S^3Gaussian: Self-Supervised Street Gaussians for Autonomous Driving

Nan Huang[1,2,*] **Xiaobao Wei**[2] **Wenzhao Zheng**[1,3,†] **Pengju An**[2] **Ming Lu**[2]
Wei Zhan[1] **Masayoshi Tomizuka**[1] **Kurt Keutzer**[1] **Shanghang Zhang**[2,‡]
https://wzzheng.net/S3Gaussian
[1]UC Berkeley [2]Peking University [3]Tsinghua University
wenzhao.zheng@outlook.com; shanghang@pku.edu.cn

Abstract

Photorealistic 3D reconstruction of street scenes is a critical technique for developing real-world simulators for autonomous driving. Despite the efficacy of Neural Radiance Fields (NeRF) for driving scenes, 3D Gaussian Splatting (3DGS) emerges as a promising direction due to its faster speed and more explicit representation. However, most existing street 3DGS methods require tracked 3D vehicle bounding boxes to decompose the static and dynamic elements for effective reconstruction, limiting their applications for in-the-wild scenarios. To facilitate efficient 3D scene reconstruction without costly annotations, we propose a self-supervised street Gaussian (S^3Gaussian) method to decompose dynamic and static elements from 4D consistency. We represent each scene with 3D Gaussians to preserve the explicitness and further accompany them with a spatial-temporal field network to compactly model the 4D dynamics. We conduct extensive experiments on the challenging Waymo-Open dataset to evaluate the effectiveness of our method. Our S^3Gaussian demonstrates the ability to decompose static and dynamic scenes and achieves the best performance without using 3D annotations. Code is available at: https://github.com/nnanhuang/S3Gaussian/.

1 Introduction

Autonomous driving has made significant progress in recent years and developed various techniques in each stage of its pipeline including perception [29, 67, 23, 56], prediction [18, 16, 31], and planning [11, 9, 10]. With the emergence of end-to-end autonomous driving which directly outputs the control signal from sensor inputs [19, 20, 24], open-loop evaluation of autonomous driving systems ceases to be effective and thus requires pressing improvement [65, 30]. As a promising solution, real-world closed-loop evaluation requires sensor inputs for controllable views, which motivates the development of high-quality scene reconstruction methods [53, 59].

Despite numerous efforts on photo-realistic reconstruction on small-scale scenes [35, 36, 7, 25, 55], the large-scale and highly dynamic characteristics of driving scenarios pose new challenges to the effective modeling of 3D scenes. To accommodate these, most existing works adopt tracked 3D bounding boxes to decompose static and dynamic elements [60, 58, 53]. Still, the costly annotations of 3D tracklets limit their applications for 3D modeling from in-the-wild data. EmerNerf [61] addressed this by simultaneously learning the scene flow and using it to connect corresponding points in the 4D NeRF field for multi-frame reconstruction, enabling the emergence of decomposition between static and dynamic objects without explicit bounding boxes. However, 3D driving scene modeling has been undergoing a shift from NeRF-based reconstruction to 3D Gaussian Splatting due

[*]Work done during an internship at UC Berkeley. †Project leader. ‡Corresponding author.

Preprint.

Figure 1: Qualitative comparison over Waymo-NOTR Datasets. On the left, we showcase results from novel view synthesis; on the right, results from dynamic scene reconstruction are displayed. With the proposed spatial-temporal network for the self-supervised scene decomposition, our method S^3Gaussian produces the best rendering quality with high fidelity and sharp details.

to its desire for low latency and explicit representation. Though EmerNerf demonstrated promising results, it can only be used for NeRF-based scene modeling, which takes a long time for training and rendering. It is still unclear how to achieve 3D Gaussian Splatting for urban scene reconstruction without explicit 3D supervision.

To address the above issues, we propose a **S**elf-**S**upervised **S**treet **Gaussians** named S^3Gaussian, offering a robust solution for dynamic street scenes without requiring 3D supervision. Specifically, to handle the complex spatial-temporal deformations inherent in driving scenes, S^3Gaussian introduces a cutting-edge spatial-temporal field for scene decomposition in a self-supervised manner. This spatial-temporal field incorporates a multi-resolution Hexplane structure encoder alongside a compact multi-head Gaussian decoder. The Hexplane encoder is designed to decompose the 4D input grid into multi-resolution, learnable feature planes, efficiently aggregating temporal and spatial information from the dynamic street scenes. During the optimization process, the multi-resolution Hexplane structure encoder effectively separates the entire scene, achieving a canonical representation for each scene. Dynamic-related features are stored within the spatial-temporal plane, while static-related features are retained in the spatial-only plane. Leveraging the densely encoded features, the multi-head Gaussian decoders calculate the deformation offsets from the canonical representations. These deformations are then added to the original 3D Gaussians' attributes, including position and spherical harmonics, allowing for a dynamic alteration of the scene representation conditioned on time series. Our main contributions are summarized as follows:

- We propose S^3Gaussian, the first self-supervised method that manages to decompose the dynamic and static 3D Gaussians in street scenes without extra manually annotated data.
- To model the complex changes in driving scenes, we introduce an efficient spatial-temporal decomposition network to automatically capture the deformation of 3D Gaussians.
- We conduct comprehensive experiments on challenging datasets, including NOTR and Waymo. Results demonstrate that S^3Gaussian achieves state-of-the-art rendering quality on scene reconstruction and novel view synthesis tasks.

2 Related Work

3D Gaussian Splatting. Recent breakthroughs in 3D Gaussian Splatting (3DGS) [26] have revolutionized scene modeling and rendering. Harnessing the power of explicit 3D Gaussians, 3DGS achieves optimal outcomes in novel view synthesis and real-time rendering while also substantially reducing parameter complexity compared to conventional representations such as meshes or voxels.

This technique seamlessly integrates the principles of point-based rendering [1] and splatting [70], facilitating rapid rendering and differentiable computation through splat-based rasterization.

While the original 3DGS model is designed for static scene representation, several researchers have extended its applicability to dynamic objects and scenes. For instance, Yang et al. [63] introduces a deformation network aimed at capturing Gaussian motion from a series of dynamic monocular images. Another approach, detailed by [57], establishes connections between neighboring Gaussians using a HexPlane, thereby enabling real-time rendering. By optimizing point clouds containing semantic logits and 3D Gaussians for novel dynamic scene representation, Yan et al. [60] achieves improvements in training and rendering speed. Similarly to NeRF's methodology, Zhou et al. [68] differentiates static backgrounds and dynamic objects within the scene and reconstructs each using distinct Gaussian Splatting methods. However, existing approaches are constrained as they can model only static or dynamic scenes individually and require supervised classification of scene types. Our objective is to autonomously learn the decomposition of static and dynamic scenes in a self-supervised manner, thereby eliminating the reliance on real annotations, such as dynamic object bounding boxes.

Street Scene Reconstruction for Autonomous Driving Simulation. Numerous efforts have been put into reconstructing scenes from autonomous driving data captured in real scenes. Existing self-driving simulation engines such as CARLA [12] or AirSim [45] suffer from costly manual effort to create virtual environments and the lack of realism in the generated data. The rapid development of Novel View Synthesis (NVS) techniques, including NeRF [35] and 3DGS [26], has attracted considerable attention within the arena of autonomous driving. Many studies [8, 17, 33, 34, 40, 43, 42, 49, 51, 53, 52, 58, 60, 62, 68] have investigated the application of these methods for reconstructing street scenes. Block-NeRF [49] and Mega-NeRF [52] propose segmenting scenes into distinct blocks for individual modeling. Urban Radiance Field [42] enhances NeRF training with geometric information from LiDAR, while DNMP [34] utilizes a pre-trained deformable mesh primitive to represent the scene. Streetsurf [17] divides scenes into close-range, distant-view, and sky categories, yielding superior reconstruction results for urban street surfaces. For modeling dynamic urban scenes, NSG [39] represents scenes as neural graphs, and MARS [58] employs separate networks for modeling background and vehicles, establishing an instance-aware simulation framework. With the introduction of 3DGS [26], DrivingGaussian [68] introduces Composite Dynamic Gaussian Graphs and incremental static Gaussians, while StreetGaussian [60] optimizes the tracked pose of dynamic Gaussians and introduces 4D spherical harmonics for varying vehicle appearances across frames.

The aforementioned methods not only suffer from prolonged training durations and sluggish rendering speeds but also fail to qualify the ability to divide dynamic and static scenes automatically. Therefore, we propose S^3Gaussian to differentiate between dynamic and static scenes in a self-supervised manner without the need for additional annotations, and perform high-fidelity and real-time neural rendering of dynamic urban street scenes, which is crucial for autonomous driving simulation.

3 Proposed Approach

We aim to learn a spatial-temporal representation of the dynamic environment of the street from a sequence of images captured by moving vehicles. However, due to the limited number of observation views and the high cost of obtaining ground truth annotations for dynamic and static objects, we aim to learn the scene decomposition of both static and dynamic components in a fully self-supervised manner, avoiding the supervision of extra annotations including bounding boxes for dynamic objects, segmentation masks for the scene decomposition, and optical flow for the motion perception.

To achieve these objectives, we propose a novel scene representation named S^3Gaussian. First, in Sec. 3.1, we lift 3D Gaussians to 4D to better represent dynamic and complex scenes. Then, in Sec. 3.2, we introduce a novel Spatial-temporal Field Network to integrate high-dimensional spatial-temporal information and decode them to transform 4D Gaussians. Finally, in Sec. 3.3, we describe the entire optimization process, eliminating extra annotations.

3.1 4D Gaussian Representations

As depicted in Figure 2, our scene representations include 3D Gaussians [26] $\mathcal{G}$ and a Spatial-temporal Field Network $\mathcal{F}$. To depict static scenes, 3D Gaussians are characterized by a covariance matrix Σ and a position vector $\mathcal{X}$, referred to as the geometric attributes. For a stable optimization, each

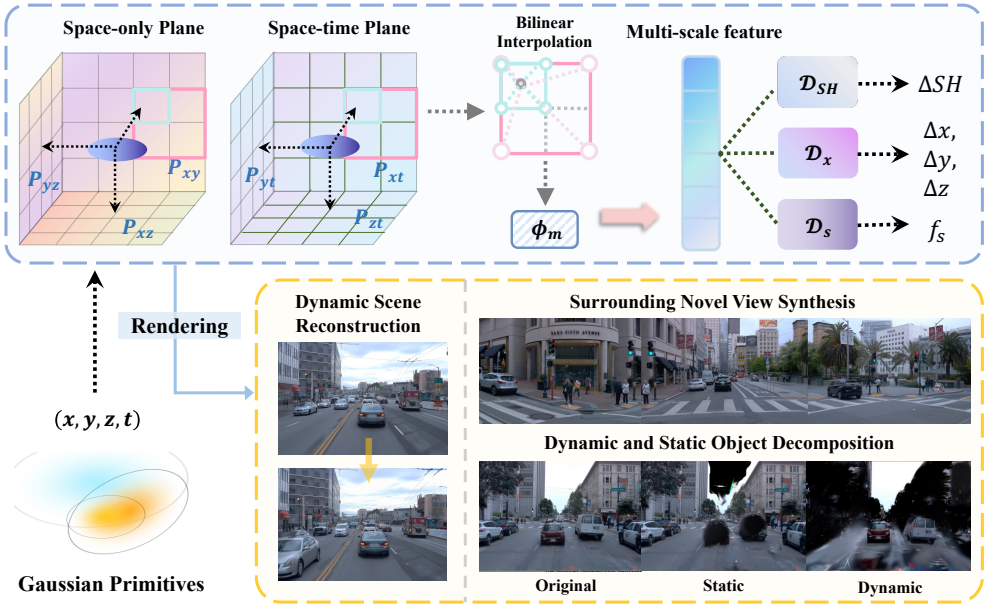

Figure 2: Pipeline of S^3Gaussian. To tackle the challenges in self-supervised street scene decomposition, our method consists of a Multi-resolution Hexplane Structure Encoder to encode 4D grid into feature planes and a multi-head Gaussian Decoder to decode them into deformed 4D Gaussians. The entire pipeline is optimized without extra annotations in a self-supervised manner, leading to superior scene decomposition ability and rendering quality.

covariance matrix is further factorized into a scaling matrix $\mathcal{S}$ and a rotation matrix $\mathcal{R}$:

$$\Sigma = \mathcal{R}\mathcal{S}\mathcal{S}^T\mathcal{R}^T \tag{1}$$

In addition to the position and covariance matrices, each Gaussian is also assigned an opacity value $\alpha \in \mathbb{R}$ and color $\mathcal{C} \in \mathbb{R}^{3(k+1)^2}$, defined by spherical harmonic (SH) coefficients, where k represents the degrees of SH functions.

The Spatial-temporal Field Network takes the position of each Gaussian $\mathcal{X}$ and the current timestep t as input, producing spatial-temporal features f. After decoding these features, the network can predict the displacement $\triangle\mathcal{G}$ of each point relative to canonical space while also obtaining semantic information f_s through the semantic feature decoder $\mathcal{D}_s$. We detail it in Sec. 3.2.

Following [64], we utilize a differentiable 3D Gaussian splatting renderer $\mathcal{R}$ to project the deformed 3D Gaussians $\mathcal{G}' = \triangle\mathcal{G} + \mathcal{G}$ into 2D [69]. Here, the covariance matrix Σ' in camera coordinates is:

$$\Sigma' = JW\Sigma W^T J^T \tag{2}$$

where J is the Jacobian matrix of the perspective projection, and W is the viewing transform matrix. The color of each pixel is calculated by N ordered points using α-blending:

$$C = \sum_{i \in N} c_i \alpha_i \prod_{j=1}^{i-1}(1 - \alpha_i) \tag{3}$$

Here, α_i and c_i represent the opacity and color of one point, computed by an optimizable per-point opacity and SH color coefficients with the view direction. The semantic map can be rendered simply by changing the color c in Eq. 3 to the semantic feature f_s.

3.2 Spatial-temporal Field Network

The primary focus of vanilla 3D Gaussians Splatting is on tasks in static scenes. However, the real world is dynamic, especially in contexts like autonomous driving. This makes the transition from 3DGS to 4D a crucial and challenging endeavor. Firstly, in dynamic scenarios, the views captured by each moving camera at each time step are sparser than in static scenes, making individual modeling

of each time step exceptionally difficult due to this sparsity. Therefore, it becomes imperative to consider information sharing across time steps [14].

Moreover, modeling all Gaussian points in space and time is impractical for large-scale or long-duration scenarios like autonomous driving due to significant memory overhead. Hence, we propose leveraging an efficient Gaussian-based spatial-temporal network to model 3D Gaussian motion. This network comprises a Multi-resolution Hexplane Structure Encoder and a minimal Multi-head Gaussian Decoder. It only needs to maintain a set of canonical 3D Gaussians and model a deformation field for each timestep. This field predicts displacement and color changes relative to the canonical space 3D Gaussians, thus capturing Gaussian motion [57]. Additionally, we incorporate a simple semantic field to assist in automatically decomposing static and dynamic Gaussians.

Multi-resolution Hexplane Structure Encoder. To efficiently aggregate temporal and spatial information across timesteps, considering that adjacent Gaussians often share similar spatial and temporal characteristics, we employ the Multi-resolution Hexplane Structure Encoder $\mathcal{E}$ with a tiny MLP ϕ_m to represent dynamic 3D scenes effectively inspired by [6, 13, 14, 46]. Specifically, the HexPlane decomposes the 4D spatial-temporal grid into six multi-resolution learnable feature planes spanning each pair of coordinate axes, each endowed with an orthogonal axis. The first three planes $\mathcal{P}_{xy}$, $\mathcal{P}_{xz}$, $\mathcal{P}_{yz}$ represent spatial-only dimensions, while the latter three $\mathcal{P}_{xt}$, $\mathcal{P}_{yt}$, $\mathcal{P}_{zt}$ represent spatial-temporal variations. This decoupling of time and space is beneficial for separating static and dynamic elements. Dynamic objects become distinctly visible on the spatial-temporal plane, while static objects solely manifest on the spatial-only plane.

Additionally, to promote spatial smoothness and coherence while compressing the model and reducing the number of features stored at the highest resolution, inspired by Instant-NGP's multi-scale hash encoding [**?**], our hexplane encoder comprises multiple copies of different resolutions. This representation effectively encodes spatial features at various scales. Therefore, our formulation is:

$$\mathcal{P}_{ij}^{\rho} \in \mathbb{R}^{d \times \rho r_i \times \rho r_j}, (i,j) \in \{(x,y),(x,z),(y,z),(x,t),(y,t),(z,t)\}, \rho \in \{1,2\} \tag{4}$$

where d is the hidden dimension of features, ρ stands for the upsampling scale, and r equals to the basic resolution. Giving a 4D coordinate (x, y, z, t), we then obtain the neural voxel features and merge all the features using a tiny MLP ϕ_m as follows:

$$f(x,y,z,t) = \phi_m(\bigcup_{\rho} \prod \pi(\mathcal{P}_{ij}^{\rho}, \psi_{ij}^{\rho}(x,y,z,t))) \tag{5}$$

where ψ_{ij}^{ρ} projects 4D coordinate (x, y, z, t) onto the corresponding plane, and π denotes bilinear interpolation, used for querying voxel features located at the four vertices. We merge the planes using Hadamard product to produce spatially localized signals, as discussed in [14].

Multi-head Gaussian Decoder. We use separate MLP heads $\mathcal{D} = (\mathcal{D}_{SH}, \mathcal{D}_x, \mathcal{D}_s)$ to decode the features obtained in Sec. 3.2. Specifically, we employ a semantic feature decoder to compute semantic features $f_s = \mathcal{D}_s(f(x,y,z,t))$. Considering that most autonomous driving scenarios involve rigid motion, we only consider deformation in the position of the Gaussians, thus $\triangle x = \mathcal{D}_x(f(x,y,z,t))$. Additionally, considering factors like illumination, the appearance of the scene varies with its global position and time. Therefore, we also introduce an SH coefficient head to model the 4D dynamic appearance model $\triangle SH = \mathcal{D}_{SH}(f(x,y,z,t))$. Finally, our deformed 4D Gaussians are formulated as: $\mathcal{G}' = \{\mathcal{X} + \triangle\mathcal{X}, \mathcal{C} + \triangle\mathcal{C}, s, r, \sigma, f_s\}$.

3.3 Self-supervised Optimization

LiDAR Prior Initialization. To initialize the positions of the 3D Gaussians, we leverage the LiDAR point cloud captured by the vehicle instead of using the original SFM [44] point cloud to provide a better geometric structure. To reduce model size, we also downsample the entire point cloud by voxelizing it and filtering out points outside the image. For colors, we initialize them randomly.

Optimization Objective. The loss function of our method consists of seven parts, and we jointly optimize our scene representation and Spatial-temporal field using it. $\mathcal{L}_{rgb}$ is the L1 loss between rendered and ground truth images and $\mathcal{L}_{ssim}$ measures the similarity between them. $\mathcal{L}_{depth}$ is the L2 loss between the estimated depth map from the LiDAR point cloud and the rendered depth map, used to supervise the expected position of the Gaussians [61, 68]. The rendered depth is computed using the positions of the Gaussians. $\mathcal{L}_{feat}$ is the L2 loss of semantic feature. Following [13, 47, 14], we

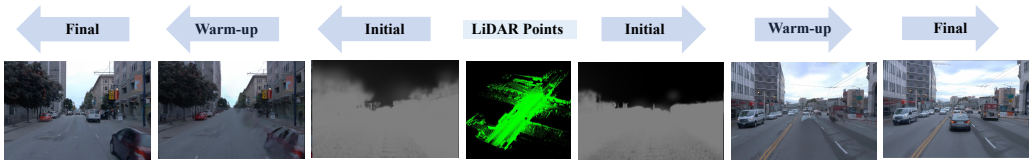

Figure 3: Illustration of the optimization process. With the LiDAR points initialization and the static 3D Gaussian Warm-up strategy, our model achieves high-quality 4D Gaussian representations of the complex dynamic scenes.

also introduce a grid-based total-variational loss $\mathcal{L}_{tv}$. Given that most elements in the scene are static, we introduce regularization constraints into the spatial-temporal network to enhance the separation of static and dynamic components. We achieve this by minimizing the expectation of $\mathbb{E}(\triangle\mathcal{X})$ and $\mathbb{E}(\triangle\mathcal{C})$, which encourages the network only to produce offset values when necessary. Then, the total loss function can be formulated as follows:

$$\mathcal{L} = \lambda_{rgb}\mathcal{L}_{rgb} + \lambda_{depth}\mathcal{L}_{depth} + \lambda_{feat}\mathcal{L}_{feat} + \lambda_{ssim}\mathcal{L}_{ssim} + \lambda_{tv}\mathcal{L}_{tv} + \lambda^{x}_{reg}\mathcal{L}^{x}_{reg} + \lambda^{y}_{reg}\mathcal{L}^{c}_{reg} \quad (6)$$

where $\lambda_{rgb} = 1.0$, $\lambda_{depth} = 0.1$, $\lambda_{feat} = 0.1$, $\lambda_{ssim} = 0.1$, $\lambda_{tv} = 0.1$, $\lambda^{x}_{reg} = 0.01$, and $\lambda^{y}_{reg} = 0.01$ are the weights assigned to each loss component.

4 Experiments

In this section, we primarily discuss the experimental methodology used to evaluate the performance of our S^3Gaussian. Details of the dataset settings, baseline methods, and implementation specifics are provided in Sec. 4.1. In Sec. 4.2, we compare our approach with state-of-the-art (SOTA) methods across various tasks. Further ablation studies and analysis are detailed in Sec. 4.3.

4.1 Experimental Setup

Datasets. NOTR dataset is a subset of the Waymo Open dataset [48] curated by [61], which comprises many challenging driving scenarios: ego-static, high-speed, exposure mismatch, dusk/dawn, gloomy, rainy, and night scenes. In contrast, many public datasets with LiDAR data suffer from a severe imbalance, eg. nuScenes [4] and nuPlan [5], predominantly featuring simple scenes with few dynamic objects. Therefore, we utilize NOTR's dynamic32 (D32) and static32 (S32) datasets, totaling 64 scenes, to obtain a balanced and diverse standard for evaluating our static and dynamic reconstruction. Furthermore, since most baseline methods are NeRF-based, to ensure a fair evaluation of our method's performance, we conduct comparisons with the current state-of-the-art Gaussian-based method, StreetGaussian [60]. We adhere to the dataset configuration used by StreetGaussian, employing the six scenes selected from the Waymo Open dataset [48], which are characterized by complex environments and significant object motion.

Baseline Methods. We evaluate our approach against state-of-the-art methods, including NeRF-based models and 3DGS-based models. MARS [58] is a modular [50] simulator based on NeRF, utilizing 2D bounding boxes to train NeRF for static and dynamic objects respectively. NSG [40] learns latent codes to model moving objects with a shared decoder. EmerNeRF [61] also builds upon NeRF but self-supervises the modeling of dynamic scenes by optimizing flow fields, representing the current SOTA in self-supervised learning for dynamic driving scene representations. The 3DGS [26] model employs anisotropic 3D Gaussian ellipsoids as an explicit 3D scene representation, achieving the strongest performance across various tasks in static scenes. StreetGaussian[60], the latest Gaussian-based method, introduces time into SH coefficients, reaching SOTA performance as well, albeit also utilizing 2D tracked boxes. For a fair comparison, we also apply LiDAR point cloud initialization to 3DGS, and depth regularization to 3DGS and MARS, mirroring our approach.

Implementation Details. We train our model for 50,000 iterations using the Adam optimizer [27], following the learning rate configurations of 3D Gaussians [26]. Additionally, we employ 5,000 steps of pure static 3D Gaussian training [26] as a warm-up for the scene [57], as illustrated in Figure 3. For the reconstruction of long sequence scenes, we divide the scene into multiple clips. Specifically, we use 50 frames per clip, where the optimized Spatial-temporal field serves as the initialization for the Spatial-temporal field of the next sequence with 50 steps. This approach maintains spatial and temporal consistency across sequences within the same scene. The basic resolution for our

Table 1: Overall performance of our methods with existing SOTA approaches on the Waymo-NOTR dataset[61]. "PSNR*" and "SSIM*" denote the PSNR and SSIM of dynamic objects respectively. The best and the second best results are denoted by pink and blue.

Data	Metrics	Scene Reconstruction				Novel View Synthesis			
		3DGS	MARS	EmerNeRF	Ours	3DGS	MARS	EmerNeRF	Ours
D32	PSNR↑	28.47	28.24	28.16	31.35	25.14	26.61	25.14	27.44
	SSIM↑	0.876	0.866	0.806	0.911	0.813	0.796	0.747	0.857
	LPIPS↓	0.136	0.252	0.228	0.106	0.165	0.305	0.313	0.137
	PSNR*↑	23.26	23.37	24.32	26.02	20.48	22.21	23.49	22.92
	SSIM*↑	0.716	0.701	0.682	0.783	0.753	0.697	0.660	0.680
S32	PSNR↑	29.42	28.31	30.00	30.73	26.82	27.63	28.89	27.05
	SSIM↑	0.891	0.879	0.834	0.883	0.836	0.848	0.814	0.825
	LPIPS↓	0.118	0.196	0.201	0.116	0.134	0.193	0.212	0.142

Table 2: Quantitative results on StreetGaussian datasets [60]. We strictly follow the experimental setting of it and borrow results from it since it has not been open-sourced.

Metrics	3D GS	NSG	MARS	EmerNeRF	StreetGaussian	Ours
PSNR↑	29.64	28.31	31.37	32.34	34.96	34.61
SSIM↑	0.918	0.862	0.904	0.886	0.945	0.95z0
LPIPS↓	0.117	0.346	0.246	0.142	0.068	0.050
PSNR*↑	16.48	19.55	23.07	25.71	25.46	25.78

multi-resolution HexPlane encoder is set to 64, then upsampled by 2 and 4 as [57]. The learning rate of it is set as 1.6×10^{-3}, decayed to 1.6×10^{-4} at the end of trainging. Each decoder in the multi-head decoder is a small MLP with the same learning rate as the HexPlane encoder. Other hyperparameters are kept consistent with 3DGS[26]. In the experiments conducted on the Waymo-NOTR dataset, we strictly adhered to the experimental settings of EmerNeRF [61]. Similarly, for the Waymo-Street dataset, our experimental setup closely followed StreetGaussian [60].

4.2 Comparisons with the State-of-the-art

The results on the Waymo-NOTR dataset demonstrate that our approach consistently outperforms other methods in scene reconstruction and novel view synthesis, as shown in Table 1. For the static32 dataset, we utilize PSNR, SSIM, and LPIPS [66] as metrics to evaluate rendering quality. For the dynamic32 dataset, we additionally include PSNR* and SSIM* metrics focusing on dynamic objects. Specifically, we project the 3D bounding boxes of dynamic objects onto the 2D image plane and calculate pixel loss only within the projected boxes as [61, 60]. Our metrics outperform those of other existing methods, indicating the superior performance of our approach in modeling dynamic objects. Moreover, although static scene representation is not our primary focus, our method also performs exceptionally well in this aspect. Thus, our approach is more versatile and general.

We also conducted qualitative comparisons, as shown in Figure 1. We emphasized regions with significant differences to provide a clearer demonstration. From the figure, it is evident that our method surpasses the state-of-the-art (SOTA) in both the synthesis of new viewpoints (left side of Figure 1) and reconstruction (right side of Figure 1) of static and dynamic scenes. Although 3DGS [26] faithfully reconstructs static objects, it fails when dealing with dynamic objects and struggles with reconstructing distant skies. The reconstruction quality of MARS [58] is poor, being effective only for very short sequences, and it struggles to reconstruct fast-moving objects. While EmerNeRF [61] can self-supervise the reconstruction of static and dynamic objects, the reconstruction quality is unsatisfactory, with issues such as ghosting, loss of plant texture details, missing lane markings, and blurry distant scenes. For novel view synthesis, our method can generate high-quality rendered images and ensure consistency between multiple camera views. In dynamic scene reconstruction, we accurately simulate dynamic objects in large-scale scenes, particularly distant dynamic objects, and mitigate issues such as loss, ghosting, or blurriness associated with these dynamic elements.

Table 2 presents the results on the dataset collected by StreetGaussian [60]. StreetGaussian is a state-of-the-art method for Gaussian-based dynamic object representation. Our approach performs

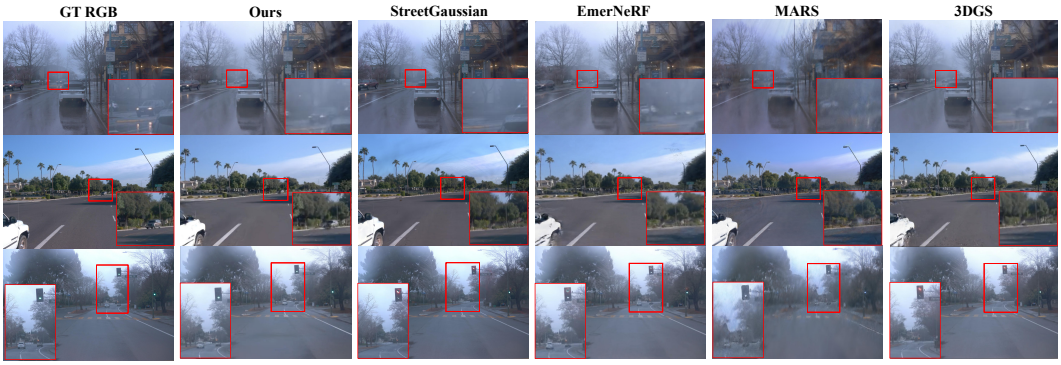

Figure 4: Qualitative comparison over Waymo-Street Datasets [60]. All results are from novel view synthesis. Compared to StreetGaussian [60], our method demonstrates a stronger ability to self-supervisedly reconstruct distant dynamic objects and is more sensitive to changes in scene details.

Table 3: Quantitative ablation studies on Waymo-NOTR dynamic32 datasets.

Task	Metrics	w/o $\mathcal{P}_{ij}^{\rho}$	w/o $\mathcal{D}_x$	w/o $\mathcal{D}_{SH}$	w/o $\mathcal{D}_s$	w/o Warm-up	Ours
Scene Reconstruct	PSNR↑	18.702	29.861	31.458	31.605	31.390	**32.135**
	SSIM↑	0.4793	0.8871	0.9157	0.9174	0.9173	**0.9355**
	PSNR*↑	16.800	24.626	26.420	26.556	26.628	**27.046**
	SSIM*↑	0.3627	0.7521	0.8162	0.8182	0.8213	**0.8284**
NVS	PSNR↑	17.245	25.850	27.959	27.981	27.955	**28.417**
	SSIM↑	0.4499	0.8174	0.8616	0.8624	**0.8641**	**0.8641**
	PSNR*↑	15.613	21.385	21.385	23.402	23.681	**23.974**
	SSIM*↑	0.3118	0.6386	0.6386	0.7138	0.7117	**0.7175**

similarly to StreetGaussian, but with the distinction that StreetGaussian uses additional bounding boxes to model dynamic objects, whereas our approach does not require any explicit supervision. As shown in Figure 4, compared to StreetGaussian [60] which uses explicit supervision, our method excels in self-supervised reconstruction of distant dynamic objects. Additionally, our method is more sensitive to changes in scene details, such as variations in traffic lights. Furthermore, StreetGaussian exhibits noise in the sky, resulting in a decrease in rendering quality.

4.3 Ablation and Analysis

We investigate the effectiveness of our method and its various components. Due to time constraints, we select 20 sequences from NOTR dynamic32 [61] for analysis, and all models are trained for a shorter duration of $30,000$ iterations. Table 3 presents the quantitative results, while Figure 5 showcases the visual comparison results.

Multi-resolution Hexplane Structure Encoder. Compared to purely explicit methods, the proposed HexPlane encoder $\mathcal{P}_{ij}^{\rho}$ allows for memory savings and enables retention of different dimensions of spatial-temporal information in the scene through various resolutions. Discarding this module and relying solely on a shallow MLP ϕ_m fails to accurately establish spatial-temporal fields and cannot simulate Gaussian deformations. Both Table 3 and Figure 5 demonstrate this, without this module, our rendering quality sharply declines. We also provide visualizations of the features of this encoder, as shown in Figure 6. As an explicit module, we can easily optimize all Gaussian features on a single voxel plane. From Figure 6, it is evident that the voxel plane features mainly concentrate on the moving parts of the scene. The trajectories of moving vehicles in the scene extend from the bottom-right to the top-right corner. As a result, spatial plane features are primarily concentrated in the bottom-right corner, whereas temporal plane features are predominantly observed on the right side. These patterns demonstrate that our encoder successfully captures both spatial and temporal information. This capability allows us to effectively self-supervise the decomposition of static and dynamic components, as illustrated in Figure 6 and Figure 2.

Multi-head Gaussian Decoder. Our proposed multi-head Gaussian decoder can decode voxel features. As indicated in Table 3, disabling this component would impact rendering quality greatly. Additionally, as shown in Figure 5, disabling the $\mathcal{D}_x$ decoder and only training Gaussian in canonical

Figure 5: Visual ablation results on the Waymo-NOTR dynamic32 dataset.

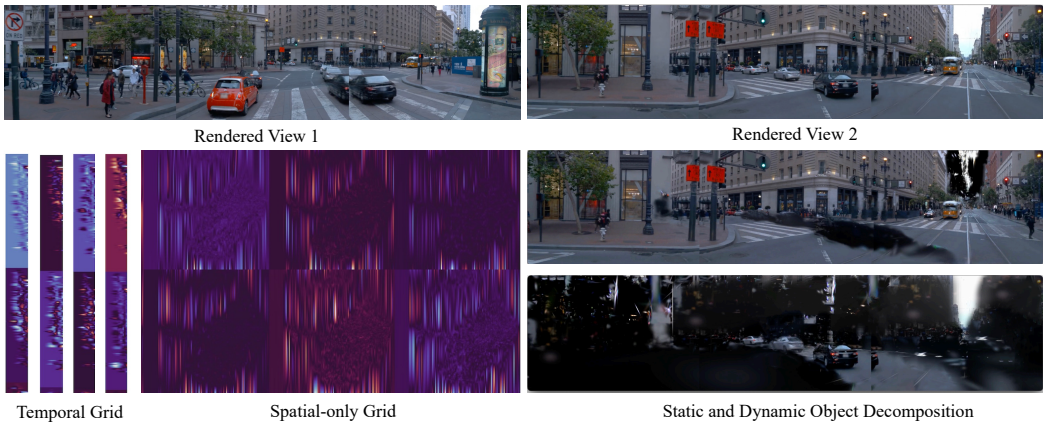

Figure 6: Visualization of HexPlane voxel grids, showcasing its capability to decompose static and dynamic elements. Spatial-only grid refers to the spatial voxel parameters, while the temporal grid refers to its time features.

space would introduce significant noise. The noise stems from Gaussian points initialized by LiDAR point clouds, resulting in a series of Gaussian points along a moving vehicle's trajectory. If these points are not deformed, it becomes challenging to optimize them afterward. Furthermore, omitting the semantic feature decoder $\mathcal{D}_s$ and color deformation decoder $\mathcal{D}_{SH}$ primarily affects rendering details. For example, the geometric structure of the truck becomes blurrier without these components.

Static Gaussian Warm-up. According to Figure 5, we found that directly training the 4D Gaussians without first optimizing 3D Gaussians for warm-up not only reduces convergence speed but also affects the final rendering quality. As shown in the 3, performing a warm-up step already yields basic static scene reconstruction, which alleviates the pressure on the 4D spatial-temporal network to learn large-scale scenes and allows the network to focus more on dynamic parts. Additionally, it stabilizes the network by avoiding early-stage numerical errors [57].

5 Conclusion

In this paper, we propose S^3Gaussian, the first self-supervised street Gaussian method to differentiate dynamic and static elements in complex driving scenes. S^3Gaussian employs a Spatial-temporal Field Network to achieve the scene decomposition automatically, which consists of a Multi-resolution Hexplane Structure Encoder and a Multi-head Gaussian Decoder. Given a 4D grid in global space, the proposed Hexplane encoder aggregates features into dynamic or static planes. Then we decode these features into the deformed 4D Gaussians. The entire pipeline is optimized without any extra annotations. Experiments on challenging datasets including NOTR and Waymo improve that S^3Gaussian show superior scene decomposition ability and obtain the state-of-the-art rendering quality across different tasks. Abundant quantitative results are implemented to shed light on the effectiveness of each component in S^3Gaussian.

A Appendix

A.1 Additional Implementation Details

Datasets Details. Our Waymo-NOTR dataset follows the setup of [61]. For camera images, we utilize three frontal cameras: FRONT LEFT, FRONT, and FRONT RIGHT, adjusted to a resolution of 640 $\times$ 960 for training and evaluation. The length of all sequences is set to 100 frames. We select every 10th frame from the sequences as the test frames and use the remaining frames for training. For our Waymo-Street dataset, consistent with [60], we use frontal cameras and downscale the input images to 1066 $\times$ 1600 for evaluating monocular reconstruction and novel view synthesis capabilities. The length of all sequences strictly follows the dataset setting released by StreetGaussian [60], with each sequence approximately 100 frames long. We select every 4th frame from the sequences as the test frames and use the remaining frames for training.

Feature Extraction. We employ the DINOv2 [38] checkpoint and the feature extractor implementation by [2]. Specifically, we use the ViT-B/14 variant and adjust the image dimensions to 644$\times$966 with a stride of 7. Given the large size of the feature maps, following [61] we use PCA decomposition to reduce the feature dimension from 768 to 3 and normalize these features to the [0,1] range.

A.2 More Related Work (Advances in Neural Radiance Fields)

In recent years, there has been a surge of interest among researchers in leveraging neural rendering techniques for scene modeling. Among these techniques, Neural Radiance Fields (NeRF) have garnered particular attention. NeRF [35] utilizes differentiable volume rendering methods, facilitating the generation of novel scenes from a mere collection of planar images accompanied by their respective camera poses. Moreover, NeRF demonstrates the capability to segregate street views into static and dynamic scenes by tracking the bounding boxes of vehicles. Despite the extensive research efforts aimed at enhancing NeRF's functionalities, which have led to notable advancements in training speed [14, 15, 37], pose optimization [3, 32, 54], scene editing [28, 43], object generation [21], and dynamic scene representation [22, 41], challenges persist, particularly regarding training and rendering speed. These challenges pose significant obstacles to the widespread adoption of NeRF in autonomous driving scenarios. Compared to NeRF-based methods, S^3Gaussians proposed 4D Gaussian representations for dynamic scenes, significantly boosting rendering speed.

A.3 Limitations

Similar to other methods [61, 58], our scene encounters difficulty in modeling objects moving at high speeds. We suspect this may be due to the deformation field's high variance, rendering it unable to model their rapid movements accurately. Moreover, views of rapidly moving dynamic objects are typically sparse, with only a few views available for capture, making reconstruction even more challenging. How to reconstruct these challenging scenes will be a focus of our future research.

References

[1] Kara-Ali Aliev, Dmitry Ulyanov, and Victor S. Lempitsky. Neural point-based graphics. *ArXiv*, abs/1906.08240, 2019.

[2] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2(3):4, 2021.

[3] Wenjing Bian, Zirui Wang, Kejie Li, Jiawang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4160–4169, 2022.

[4] 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 *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[5] Holger Caesar, Juraj Kabzan, KokSeang Tan, FongWhye Kit, EricM. Wolff, AlexH. Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning

benchmark for autonomous vehicles. *arXiv: Computer Vision and Pattern Recognition,arXiv: Computer Vision and Pattern Recognition*, 2021.

[6] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 130–141, 2023.

[7] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision*, pages 333–350. Springer, 2022.

[8] Yurui Chen, Chun Gu, Junzhe Jiang, Xiatian Zhu, and Li Zhang. Periodic vibration gaussian: Dynamic urban scene reconstruction and real-time rendering. *ArXiv*, abs/2311.18561, 2023.

[9] Jie Cheng, Yingbing Chen, Qingwen Zhang, Lu Gan, Chengju Liu, and Ming Liu. Real-time trajectory planning for autonomous driving with gaussian process and incremental refinement. In *ICRA*, pages 8999–9005, 2022.

[10] Jie Cheng, Xiaodong Mei, and Ming Liu. Forecast-MAE: Self-supervised pre-training for motion forecasting with masked autoencoders. *ICCV*, 2023.

[11] Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *CoRL*, 2023.

[12] Alexey Dosovitskiy, Germán Ros, Felipe Codevilla, Antonio M. López, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on Robot Learning*, 2017.

[13] Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner, and Qi Tian. Fast dynamic radiance fields with time-aware neural voxels. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022.

[14] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12479–12488, 2023.

[15] Stephan J. Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien P. C. Valentin. Fastnerf: High-fidelity neural rendering at 200fps. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14326–14335, 2021.

[16] Junru Gu, Chenxu Hu, Tianyuan Zhang, Xuanyao Chen, Yilun Wang, Yue Wang, and Hang Zhao. Vip3d: End-to-end visual trajectory prediction via 3d agent queries. *arXiv preprint arXiv:2208.01582*, 2022.

[17] 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*, abs/2306.04988, 2023.

[18] Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. Fiery: Future instance prediction in bird's-eye view from surround monocular cameras. In *ICCV*, 2021.

[19] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *ECCV*, 2022.

[20] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *CVPR*, pages 17853–17862, 2023.

[21] Nan Huang, Ting Zhang, Yuhui Yuan, Dong Chen, and Shanghang Zhang. Customize-it-3d: High-quality 3d creation from a single image using subject-specific knowledge prior, 2024.

[22] Xin Huang, Qi Zhang, Feng Ying, Hongdong Li, Xuan Wang, and Qing Wang. Hdr-nerf: High dynamic range neural radiance fields. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18377–18387, 2021.

[23] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. In *CVPR*, pages 9223–9232, 2023.

[24] Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. *arXiv preprint arXiv:2303.12077*, 2023.

[25] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.

[26] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023.

[27] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[28] Yuan Li, Zhi Lin, David W. Forsyth, Jia-Bin Huang, and Shenlong Wang. Climatenerf: Extreme weather synthesis in neural radiance field. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3204–3215, 2022.

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

[30] Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahan Li, Jan Kautz, Tong Lu, and Jose M Alvarez. Is ego status all you need for open-loop end-to-end autonomous driving? In *CVPR*, 2024.

[31] Ming Liang, Bin Yang, Wenyuan Zeng, Yun Chen, Rui Hu, Sergio Casas, and Raquel Urtasun. Pnpnet: End-to-end perception and prediction with tracking in the loop. In *CVPR*, 2020.

[32] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5721–5731, 2021.

[33] Jeffrey Yunfan Liu, Yun Chen, Ze Yang, Jingkang Wang, Sivabalan Manivasagam, and Raquel Urtasun. Real-time neural rasterization for large scenes. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8382–8393, 2023.

[34] Fan Lu, Yan Xu, Guang-Sheng Chen, Hongsheng Li, Kwan-Yee Lin, and Changjun Jiang. Urban radiance field representation with deformable neural mesh primitives. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 465–476, 2023.

[35] 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.

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

[37] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, page 1–15, 2022.

[38] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[39] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2855–2864, 2020.

[40] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[41] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10313–10322, 2020.

[42] Konstantinos Rematas, An Liu, Pratul P. Srinivasan, Jonathan T. Barron, Andrea Tagliasacchi, Thomas A. Funkhouser, and Vittorio Ferrari. Urban radiance fields. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12922–12932, 2021.

[43] Viktor Rudnev, Mohamed A. Elgharib, William H. B. Smith, Lingjie Liu, Vladislav Golyanik, and Christian Theobalt. Nerf for outdoor scene relighting. In *European Conference on Computer Vision*, 2021.

[44] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[45] S. Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *International Symposium on Field and Service Robotics*, 2017.

[46] Ruizhi Shao, Zerong Zheng, Hanzhang Tu, Boning Liu, Hongwen Zhang, and Yebin Liu. Tensor4d: Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16632–16642, 2023.

[47] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[48] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Sheng Zhao, Shuyang Cheng, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset, 2020.

[49] 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. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8238–8248, 2022.

[50] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David Mcallister, Justin Kerr, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings*. ACM, 2023.

[51] Adam Tonderski, Carl Lindstrom, Georg Hess, William Ljungbergh, Lennart Svensson, and Christoffer Petersson. Neurad: Neural rendering for autonomous driving. *ArXiv*, abs/2311.15260, 2023.

[52] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-nerf: Scalable construction of large-scale nerfs for virtual fly- throughs. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12912–12921, 2021.

[53] Haithem Turki, Jason Y Zhang, Francesco Ferroni, and Deva Ramanan. Suds: Scalable urban dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12375–12385, 2023.

[54] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf-: Neural radiance fields without known camera parameters. *ArXiv*, abs/2102.07064, 2021.

[55] Xiaobao Wei, Renrui Zhang, Jiarui Wu, Jiaming Liu, Ming Lu, Yandong Guo, and Shanghang Zhang. Noc: High-quality neural object cloning with 3d lifting of segment anything. *arXiv preprint arXiv:2309.12790*, 2023.

[56] Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In *ICCV*, pages 21729–21740, 2023.

[57] 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*, abs/2310.08528, 2023.

[58] Zirui Wu, Tianyu Liu, Liyi Luo, Zhide Zhong, Jianteng Chen, Hongmin Xiao, Chao Hou, Haozhe Lou, Yuantao Chen, Runyi Yang, Yuxin Huang, Xiaoyu Ye, Zike Yan, Yongliang Shi, Yiyi Liao, and Hao Zhao. Mars: An instance-aware, modular and realistic simulator for autonomous driving. *CICAI*, 2023.

[59] 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.

[60] Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians for modeling dynamic urban scenes. *ArXiv*, abs/2401.01339, 2024.

[61] Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, and Yue Wang. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision, 2023.

[62] Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1389–1399, 2023.

[63] Ziyi Yang, Xinyu Gao, Wenming Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *ArXiv*, abs/2309.13101, 2023.

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

[65] Jiang-Tian Zhai, Ze Feng, Jinhao Du, Yongqiang Mao, Jiang-Jiang Liu, Zichang Tan, Yifu Zhang, Xiaoqing Ye, and Jingdong Wang. Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes. *arXiv preprint arXiv:2305.10430*, 2023.

[66] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[67] Yunpeng Zhang, Zheng Zhu, Wenzhao Zheng, Junjie Huang, Guan Huang, Jie Zhou, and Jiwen Lu. Beverse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving. *arXiv preprint arXiv:2205.09743*, 2022.

[68] Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. *ArXiv*, abs/2312.07920, 2023.

[69] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Surface splatting. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 371–378, 2001.

[70] Matthias Zwicker, Hanspeter Pfister, Jeroen van Baar, and Markus H. Gross. Ewa splatting. *IEEE Trans. Vis. Comput. Graph.*, 8:223–238, 2002.