2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Self-Supervised Street Gaussians for Autonomous Driving

Nan Huang[1,*], Xiaobao Wei[1,*], Wenzhao Zheng[2,3,†], Pengju An[1], Ming Lu[1],
Wei Zhan[2], Masayoshi Tomizuka[2], Kurt Keutzer[2], Shanghang Zhang[1,✉]

Fig. 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.

***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.**

[1]State Key Laboratory of Multimedia Information Processing, School of Computer Science, Peking University. [2]UC Berkeley. [3]Tsinghua University.
*Equal contribution.
†Project leader.
✉Corresponding author: shanghang@pku.edu.cn

I. Introduction

Autonomous driving has made significant progress in recent years and developed various techniques in each stage of its pipeline including perception [20], [52], [16], [41], prediction [14], [12], [22], and planning [8], [6], [7]. With the emergence of end-to-end autonomous driving which directly outputs the control signal from sensor inputs [15], [17], open-loop evaluation of autonomous driving systems ceases to be effective and thus requires pressing improvement [49], [21]. 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. [39], [44].

Despite numerous efforts on photo-realistic reconstruction on small-scale scenes [24], [25], [4], [18], [40], 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 [45], [43], [39]. Still, the costly annotations of 3D tracklets limit their applications for 3D modeling from in-the-wild data. EmerNerf [46] 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 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 3D Gaussians' attributes, including position and spherical harmonics, allowing for a dynamic alteration of the scene representation conditioned on time series.

II. Related Work

Street Scene Reconstruction for Autonomous Driving Simulation. In recent years, a lot of effort has been put into reconstructing scenes from autonomous driving data captured in real scenes. Existing self-driving simulation engines such as CARLA [9] or AirSim [31] 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 [24] and 3DGS [18], has attracted considerable attention within the arena of autonomous driving. Numerous studies [29], [37], [39], [47] have investigated the application of these methods for reconstructing street scenes. Block-NeRF [35] and Mega-NeRF [38] propose segmenting scenes into distinct blocks for individual modeling. Urban Radiance Field [28] enhances NeRF training with geometric information from LiDAR, while DNMP [23] utilizes a pre-trained deformable mesh primitive to represent the scene. Streetsurf [13] divides scenes into close-range, distant-view, and sky categories, yielding superior reconstruction results for urban street surfaces. In terms of geometric understanding, OccNeRF [50] advances volumetric rendering by leveraging temporal photometric consistency to reconstruct 3D occupancy in unbounded scenes without explicit 3D supervision. For modeling dynamic urban scenes, NSG [27] represents scenes as neural graphs, and MARS [43] employs separate networks for modeling background and vehicles, establishing an instance-aware simulation framework. With the introduction of 3DGS [18], DrivingGaussian [53] introduces Composite Dynamic Gaussian Graphs and incremental static Gaussians, while StreetGaussian [45] optimizes the tracked pose of dynamic Gaussians and introduces 4D SH (spherical harmonics) for varying vehicle appearances across frames. However, these methods fail to qualify the ability to divide dynamic and static scenes automatically. PVG [5] pioneered this extension by introducing periodic vibration-based temporal dynamics for unified representation of both static and dynamic elements, but have poor render performance.

Therefore, we propose S^3Gaussian, which can 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.

III. Methodology

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.

A. 4D Gaussian Representations

As depicted in Figure 2, our scene representations include 3D Gaussians [18] $\mathcal{G}$ 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 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

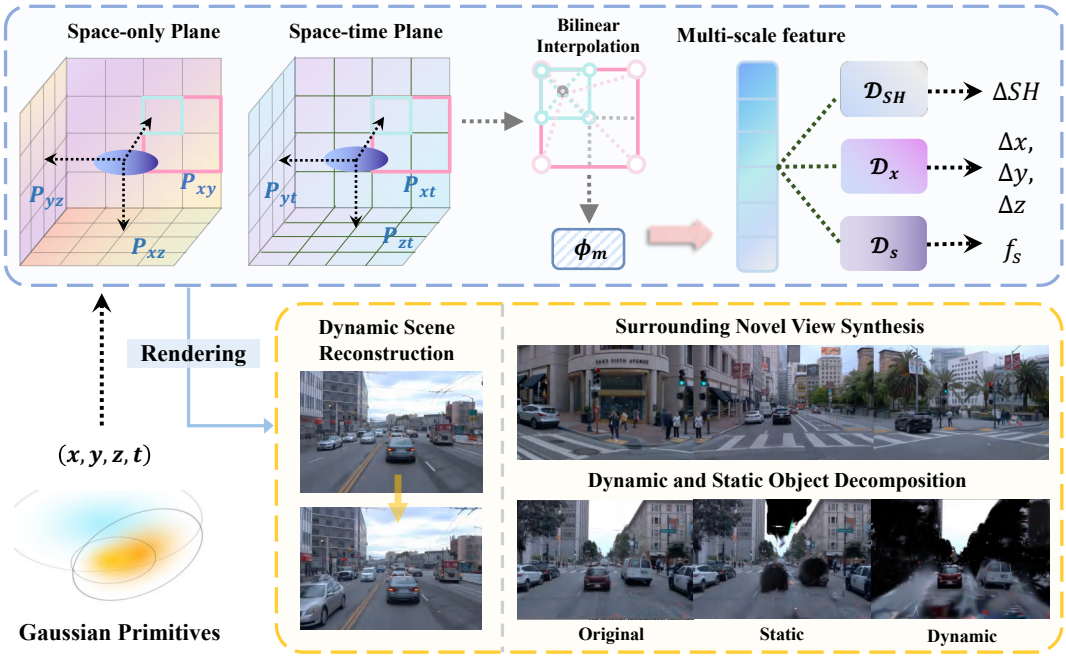

Fig. 2: Pipeline of S^3Gaussian. To tackle the challenges in self-supervised street scene decomposition, our method consists of a Multi-resolution Hexplane Structure Encoder to encode 4D grid into feature planes and a multi-head Gaussian Decoder to decode them into deformed 4D Gaussians. The entire pipeline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

semantic information f_s through the semantic feature decoder $\mathcal{D}_s$. We detail it in Sec. III-B.

B. 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 [11].

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 [42]. Additionally, we incorporate a simple semantic field to assist in automatically decomposing static and dynamic Gaussians.

a) 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 [3], [10], [11], [32]. 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 [26], our hexplane encoder comprises multiple copies of different resolutions. This representation effectively encodes spatial features at various scales. Therefore, our formulation is:

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

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:

$$f(x,y,z,t) = \phi_m\Big(\bigcup_{\rho} \prod \pi(\mathcal{P}_{ij}^{\rho}, \psi_{ij}^{\rho}(x,y,z,t))\Big) \tag{3}$$

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 [11].

b) 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. III-B. 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: $\mathcal{G}' = \{\mathcal{X} + \triangle\mathcal{X}, \mathcal{C} + \triangle\mathcal{C}, s, r, \sigma, f_s\}$.

C. Self-supervised Optimization

a) 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 [30] 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.

b) 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 [46], [53]. The rendered depth is computed using the positions of the Gaussians. $\mathcal{L}_{feat}$ is the L2 loss of semantic feature. Following [10], [33], [11], we 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:

$$\begin{aligned}\mathcal{L} = \lambda_{rgb}\mathcal{L}_{rgb} + \lambda_{depth}\mathcal{L}_{depth} + \lambda_{feat}\mathcal{L}_{feat} \\ + \lambda_{ssim}\mathcal{L}_{ssim} + \lambda_{tv}\mathcal{L}_{tv} + \lambda_{reg}^{x}\mathcal{L}_{reg}^{x} + \lambda_{reg}^{y}\mathcal{L}_{reg}^{c}\end{aligned} \tag{4}$$

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

IV. Experiments

A. Experimental Setup

Datasets. NOTR dataset is a subset of the Waymo Open dataset [34] curated by [46]. In contrast, many public datasets with LiDAR data suffer from a severe imbalance, eg. nuScenes [1] and nuPlan [2], 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 diverse standard for evaluating our static and dynamic reconstruction. Furthermore, we follow StreetGaussian [45]. employ the six scenes selected from the Waymo Open dataset [34], 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 [43] is a modular [36] simulator based on NeRF, utilizing 2D bounding boxes to train NeRF for static and dynamic objects respectively. NSG [27] learns latent codes to model moving objects with a shared decoder. EmerNeRF [46] 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 [18] model employs anisotropic 3D Gaussian ellipsoids as an explicit 3D scene representation, achieving the strongest performance across various tasks in static scenes. StreetGaussian [45], 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 [19], following the learning rate configurations of 3D Gaussians [18]. Additionally, we employ 5,000 steps of pure static 3D Gaussian training [18] as a warm-up for the scene [42]. For the reconstruction of long sequence scenes, we divide the scene into multiple clips, so the varying durations of the scene do not become a factor in motion pattern learning. 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. Our method maintains consistency in most cases by extending the state of the previous clip into the next one. The basic resolution for our multi-resolution HexPlane encoder is set to 64, then upsampled by 2 and 4 as [42]. 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[18]. In the experiments conducted on the Waymo-NOTR dataset, we strictly adhered to the experimental settings of EmerNeRF [46]. Similarly, for the Waymo-Street dataset, our experimental setup closely followed StreetGaussian [45]. All experiments are conducted on NVIDIA Tesla V100 with 22GB GPU memory. And S^3Gaussian costs about 10GB GPU memory for training.

B. Comparisons with the State-of-the-art

The results on the Waymo-NOTR dataset demonstrate that our approach consistently outperforms other methods

Fig. 3: 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.

Fig. 4: Qualitative comparison over Waymo-Street Datasets [45]. All results are from novel view synthesis. Compared to StreetGaussian [45], our method demonstrates a stronger ability to self-supervisedly reconstruct distant dynamic objects and is more sensitive to changes in scene details.

TABLE I: Overall performance of our methods with existing SOTA approaches on the Waymo-NOTR dataset[46]. "PSNR*" and "SSIM*" denote the PSNR and SSIM of dynamic objects respectively. The **best** and the second best results are denoted in bold and underlined.

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 II: Quantitative results on StreetGaussian datasets [45]. We strictly follow the experimental setting.

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	**0.950**
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**

in scene reconstruction and novel view synthesis, as shown in Tab. I. For the static32 dataset, we utilize PSNR, SSIM, and LPIPS [51] as metrics to evaluate rendering quality. LPIPS is a perceptual-based metric to assess visual quality, ensuring the reconstructed scenes align with human perception of dynamic environments. 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

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

TABLE III: 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**

boxes as [46], [45]. 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 Fig. 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 Fig. 1) and reconstruction (right side of Fig. 1) of static and dynamic scenes. Although 3DGS [18] faithfully reconstructs static objects, it fails when dealing with dynamic objects and struggles with reconstructing distant skies. The reconstruction quality of MARS [43] is poor, being effective only for very short sequences, and it struggles to reconstruct fast-moving objects. While EmerNeRF [46] 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. Tab. II presents the results on the dataset collected by StreetGaussian [45]. StreetGaussian is a state-of-the-art method for Gaussian-based dynamic object representation. Our approach performs 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 Fig. 4, compared to StreetGaussian [45] 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.

TABLE IV: Comparison on Dynamic-32 subset of NOTR dataset.

Method	PSNR↑	SSIM↑	LPIPS↓	PSNR*↑	SSIM*↑
4D-GS [48]	30.04	0.898	0.114	24.58	0.731
Ours	**31.35**	**0.911**	**0.106**	**26.02**	**0.783**

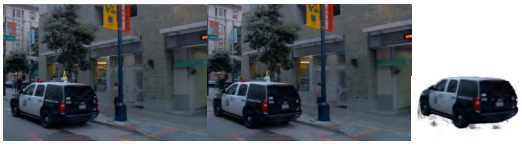

Fig. 6: Left: ground truth, middle: reconstruction, right: decomposition.

C. Instance-level Decomposition

Due to the explicit nature of the Gaussian representation, with the additional semantic feature implemented for each Gaussian, we are able to perform clustering on all the Gaussians and use the clustering results to enable control over individual instances. We conduct primitive experiments for instance-level understanding as shown in Fig 6.

D. Comparisons with 4D-GS

We further include a comparison with 4D-GS in Table IV for the reconstruction task on the Waymo datasets. The results demonstrate our method is clearly different from 4D-GS and performs better reconstruction quality.

E. 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 [46] for analysis, and all models are trained for a shorter duration of $30,000$ iterations. Tab. III presents the quantitative results, while Fig. 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 Tab. III and Fig. 5 demonstrate this, without this module, our rendering quality sharply declines. We also provide visualizations of the features of this encoder, as shown in Fig. 3. As an explicit module, we can easily optimize all Gaussian features on a single voxel plane. From Fig. 3, 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 Fig. 3 and Fig. 2.

Multi-head Gaussian Decoder. Our proposed multi-head Gaussian decoder can decode voxel features. As indicated in Tab. III, disabling this component would impact rendering quality greatly. Additionally, as shown in Fig. 5, disabling the $\mathcal{D}_x$ decoder and only training Gaussian in canonical 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. On the other hand, omitting the semantic feature decoder $\mathcal{D}_s$ and color deformation decoder $\mathcal{D}_{SH}$ primarily affects rendering details. For example, in Fig. 5, the geometric structure of the truck becomes blurrier without these components.

Static Gaussian Warm-up. According to Fig. 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. Additionally, it stabilizes the network by avoiding early-stage numerical errors [42].

V. Conclusion

In this paper, we propose S^3Gaussian, a 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, 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 deform these features into 4D Gaussians. The entire pipeline is optimized without any extra annotations.

Acknowledgement. This work was supported by the National Natural Science Foundation of China (62476011), and by the Beijing Natural Science Foundation (L252060).

References

[1] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020. [Online]. Available: http://dx.doi.org/10.1109/cvpr42600.2020.01164

[2] H. Caesar, J. Kabzan, K. Tan, F. Kit, E. Wolff, A. Lang, L. Fletcher, O. Beijbom, and S. Omari, "nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles." *arXiv: Computer Vision and Pattern Recognition,arXiv: Computer Vision and Pattern Recognition*, Jun 2021.

[3] A. Cao and J. Johnson, "Hexplane: A fast representation for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 130–141.

[4] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "Tensorf: Tensorial radiance fields," in *European Conference on Computer Vision*. Springer, 2022, pp. 333–350.

[5] Y. Chen, C. Gu, J. Jiang, X. Zhu, and L. Zhang, "Periodic vibration gaussian: Dynamic urban scene reconstruction and real-time rendering," *ArXiv*, vol. abs/2311.18561, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:265506369

[6] J. Cheng, Y. Chen, Q. Zhang, L. Gan, C. Liu, and M. Liu, "Real-time trajectory planning for autonomous driving with gaussian process and incremental refinement," in *ICRA*, 2022, pp. 8999–9005.

[7] J. Cheng, X. Mei, and M. Liu, "Forecast-MAE: Self-supervised pre-training for motion forecasting with masked autoencoders," *ICCV*, 2023.

[8] D. Dauner, M. Hallgarten, A. Geiger, and K. Chitta, "Parting with misconceptions about learning-based vehicle motion planning," in *CoRL*, 2023.

[9] A. Dosovitskiy, G. Ros, F. Codevilla, A. M. López, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on Robot Learning*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:5550767

[10] J. Fang, T. Yi, X. Wang, L. Xie, X. Zhang, W. Liu, M. Nießner, and Q. Tian, "Fast dynamic radiance fields with time-aware neural voxels," in *SIGGRAPH Asia 2022 Conference Papers*, 2022, pp. 1–9.

[11] S. Fridovich-Keil, G. Meanti, F. R. Warburg, B. Recht, and A. Kanazawa, "K-planes: Explicit radiance fields in space, time, and appearance," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 479–12 488.

[12] J. Gu, C. Hu, T. Zhang, X. Chen, Y. Wang, Y. Wang, and H. Zhao, "Vip3d: End-to-end visual trajectory prediction via 3d agent queries," *arXiv preprint arXiv:2208.01582*, 2022.

[13] J. Guo, N. Deng, X. Li, Y. Bai, B. Shi, C. Wang, C. Ding, D. Wang, and Y. Li, "Streetsurf: Extending multi-view implicit surface reconstruction to street views," *ArXiv*, vol. abs/2306.04988, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:259108796

[14] A. Hu, Z. Murez, N. Mohan, S. Dudas, J. Hawke, V. Badrinarayanan, R. Cipolla, and A. Kendall, "Fiery: Future instance prediction in bird's-eye view from surround monocular cameras," in *ICCV*, 2021.

[15] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, *et al.*, "Planning-oriented autonomous driving," in *CVPR*, 2023, pp. 17 853–17 862.

[16] Y. Huang, W. Zheng, Y. Zhang, J. Zhou, and J. Lu, "Tri-perspective view for vision-based 3d semantic occupancy prediction," in *CVPR*, 2023, pp. 9223–9232.

[17] B. Jiang, S. Chen, Q. Xu, B. Liao, J. Chen, H. Zhou, Q. Zhang, W. Liu, C. Huang, and X. Wang, "Vad: Vectorized scene representation for efficient autonomous driving," *arXiv preprint arXiv:2303.12077*, 2023.

[18] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, pp. 1–14, 2023.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[20] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Q. Yu, and J. Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *ECCV*, 2022.

[21] Z. Li, Z. Yu, S. Lan, J. Li, J. Kautz, T. Lu, and J. M. Alvarez, "Is ego status all you need for open-loop end-to-end autonomous driving?" in *CVPR*, 2024.

[22] M. Liang, B. Yang, W. Zeng, Y. Chen, R. Hu, S. Casas, and R. Urtasun, "Pnpnet: End-to-end perception and prediction with tracking in the loop," in *CVPR*, 2020.

[23] F. Lu, Y. Xu, G.-S. Chen, H. Li, K.-Y. Lin, and C. Jiang, "Urban radiance field representation with deformable neural mesh primitives," *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 465–476, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:259991347

[24] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[25] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM transactions on graphics (TOG)*, vol. 41, no. 4, pp. 1–15, 2022.

[26] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics*, p. 1–15, Jul 2022. [Online]. Available: http://dx.doi.org/10.1145/3528223.3530127

[27] J. Ost, F. Mannan, N. Thuerey, J. Knodt, and F. Heide, "Neural scene graphs for dynamic scenes," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2855–2864, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:227118710

[28] K. Rematas, A. Liu, P. P. Srinivasan, J. T. Barron, A. Tagliasacchi, T. A. Funkhouser, and V. Ferrari, "Urban radiance fields," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12 922–12 932, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:244714334

[29] V. Rudnev, M. A. Elgharib, W. H. B. Smith, L. Liu, V. Golyanik, and C. Theobalt, "Nerf for outdoor scene relighting," in *European Conference on Computer Vision*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:250921347

[30] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016. [Online]. Available: http://dx.doi.org/10.1109/cvpr.2016.445

[31] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *International Symposium on Field and Service Robotics*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:20999239

[32] R. Shao, Z. Zheng, H. Tu, B. Liu, H. Zhang, and Y. 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*, 2023, pp. 16 632–16 642.

[33] C. Sun, M. Sun, and H.-T. Chen, "Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2022. [Online]. Available: http://dx.doi.org/10.1109/cvpr52688.2022.00538

[34] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, S. Zhao, S. Cheng, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo open dataset," 2020.

[35] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar, "Block-nerf: Scalable large scene neural view synthesis," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8238–8248, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:246706356

[36] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, D. Mcallister, J. Kerr, and A. Kanazawa, "Nerfstudio: A modular framework for neural radiance field development," in *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings*, ser. SIGGRAPH '23. ACM, July 2023. [Online]. Available: http://dx.doi.org/10.1145/3588432.3591516

[37] A. Tonderski, C. Lindstrom, G. Hess, W. Ljungbergh, L. Svensson, and C. Petersson, "Neurad: Neural rendering for autonomous driving," *ArXiv*, vol. abs/2311.15260, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:265457218

[38] H. Turki, D. Ramanan, and M. Satyanarayanan, "Mega-nerf: Scalable construction of large-scale nerfs for virtual fly- throughs," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12 912–12 921, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:245334780

[39] H. Turki, J. Y. Zhang, F. Ferroni, and D. Ramanan, "Suds: Scalable urban dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 375–12 385.

[40] X. Wei, R. Zhang, J. Wu, J. Liu, M. Lu, Y. Guo, and S. Zhang, "Noc: High-quality neural object cloning with 3d lifting of segment anything," *arXiv preprint arXiv:2309.12790*, 2023.

[41] Y. Wei, L. Zhao, W. Zheng, Z. Zhu, J. Zhou, and J. Lu, "Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving," in *ICCV*, 2023, pp. 21 729–21 740.

[42] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang, "4d gaussian splatting for real-time dynamic scene rendering," *ArXiv*, vol. abs/2310.08528, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:263908793

[43] Z. Wu, T. Liu, L. Luo, Z. Zhong, J. Chen, H. Xiao, C. Hou, H. Lou, Y. Chen, R. Yang, Y. Huang, X. Ye, Z. Yan, Y. Shi, Y. Liao, and H. Zhao, "Mars: An instance-aware, modular and realistic simulator for autonomous driving," *CICAI*, 2023.

[44] Z. Xie, J. Zhang, W. Li, F. Zhang, and L. Zhang, "S-nerf: Neural radiance fields for street views," *arXiv preprint arXiv:2303.00749*, 2023.

[45] Y. Yan, H. Lin, C. Zhou, W. Wang, H. Sun, K. Zhan, X. Lang, X. Zhou, and S. Peng, "Street gaussians for modeling dynamic urban scenes," *ArXiv*, vol. abs/2401.01339, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:266725323

[46] J. Yang, B. Ivanovic, O. Litany, X. Weng, S. W. Kim, B. Li, T. Che, D. Xu, S. Fidler, M. Pavone, and Y. Wang, "Emernerf: Emergent spatial-temporal scene decomposition via self-supervision," 2023.

[47] Z. Yang, Y. Chen, J. Wang, S. Manivasagam, W.-C. Ma, A. J. Yang, and R. Urtasun, "Unisim: A neural closed-loop sensor simulator," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1389–1399, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:260438489

[48] Z. Yang, H. Yang, Z. Pan, and L. Zhang, "Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting," *arXiv preprint arXiv:2310.10642*, 2023.

[49] J.-T. Zhai, Z. Feng, J. Du, Y. Mao, J.-J. Liu, Z. Tan, Y. Zhang, X. Ye, and J. Wang, "Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes," *arXiv preprint arXiv:2305.10430*, 2023.

[50] C. Zhang, J. Yan, Y. Wei, J. Li, L. Liu, Y. Tang, Y. Duan, and J. Lu, "Occnerf: Advancing 3d occupancy prediction in lidar-free environments," *IEEE Transactions on Image Processing*, 2025.

[51] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018. [Online]. Available: http://dx.doi.org/10.1109/cvpr.2018.00068

[52] Y. Zhang, Z. Zhu, W. Zheng, J. Huang, G. Huang, J. Zhou, and J. Lu, "Beverse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving," *arXiv preprint arXiv:2205.09743*, 2022.

[53] X. Zhou, Z. Lin, X. Shan, Y. Wang, D. Sun, and M.-H. Yang, "Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes," *ArXiv*, vol. abs/2312.07920, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:266191747