussian takes sequential data from multi-sensor, including multi-camera images and LiDAR. **Middle:** To represent the large-scale dynamic driving scenes, we propose Composite Gaussian Splatting, which consists of two components. The first part incrementally reconstructs the extensive static background, while the second constructs multiple dynamic objects with a Gaussian graph and dynamically integrates them into the scene. **Right:** DrivingGaussian demonstrates good performance across multiple tasks and application scenarios.

31], pose optimization [4, 23, 45], scene editing [19, 36], and dynamic scene representation [8, 11, 16, 33]. Nevertheless, applying NeRF to large-scale unbounded scenes, such as surrounding driving scenarios, remains a challenge.

NeRF for Unbounded Scenes. For large-scale unbounded scenes, [26, 28, 41, 43, 46, 56] have introduced refined versions of NeRF to model multi-scale urban-level static scenes. Inspired by the mipmapping approach to preventing aliasing, [2, 3] extend NeRF to unbounded scenes. To enable high-fidelity rendering, [51] combines the compact multi-resolution ground feature planes with NeRF for large urban scenes. [14] proposes a close-range-vs-distant-view disentanglement approach, which can model unbounded street views but ignores dynamic objects on the road. However, these methods model scenes under the assumption that the scene remains static and face challenges in effectively capturing dynamic elements.

Meanwhile, previous NeRF-based methods highly relied on accurate camera poses. Without precise poses, [24, 29] enable synthesis from dynamic monocular video. However, these methods are confined to forward monocular viewpoints and encounter challenges when dealing with inputs from surrounding multi-camera setups. For dynamic urban scenes, [32, 40] extend NeRF to dynamic scenes with multiple objects using a scene graph. [48, 54] propose instance-aware, modular, and realistic simulators for monocular dynamic scenes. [44, 50, 53] decompose the scene into static background and dynamic objects and constructs the scene with the help of LiDAR depth and 2D optical flow.

The quality of views synthesized by the aforementioned NeRF-based methods deteriorates in scenarios with multiple dynamic objects and variations and lighting variations, owing to their dependency on ray sampling. In addition, the utilization of LiDAR is confined to providing auxiliary depth supervision, and its potential benefits in reconstruction, such as providing geometric priors, are not explored.

To address these limitations, we utilize Composite Gaussian Splatting to model unbounded dynamic scenes, where the static background is incrementally reconstructed as the ego vehicle moves, and multiple dynamic objects are modeled and integrated into the entire scene through Gaussian graph. Meanwhile, LiDAR is utilized as the initialization for Gaussians, offering more precise geometric shape priors and comprehensive descriptions for 3D representation.

3D Gaussian Splatting. Recent 3D Gaussian Splatting (3DGS) [17] model a static scene with numerous 3D Gaussians, achieving optimal results in novel view synthesis and training speeds. Compared with previous explicit scene representations (e.g., mesh, voxels), the 3DGS can model complex shapes with fewer parameters. Unlike implicit neural rendering, the 3DGS allows fast rendering and differentiable computation with splat-based rasterization.

Dynamic 3D Gaussian Splatting. The original 3DGS is designed to represent static scenes, and some researchers have extended it to dynamic objects/scenes. Given a set of dynamic monocular images, [55] introduces a deformation network to model the motion of Gaussians. [47] connects adjacent Gaussians via a HexPlane, enabling real-time rendering. However, these two approaches are explicitly designed for monocular single-camera scenes focused on a center object. [27] parameterizes the entire scene using a set of dynamic Gaussians that evolve. However, it requires a camera array with dense multi-view as inputs.

In real-world autonomous driving scenes, the high-speed movement of data collection platforms leads to extensive and complex background variations, often captured by sparse views (e.g., 2-4 views). Moreover, fast-moving dynamic objects with intense spatial changes and occlusion further complicate the situation. Collectively, these factors pose significant challenges for existing methods.

3. Method

3.1. Composite Gaussian Splatting

3DGS performs well in purely static scenes but has significant limitations in mixed scenes involving large-scale static backgrounds and multiple dynamic objects. As illustrated in Figure 2, we aim to represent surrounding dynamic driving scenes with Composite Gaussian Splatting for large-scale static backgrounds and dynamic objects.

Incremental Static 3D Gaussians. The static backgrounds of driving scenes pose challenges due to their large scale, long duration, and variations in ego vehicle movement across multiple cameras. As the ego vehicle moves, the static background frequently undergoes temporal-spatial alterations. Prematurely incorporating distant street scenes from distant time steps can cause scale confusion due to the perspective principle, resulting in unpleasant artifacts and blurring. To solve this, we enhance 3DGS by introducing Incremental Static 3D Gaussians (Figure 3), leveraging the perspective changes introduced by the vehicle's movements and temporal relationships between adjacent frames.

Specifically, we first uniformly divide the static scene into N bins based on the depth range provided by LiDAR prior (Section 3.2). These bins are arranged in chronological order, denoted as $\{\mathrm{b}_i\}^N$, where each bin contains multi-camera images from one or more time steps. For the scene within the first bin, we initialize the Gaussians using LiDAR priors (similarly applicable to SfM points):

$$p_{b_0}(l|\mu,\Sigma) = e^{-\frac{1}{2}(l-\mu)^\top \Sigma^{-1}(l-\mu)} \tag{1}$$

where $l \in \mathbb{R}^3$ is the position of the LiDAR prior; μ is the mean of the LiDAR points; $\Sigma \in \mathbb{R}^{3\times 3}$ is an anisotropic covariance matrix; and $^\top$ is the transpose operator. We utilize the surrounding views within this bin segment as supervision to update the parameters of the Gaussian model, including position $P(x,y,z)$, covariance matrix Σ, coefficients of spherical harmonics for view-dependent color $C(r,g,b)$, along with an opacity α.

For the subsequent bins, we use the Gaussians from the previous bin as the position priors and align the adjacent bins based on their overlapping regions. The 3D center for each bin can be defined as:

$$\hat{P}_{b+1}(G_s) = P_b(G_s) \bigcup (x_{b+1}, y_{b+1}, z_{b+1}) \tag{2}$$

where $\hat{P}$ is the collection of 3D center for Gaussians G_s of all currently visible regions, $(x_{b+1}, y_{b+1}, z_{b+1})$ is the Gaussians coordinate within the $b+1$ region. Iteratively, we incorporate scenes from the subsequent bins into the previously constructed Gaussians with multiple surrounding frames as supervision. The incremental static Gaussian model G_s can be defined as:

$$\hat{C}(G_s) = \sum_{b=1}^{N} \Gamma_b \, \alpha_b \, C_b, \quad \Gamma_b = \prod_{i=1}^{b-1}(1-\alpha_b) \tag{3}$$

where C denotes the color corresponding to each single Gaussian at a certain view, α is the opacity, and Γ is the accumulated transmittance of the scene according to the α at all the bins. During this process, the overlapping regions between surrounding multi-camera images are used to form the Gaussian models' implicit alignment jointly.

Note that during the incremental construction of static Gaussian models, there might be differences in sampling the same scene between the front and rear cameras [14]. To address this, we employ a weighted averaging to reconstruct the scene's colors as accurately as possible during the 3D Gaussian projection:

$$\tilde{C} = \varsigma(G_s) \sum \omega(\hat{C}(G_s)|R,T) \tag{4}$$

where $\tilde{C}$ is the optimized pixel color, ς denotes the differential splatting, ω is a learnable weight coefficient, $[R,T]$ is the view-matirx for aligning multi-camera views.

Composite Dynamic Gaussian Graph. The autonomous driving environment is highly complex, involving multiple dynamic objects and temporal changes. As shown in Figure 3, objects are often observed from limited views (e.g., 2-4 views) due to the movements of the ego vehicle and dynamic objects. The high speed also leads to significant spatial changes in dynamic objects, making it challenging to represent them using fixed Gaussians.

To address the challenges, we introduce the Composite Dynamic Gaussian Graph, enabling the construction of multiple dynamic objects in large-scale, long-term driving scenes. We first decompose dynamic foregrounds from static backgrounds to build the dynamic Gaussian graph using bounding boxes provided by the datasets or predicted results from the 3D object detector and tracker. Dynamic objects are identified by their object ID and corresponding timestamps of appearance. Additionally, SAM-based models [18, 35] are employed for precise pixel-wise extraction of dynamic objects based on the range of bounding boxes.

We then build the dynamic Gaussian graph as

$$H = < O, G_d, M, P, A, T >, \tag{5}$$

where each node stores an instance object $o \in O$, $g_i \in G_d$ denotes the corresponding dynamic Gaussians, and $m_o \in M$ is the transform matrix for each object. $p_o(x_t, y_t, z_t) \in P$ is the center coordinate of the bounding box, and $a_o = (\theta_t, \phi_t) \in A$ is the orientation of the bounding box at time

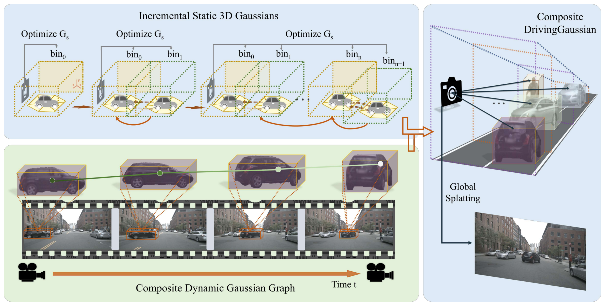

Figure 3. **Composite Gaussian Splatting with Incremental Static 3D Gaussians and Dynamic Gaussian Graph.** We adopt Composite Gaussian Splatting to decompose the whole scene into static background and dynamic objects, reconstructing each part separately and integrating them for global rendering.

step $t \in T$. For each dynamic object, we instantiate a separate ensemble of dynamic Gaussians and collectively optimize all parameters associated with these object Gaussians. Using the transformation matrix m_o, we transform the coordinate system of the target object o to the world coordinate where the static background resides:

$$m_o^{-1} = R_o^{-1} S_o^{-1} \tag{6}$$

where R_o^{-1} and S_o^{-1} are the rotation and translation matrices corresponding to each object.

After optimizing all nodes in the dynamic Gaussian graph, we combine dynamic objects and static backgrounds using a Composite Gaussian Graph. Each node's Gaussian distribution is concatenated into the static Gaussian field based on the bounding box position and orientation in chronological order. In cases of occlusion between multiple dynamic objects, we adjust the opacity based on the distance from the camera center: closer objects have higher opacity, following the principles of light propagation:

$$\alpha_{o,t} = \sum \frac{(p_t - b_o)^2 \cdot \cot a_o}{\|(b_o|R_o, S_o) - \rho\|^2} \alpha_{p_0} \tag{7}$$

where $\alpha_{o,t}$ is the adjusted opacity of dynamic Gaussians for object o at time step t, $p_t = (x_t, y_t, z_t)$ is the center of Gaussians for the object. $[R_o, S_o]$ denotes the object-to-world transform matrix, ρ denotes the center of camera view, and α_{p_0} is the opacity of Gaussians.

Finally, the composite Gaussian graph, including both static background and multiple dynamic objects, can be formulated as:

$$G_{comp} = \sum H < O, G_d, M, P, A, T > + G_s \tag{8}$$

where G_s is obtained in Section 3.1 through Incremental Static 3D Gaussians, and H denotes the optimized dynamic Gaussian graph of the entire scene.

3.2. LiDAR Prior with surrounding views

The primitive 3DGS attempts to initialize Gaussians via structure-from-motion (SfM). However, unbounded urban scenes for autonomous driving contain many multiscale backgrounds and foregrounds. Nonetheless, they are only glimpsed through exceedingly sparse views, resulting in erroneous and incomplete recovery of geometric structures.

To provide better initialization for Gaussians, we introduce the LiDAR prior to 3D Gaussian to obtain better geometries and maintain multi-camera consistency in surrounding view registration. At each timestep $t \in T$, given a set of multi-camera images $\{I_t^i | i = 1 \dots N\}$ collected from the moving platform and multi-frame LiDAR sweeps L_t. We aim to minimize multi-camera registration errors using LiDAR-image multi-modal data and obtain accurate point positions and geometric priors.

We first merge multiple frames of LiDAR sweeps to obtain the complete point cloud of the scene, denoted as L. We follow Colmap [39] and extract image features $X = x_p^q$ from each image individually. Next, we project the LiDAR points onto surrounding images. For each LiDAR point l, we transform its coordinates to the camera coordinate system and match it with the 2D-pixel of the camera image plane through projection:

$$x_p^q = K[R_t^i \cdot l_s + T_t^i] \tag{9}$$

where x_p^q is the 2D pixel of the image, I_t^i, R_t^i and T_t^i are orthogonal rotation matrices and translation vectors, respectively. $K \in \mathbb{R}^{3\times 3}$ is the known camera intrinsic. Notably, points from LiDAR might be projected onto multiple pixels across multiple images. Therefore, we select the point with the shortest Euclidean distance to the image plane and retain it as the projected point [1, 13], assigning color.

Similar to former works [15, 38] in 3D reconstruction, we extend the dense bundle adjustment (DBA) to multi-camera setup and obtain the updated LiDAR points. Experiment results prove that initializing with LiDAR prior to aligning with surrounding multi-camera images aids in providing the 3D Gaussians with more precise geometry priors.

3.3. Global Rendering via Gaussian Splatting

We adopt the differentiable 3D Gaussian splatting renderer ς from [17] and project the global composite 3D Gaussian into 2D, where the covariance matrix $\widetilde{\Sigma}$ is given by:

$$\widetilde{\Sigma} = J E \, \Sigma \, E^\top J^\top \tag{10}$$

where J is the Jacobian matrix of the perspective projection, and E denotes the world-to-camera matrix.

The composite Gaussian field projects the global 3D Gaussian onto multiple 2D planes and is supervised using surrounding views at each time step. In the global rendering process, Gaussians from the next time step are initially invisible to the current and subsequently incorporated with the supervision of corresponding global images.

The loss function of our method consists of three parts. Following [17, 49], we first introduce the Tile Structural

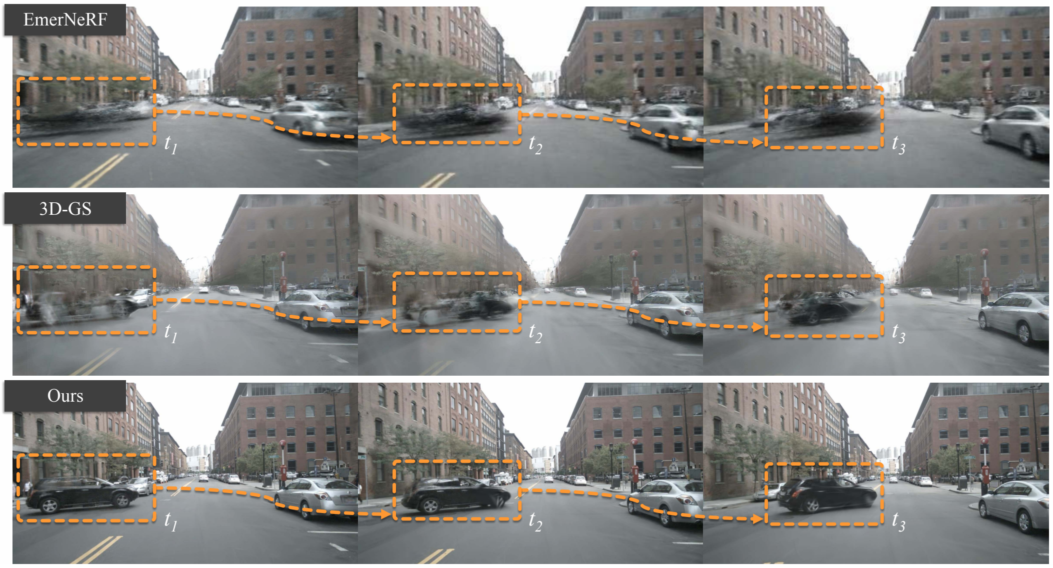

Figure 4. **Qualitative comparison on dynamic reconstruction.** We demonstrate the qualitative comparison results with our main competitors EmerNeRF [53] and 3DGS [17] on dynamic reconstruction for 4D driving scenes of nuScenes. DrivingGaussian enables the high-quality reconstruction of dynamic objects at high speed while maintaining temporal consistency.

Similarity (TSSIM) to Gaussian Splatting, which measures the similarity between the rendered tile and the corresponding ground truth.

$$L_{TSSIM}(\delta) = 1 - \frac{1}{Z}\sum_{z=1}^{Z} SSIM(\Psi(\hat{C}), \Psi(C)) \quad (11)$$

where we split the screen into M tiles, δ is the training parameters of the Gaussians, $\Psi(\hat{C})$ denotes the rendered tile from Composite Gaussian Splatting, and $\Psi(C)$ denotes the paired ground-truth tile.

We also introduce robust loss for reducing outliers [37] in 3D Gaussians, which can be defined as:

$$L_{Robust}(\delta) = \kappa(\|\hat{I} - I\|_2) \quad (12)$$

where $\kappa \in (0, 1]$ is set to a free variable that controls the robustness of the loss, I and $\hat{I}$ denote the ground truth and synthesis image, respectively.

The LiDAR loss is further employed by supervising the expected Gaussians' position from the LiDAR, obtaining better geometric structure and edge shapes:

$$L_{LiDAR}(\delta) = \frac{1}{s}\sum \|P(G_{comp}) - L_s\|^2 \quad (13)$$

where $P(G_{comp})$ is the position of 3D Gaussians, and L_s is the LiDAR point prior. We optimize the Composite Gaussians by minimizing the sum of three losses.

4. Experiments

4.1. Datasets

The nuScenes [5] dataset is a public large-scale dataset for autonomous driving, containing 1000 driving scenes collected with multiple sensors (6 cameras, 1 LiDAR, etc). It has annotations of 23 object classes with 3D bounding boxes ground truth. Our experiment uses the keyframes of driving scenes with images collected from 6 cameras and corresponding LiDAR sweeps (optional) as input. The KITTI-360 [22] dataset contains multiple sensors, corresponding to over 320k images and point clouds with instance annotations. Even though the dataset provides stereo camera images, we only use a single camera to validate that our method also performs well in monocular settings.

4.2. Implementation Details

Our implementation is primarily based on the 3DGS [17] and 4DGS [47] framework, with fine-tuned optimization parameters to fit the driving scenes. Instead of using SFM or randomly initialized points as input, we employ the LiDAR prior mentioned in Section 3.2 as the initialization. We increase the total training iterations to 60,000, set the threshold for densifying grad to 0.001, and reset the opacity interval to 900. The learning rate of Incremental Static 3D Gaussians remains the same as in the official setting, while the learning rate of the Composite Dynamic Gaussian Graph exponentially decays from 1.6e-3 to 1.6e-6. We evaluate our method using various metrics, including PSNR, SSIM, and LPIPS. All the experiments are carried out on 8 RTX8000 with 384 GB memory in total.

4.3. Results and Comparisons

Comparisons of surrounding view reconstruction on nuScenes. We conduct benchmarking against the state-of-the-art approaches, including NeRF-based methods [2, 3, 31, 32, 34, 44, 50, 53] and 3DGS-based methods [17, 47].

Table 1. **Overall perforamnce of DrivingGaussian with existing state-of-the-art approaches on the nuScenes dataset.** Ours-S denotes the DrivingGaussian with SfM points initialization, and Ours-L denotes training the Gaussian model with LiDAR prior.

Methods	Input	PSNR ↑	SSIM ↑	LPIPS ↓
Instant-NGP [31]	Images	16.78	0.519	0.570
NeRF+Time	Images	17.54	0.565	0.532
NSG [32]	Images	21.67	0.671	0.424
Mip-NeRF [2]	Images	18.08	0.572	0.551
Mip-NeRF360 [3]	Images	22.61	0.688	0.395
Urban-NeRF [34]	Images + LiDAR	20.75	0.627	0.480
S-NeRF [50]	Images + LiDAR	25.43	0.730	0.302
SUDS [44]	Images + LiDAR	21.26	0.603	0.466
EmerNeRF [53]	Images + LiDAR	26.75	0.760	0.311
3DGS [17]	Images + SfM Points	26.08	0.717	0.298
4DGS [47]	Images + SfM Points	19.79	0.622	0.473
Ours-S	Images + SfM Points	28.36	0.851	0.256
Ours-L	Images + LiDAR	28.74	0.865	0.237

Table 2. **Overall perforamcne of DrivingGaussian with existing state-of-the-art approaches on the KITTI-360 dataset.**

Methods	PSNR ↑	SSIM ↑
NeRF [30]	21.94	0.781
NSG [32]	22.89	0.836
Point-NeRF [52]	21.54	0.793
Mip-NeRF360 [3]	23.27	0.836
SUDS [44]	23.30	0.844
DNMP [26]	23.41	0.846
3DGS [17]	22.93	0.847
Ours-S	25.18	0.862
Ours-L	25.62	0.868

As shown in Table 1, our method outperforms Instant-NGP [31] by a large margin, which employs a hash-based NeRF. Compared with NSG [32], Mip-NeRF [2] and Mip-NeRF360 [3], our method also significantly surpasses them across all evaluated metrics.

Urban-NeRF [34] first introduces LiDAR to NeRF to reconstruct urban scenes. However, it primarily only leverages LiDAR to provide depth supervision. Instead, we leverage LiDAR as a more effective geometric prior and incorporate it into 3DGS. Our proposed method achieves superior results compared to S-NeRF [50] and SUDS [44], both of which decompose the scene into static background and dynamic objects and construct the scene with the help of LiDAR. Compared to our primary competitor, EmerNeRF [53], which applies spatial-temporal representation for dynamic driving scenes using flow fields. Our method outperforms it across all metrics, eliminating the necessity for estimating scene flow. For Gaussian-based approaches, our method boosts the performance of our baseline method 3DGS [17] and 4DGS [47] on driving scenes across all evaluated metrics and achieves optimal results.

We also compare qualitatively with our main competitors EmerNeRF [53] and 3DGS [17] on challenging nuScenes driving scenes. For surrounding view synthesis with multi-camera, as shown in Figure 1, our method enables the generation of photo-realistic rendering images and ensures view consistency across multi-camera. Meanwhile, EmerNeRF [53] and 3DGS [17] struggle in challenging regions, displaying undesirable visual artifacts such as ghosting, dynamic object disappearance, loss of plant texture details, lane markings, and distant scene blurring.

We further demonstrate the reconstruction results for dynamic temporal scenes. Our method accurately models dynamic objects within large-scale scenes, mitigating issues such as loss, ghosting, or blurring of these dynamic elements. We also maintain consistency in constructing dynamic objects over time, even when they move at a relatively fast speed. In comparison, [17, 53] both fail in modeling fast-moving dynamic objects, as shown in Figure 4.

Comparisons of mono-view reconstruction on KITTI-360. To further validate the effectiveness of our method on monocular driving scene setting, we conduct experiments with the KITTI-360 dataset and compare it with existing SOTA methods, including NeRF-based methods [3, 30], point-based method Point-NeRF [52], graph-based method NSG [32], flow-based method SUDS [44], mesh-based method DNMP [26], and 3DGS [17]. As shown in Table 2, our method demonstrates the optimal performance in monocular driving scenes, surpassing existing methods by a large margin. More quantitative and qualitative results are available in the supplementary material.

4.4. Ablation Study

Initialization prior for Gaussians. Comparative experiments are conducted to analyze the effect of different priors and initialization methods on the 3DGS. The original 3DGS provides two initialization modes: randomly generated points and SfM points computed by COLMAP [39]. We additionally offer two other initialization approaches: point clouds exported from a pre-trained NeRF model and points generated with LiDAR prior.

Meanwhile, to analyze the effect of point cloud quantity, we down-sample the LiDAR to 600K and apply adaptive filtering (1M) to control the number of LiDAR points. Differ-

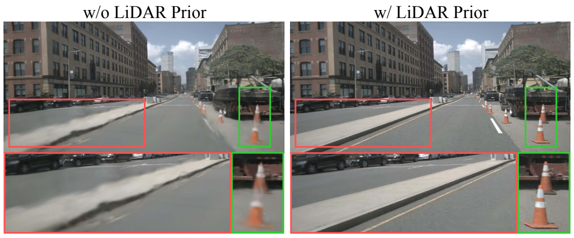

Figure 5. **Qualitative comparison with different initialization methods for 3D Gaussians.** The LiDAR prior for 3D Gaussians aids in obtaining better geometries and precise details.

Table 3. **Effect of different initialization methods on the Gaussian model.** LiDAR-600K † denotes for downsampling the original LiDAR points to a corresponding point cloud magnitude. LiDAR-1M ‡ denotes denoising and removing outliers in LiDAR points, which is used in our method.

Methods	PSNR ↑	SSIM ↑	LPIPS ↓
Random-600K	22.18	0.653	0.424
Random-1M	22.23	0.653	0.421
SfM-600K	28.36	0.851	0.256
NeRF-1M	28.51	0.858	0.251
LiDAR-600K †	28.49	0.854	0.245
LiDAR-1M ‡	28.74	0.865	0.237
LiDAR-2M	28.78	0.867	0.237

ent maximum thresholds (600K and 1M) are set for random initializations. Here, SfM-600K denotes the points computed by COLMAP, NeRF-1M denotes the points generated by the pre-trained NeRF model, and LiDAR-2M refers to the original quantity of LiDAR points.

As shown in Table 3, randomly generated points lead to the worst results as they lack any geometric prior. Initializing with SfM points also cannot adequately recover the scene's precise geometries due to the sparse points and intolerable structural errors. Leveraging point clouds generated from a pre-trained NeRF model provides a relatively accurate geometric prior, but there are still noticeable outliers. For the model initialized with LiDAR prior, although downsampling results in loss of geometric information in some local regions, it still retains relatively accurate structural priors, thus surpassing SfM (Figure 5). We can also observe that experiment results do not linearly change with increasing LiDAR point quantities. We deduce this is because overly dense points store redundant features that interfere with the optimization of the Gaussian model.

Effectiveness of Each Module. We analyze how each proposed module contributes to the final performance. As shown in Table 4, the Composite Dynamic Gaussian Graph module plays a crucial role in reconstructing dynamic driving scenes, while the Incremental Static 3D Gaussians module enables high-quality large-scale background reconstruction. These two novel modules significantly enhance the modeling quality of complex driving scenes. Regarding the proposed loss functions, results indicate that both L_{TSSIM} and L_{Robust} notably improve the rendering quality, enhancing texture details and removing artifacts. L_{LiDAR}, assisted by LiDAR prior, helps Gaussians achieve better geometric priors. Experimental results also demonstrate that DrivingGaussian performs well even without LiDAR prior, showcasing strong robustness for various initialization methods.

Table 4. **Effect of each module in our proposed method.** IS3G is short for the Incremental Static 3D Gaussians module, and CDGG is short for the Composite Dynamic Gaussian Graph module.

Model	PSNR ↑	SSIM ↑	LPIPS ↓
w/o IS3G	27.72	0.771	0.295
w/o CDGG	26.97	0.752	0.306
w/o L_{TSSIM}	27.88	0.783	0.280
w/o L_{Robust}	28.05	0.814	0.271
w/o L_{LiDAR}	28.45	0.854	0.248
Ours-S	28.36	0.851	0.256
Ours-L	28.74	0.865	0.237

4.5. Corner Case Simulation

We demonstrate the effectiveness of our approach to simulating corner cases in real-world driving scenes. As shown in Figure 6, we can insert arbitrary dynamic objects into the reconstructed scenes. The simulated scene maintains temporal coherence and exhibits good inter-sensor consistency among multiple sensors. Our method enables controllable simulation and editing for autonomous driving scenes, facilitating safe self-driving systems research.

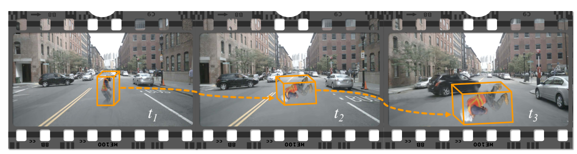

Figure 6. **Example of corner case simulation:** A man walking on the road suddenly falls, and a car approaches ahead.

5. Conclusion

We introduce DrivingGaussian, a novel framework for representing large-scale dynamic autonomous driving scenes based on the proposed Composite Gaussian Splatting. DrivingGaussian progressively models the static background with incremental static 3D Gaussians and captures multiple moving objects using a composite dynamic Gaussian graph. We further leverage LiDAR prior for accurate geometric structures and multi-view consistency. DrivingGaussian achieves state-of-the-art performance on two autonomous driving datasets, allowing high-quality surrounding view synthesis and dynamic scene reconstruction.

6. Acknowledgment

This work was supported by National Key R&D Program of China (Grant No.2022ZD0160305).

References

[1] Chunge Bai, Ruijie Fu, and Xiang Gao. Colmap-pcd: An open-source tool for fine image-to-point cloud registration. *arXiv preprint arXiv:2310.05504*, 2023. 5

[2] 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 *ICCV*, pages 5855–5864, 2021. 3, 6, 7

[3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, pages 5470–5479, 2022. 3, 6, 7

[4] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. In *CVPR*, pages 4160–4169, 2023. 3

[5] 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 *CVPR*, pages 11621–11631, 2020. 6

[6] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021. 2

[7] Qi Cai, Yingwei Pan, Ting Yao, Chong-Wah Ngo, and Tao Mei. Objectfusion: Multi-modal 3d object detection with object-centric fusion. In *ICCV*, pages 18067–18076, 2023. 2

[8] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *CVPR*, pages 130–141, 2023. 3

[9] Xuanyao Chen, Tianyuan Zhang, Yue Wang, Yilun Wang, and Hang Zhao. Futr3d: A unified sensor fusion framework for 3d detection. In *CVPR*, pages 172–181, 2023. 2

[10] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, pages 5501–5510, 2022. 2

[11] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *ICCV*, pages 5712–5721, 2021. 3

[12] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *ICCV*, pages 14346–14355, 2021. 2

[13] Vitor Guizilini, Igor Vasiljevic, Rares Ambrus, Greg Shakhnarovich, and Adrien Gaidon. Full surround monodepth from multiple cameras. *RAL*, pages 5397–5404, 2022. 5

[14] Jianfei Guo, Nianchen Deng, Xinyang Li, Yeqi Bai, Botian Shi, Chiyu Wang, Chenjing Ding, Dongliang Wang, and Yikang Li. Streetsurf: Extending multi-view implicit surface reconstruction to street views. *arXiv preprint arXiv:2306.04988*, 2023. 3, 4

[15] Quentin Herau, Nathan Piasco, Moussab Bennehar, Luis Roldão, Dzmitry Tsishkou, Cyrille Migniot, Pascal Vasseur, and Cédric Demonceaux. Moisst: Multi-modal optimization of implicit scene for spatiotemporal calibration. *arXiv preprint arXiv:2303.03056*, 2023. 5

[16] Xin Huang, Qi Zhang, Ying Feng, Hongdong Li, Xuan Wang, and Qing Wang. Hdr-nerf: High dynamic range neural radiance fields. In *CVPR*, pages 18398–18408, 2022. 3

[17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *TOG*, 42(4):1–14, 2023. 1, 2, 3, 5, 6, 7

[18] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 4

[19] Yuan Li, Zhi-Hao Lin, David Forsyth, Jia-Bin Huang, and Shenlong Wang. Climatenerf: Extreme weather synthesis in neural radiance field. In *ICCV*, pages 3227–3238, 2023. 3

[20] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *ECCV*, pages 1–18. Springer, 2022. 2

[21] Tingting Liang, Hongwei Xie, Kaicheng Yu, Zhongyu Xia, Zhiwei Lin, Yongtao Wang, Tao Tang, Bing Wang, and Zhi Tang. Bevfusion: A simple and robust lidar-camera fusion framework. *Advances in Neural Information Processing Systems*, 35:10421–10434, 2022. 2

[22] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3292–3310, 2022. 6

[23] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *ICCV*, pages 5741–5751, 2021. 3

[24] Yu-Lun Liu, Chen Gao, Andreas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang, Johannes Kopf, and Jia-Bin Huang. Robust dynamic radiance fields. In *CVPR*, pages 13–23, 2023. 3

[25] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han. Bevfusion: Multitask multi-sensor fusion with unified bird's-eye view representation. In *ICRA*, pages 2774–2781. IEEE, 2023. 2

[26] Fan Lu, Yan Xu, Guang Chen, Hongsheng Li, Kwan-Yee Lin, and Changjun Jiang. Urban radiance field representation with deformable neural mesh primitives. In *ICCV*, pages 465–476, 2023. 3, 7

[27] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*, 2023. 2, 3

[28] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, pages 7210–7219, 2021. 3

[29] Andreas Meuleman, Yu-Lun Liu, Chen Gao, Jia-Bin Huang, Changil Kim, Min H Kim, and Johannes Kopf. Progressively

optimized local radiance fields for robust view synthesis. In *CVPR*, pages 16539–16548, 2023. 3

[30] 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*, pages 99–106, 2021. 2, 7

[31] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *TOG*, 41(4):1–15, 2022. 3, 6, 7

[32] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *CVPR*, pages 2856–2865, 2021. 2, 3, 6, 7

[33] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *CVPR*, pages 10318–10327, 2021. 3

[34] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *CVPR*, pages 12932–12942, 2022. 6, 7

[35] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 4

[36] Viktor Rudnev, Mohamed Elgharib, William Smith, Lingjie Liu, Vladislav Golyanik, and Christian Theobalt. Nerf for outdoor scene relighting. In *ECCV*, pages 615–631. Springer, 2022. 3

[37] Sara Sabour, Suhani Vora, Daniel Duckworth, Ivan Krasin, David J Fleet, and Andrea Tagliasacchi. Robustnerf: Ignoring distractors with robust losses. In *CVPR*, pages 20626–20636, 2023. 6

[38] Aron Schmied, Tobias Fischer, Martin Danelljan, Marc Pollefeys, and Fisher Yu. R3d3: Dense 3d reconstruction of dynamic scenes from multiple cameras. In *ICCV*, pages 3216–3226, 2023. 5

[39] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113, 2016. 5, 7

[40] Yeji Song, Chaerin Kong, Seoyoung Lee, Nojun Kwak, and Joonseok Lee. Towards efficient neural scene graphs by learning consistency fields. *arXiv preprint arXiv:2210.04127*, 2022. 2, 3

[41] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *CVPR*, pages 8248–8258, 2022. 2, 3

[42] Siyu Teng, Xuemin Hu, Peng Deng, Bai Li, Yuchen Li, Yunfeng Ai, Dongsheng Yang, Lingxi Li, Zhe Xuanyuan, Fenghua Zhu, et al. Motion planning for autonomous driving: The state of the art and future perspectives. *IEEE Transactions on Intelligent Vehicles*, 2023. 2

[43] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In *CVPR*, pages 12922–12931, 2022. 2, 3

[44] Haithem Turki, Jason Y Zhang, Francesco Ferroni, and Deva Ramanan. Suds: Scalable urban dynamic scenes. In *CVPR*, pages 12375–12385, 2023. 3, 6, 7

[45] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 3

[46] Zian Wang, Tianchang Shen, Jun Gao, Shengyu Huang, Jacob Munkberg, Jon Hasselgren, Zan Gojcic, Wenzheng Chen, and Sanja Fidler. Neural fields meet explicit geometric representations for inverse rendering of urban scenes. In *CVPR*, pages 8370–8380, 2023. 2, 3

[47] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. *arXiv preprint arXiv:2310.08528*, 2023. 2, 3, 6, 7

[48] 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. *arXiv preprint arXiv:2307.15058*, 2023. 3

[49] Zeke Xie, Xindi Yang, Yujie Yang, Qi Sun, Yixiang Jiang, Haoran Wang, Yunfeng Cai, and Mingming Sun. S3im: Stochastic structural similarity and its unreasonable effectiveness for neural fields. In *ICCV*, pages 18024–18034, 2023. 5

[50] Ziyang Xie, Junge Zhang, Wenye Li, Feihu Zhang, and Li Zhang. S-nerf: Neural radiance fields for street views. *arXiv preprint arXiv:2303.00749*, 2023. 3, 6, 7

[51] Linning Xu, Yuanbo Xiangli, Sida Peng, Xingang Pan, Nanxuan Zhao, Christian Theobalt, Bo Dai, and Dahua Lin. Grid-guided neural radiance fields for large urban scenes. In *CVPR*, pages 8296–8306, 2023. 3

[52] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *CVPR*, pages 5438–5448, 2022. 7

[53] Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, et al. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. *arXiv preprint arXiv:2311.02077*, 2023. 1, 3, 6, 7

[54] Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *CVPR*, pages 1389–1399, 2023. 3

[55] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint arXiv:2309.13101*, 2023. 2, 3

[56] MI Zhenxing and Dan Xu. Switch-nerf: Learning scene decomposition with mixture of experts for large-scale neural radiance fields. In *The Eleventh International Conference on Learning Representations*, 2022. 2, 3