Unified Sensor Simulation for Autonomous Driving

Nikolay Patakin , **Arsenii Shirokov** , **Anton Konushin** and **Dmitry Senushkin**
Lomonosov Moscow State University

Abstract

In this work, we introduce **XSIM**, a sensor simulation framework for autonomous driving. XSIM extends 3DGUT splatting with a generalized rolling-shutter modeling tailored for autonomous driving applications. Our framework provides a unified and flexible formulation for appearance and geometric sensor modeling, enabling rendering of complex sensor distortions in dynamic environments. We identify spherical cameras, such as LiDARs, as a critical edge case for existing 3DGUT splatting due to cyclic projection and time discontinuities at azimuth boundaries leading to incorrect particle projection. To address this issue, we propose a phase modeling mechanism that explicitly accounts temporal and shape discontinuities of Gaussians projected by the Unscented Transform at azimuth borders. In addition, we introduce an extended 3D Gaussian representation that incorporates two distinct opacity parameters to resolve mismatches between geometry and color distributions. As a result, our framework provides enhanced scene representations with improved geometric consistency and photorealistic appearance. We evaluate our framework extensively on multiple autonomous driving datasets, including Waymo Open Dataset, Argoverse 2, and PandaSet. Our framework consistently outperforms strong recent baselines and achieves state-of-the-art performance across all datasets.

1 Introduction

Development of safe and reliable autonomous vehicles necessitates access to large-scale and diverse datasets for both training and evaluation. However, acquiring diverse real-world data remains prohibitively expensive and labor-intensive. Simulation based on real data has emerged as a promising alternative, enabling the creation of augmented datasets in a cost and time efficient manner for various downstream applications [Yuan *et al.*, 2024; Adamkiewicz *et al.*, 2022; Ljungbergh *et al.*, 2025]. Recently, 3D Gaussian Splatting (3DGS) [Kerbl *et al.*, 2023; Wu *et al.*, 2025; Hess *et al.*, 2024] has introduced a photorealistic rendering engine lowering sim-to-real gap and offering an effective balance among visual fidelity and computational efficiency.

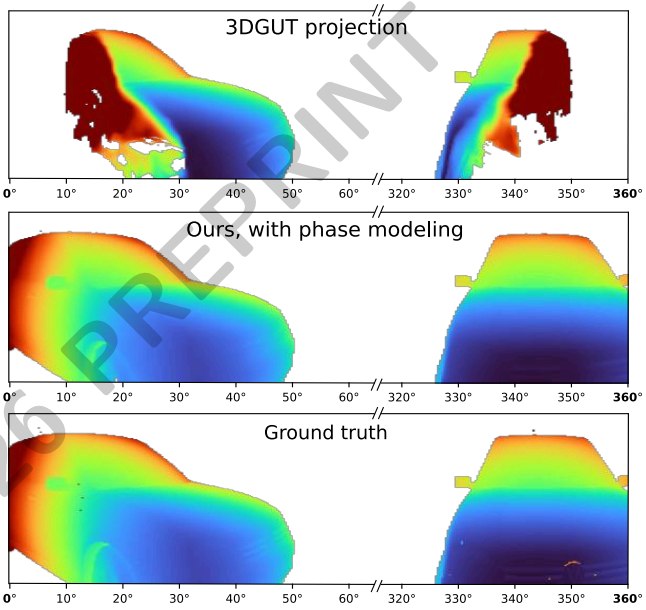

Figure 1. Synthetic example of LiDAR rendering. In regions near the azimuth discontinuity border, the standard 3DGUT projection results in partially missing and distorted range image renders. Our phase modeling approach alleviates this issue and also effectively handles surfaces observed twice due to the rolling shutter.

While existing simulators predominantly rely on EWA splatting [Zwicker *et al.*, 2001], the recently proposed 3DGUT framework [Wu *et al.*, 2025] offers benefits for sensor simulation in autonomous driving scenarios. Real-world driving data is typically captured using cameras with highly nonlinear and dynamic effects, such as rolling shutter and optical distortions. These effects are difficult to model accurately with EWA splatting due to its reliance on linearized particle projection during rasterization, resulting in specialized rendering procedures for each camera type. In contrast, 3DGUT projects Gaussians through arbitrary camera models using the Unscented Transform (UT), effectively emulating ray-based rendering within a rasterization framework and enabling accurate modeling of complex distortions. Building

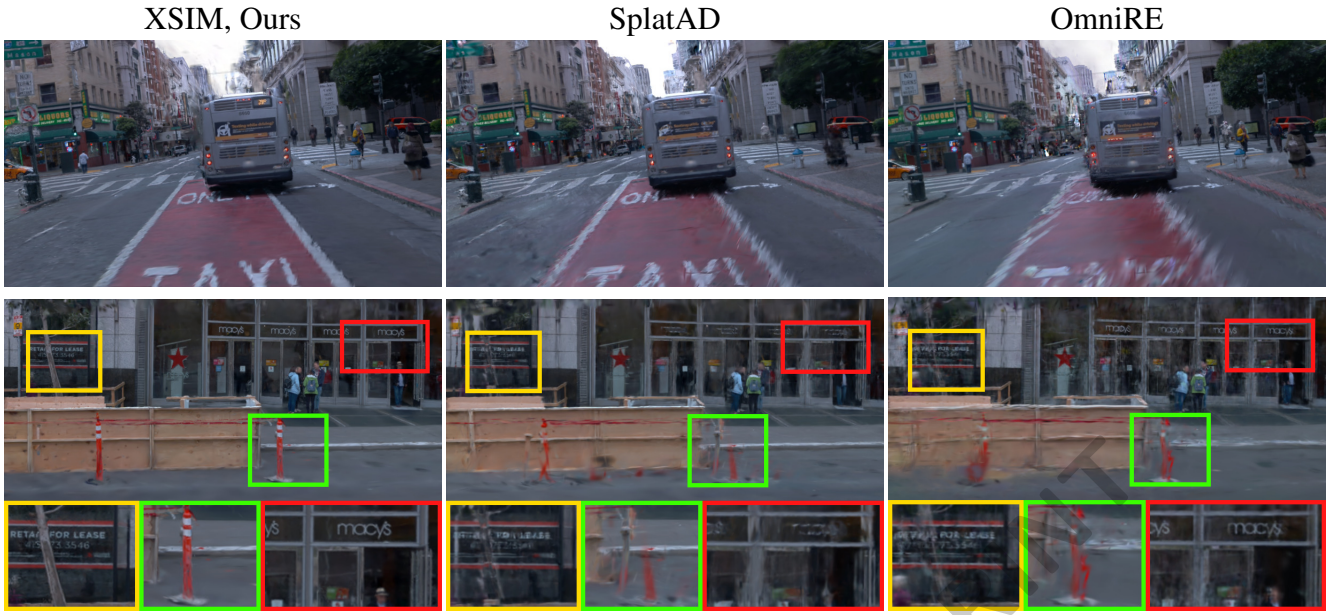

Figure 2. Novel view synthesis (Lane shift 3m). Qualitative comparison on Waymo Open Dataset demonstrates that XSIM provides scene representation which can be consistently rendered from novel ego-vehicle trajectories.

on this capability, we introduce a sensor simulation framework that adapts 3DGUT splatting to autonomous driving data. We further extend the framework with a generalized rolling-shutter modeling, enabling accurate simulation of LiDAR and arbitrary camera sensors in unified manner. However, we observe that UT-based projection for spherical cameras introduces challenges for Gaussian particles spanning the azimuth boundary. Naive processing of such particles leads to missing projections or distorted positions and shapes due to cyclic azimuth parameterization and temporal discontinuities at azimuth edges. To address this issue, we propose a phase modeling mechanism that explicitly accounts for these effects, enabling accurate rendering of spherical cameras with rolling shutter.

Autonomous driving scenarios require jointly modeling geometric and appearance sensors. Accurate appearance modeling typically necessitates multiple transparent Gaussian particles to capture complex lighting effects, whereas geometric sensors provide precise surface measurements that can be faithfully represented using a single opaque Gaussian. To address this distribution mismatch, we extend scene parameterization with two distinct opacity parameters per Gaussian, jointly optimized for color and geometry distributions. Such parameterization alleviates potential mismatches within a unified representation and results in enhanced quality.

Eventually, our main contributions are three-fold:

- We introduce XSIM – the sensor simulation framework for autonomous driving extending 3DGUT splatting and enabling rendering LiDAR and camera sensors in a unified manner with generalized rolling shutter modeling.
- We propose a phase modeling mechanism for spherical rasterization that explicitly accounts temporal and shape discontinuities of Gaussians projected by the Unscented Transform at azimuth borders.
- We introduce an extended 3D Gaussian representation that incorporates two distinct opacity parameters per Gaussian, jointly optimized for color and geometry distributions, mitigating distribution mismatches within a unified representation.

2 Related Work

3DGS and NeRF for automotive scenes. In automotive applications Neural Radiance Field (NeRF) [Mildenhall *et al.*, 2020; Barron *et al.*, 2023; Müller *et al.*, 2022; Wu *et al.*, 2023; Xie *et al.*, 2023] and 3D Gaussian Splatting (3DGS) [Kerbl *et al.*, 2023; Wu *et al.*, 2025; Hess *et al.*, 2024; Zhang *et al.*, 2024] gathered significant attention since it provide flexible tools for photorealistic sensor modeling. Initial research efforts [Ost *et al.*, 2021; Fu *et al.*, 2022; Kundu *et al.*, 2022; Kerbl *et al.*, 2023; Moenne-Loccoz *et al.*, 2024] emphasized the reconstruction of scene semantic and appearance guided by RGB camera images. Later works enriched scene representation by decoupled modeling of dynamic actors and static layout leveraging either explicit scene graphs [Yang *et al.*, 2023; Tonderski *et al.*, 2024; Yang *et al.*, 2024; Hess *et al.*, 2024; Zhou *et al.*, 2024c; Yan *et al.*, 2024; Zhou *et al.*, 2024b; Zhou *et al.*, 2024a; Jiang *et al.*, 2025] or time-dependent implicit representations [Go *et al.*, 2025; Chen *et al.*, 2023; Peng *et al.*, 2025]. Recent approaches further extended prior methods to encompass accurate sensor simulations including modeling of complex camera distortions [Xie *et al.*, 2023; Yang *et al.*, 2023; Tonderski *et al.*, 2024; Chen *et al.*, 2025a; Zhang *et al.*, 2024; Wu *et al.*, 2025; Moenne-Loccoz *et al.*, 2024] and LiDAR [Huang *et al.*, 2023; Wu *et al.*, 2024; Yang *et al.*, 2023;

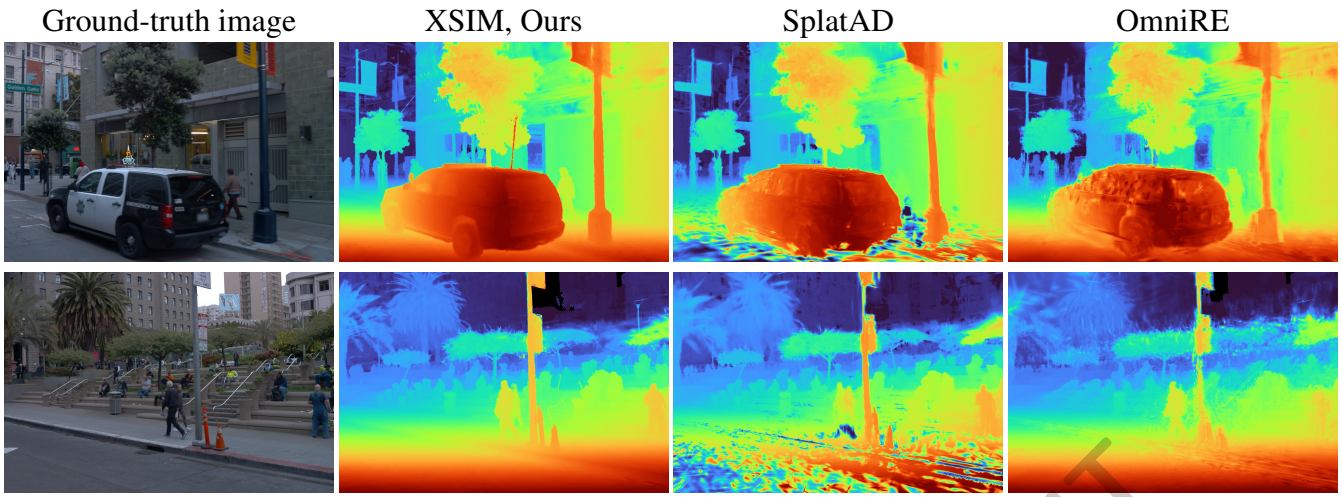

Figure 3. Depth map rendering. Qualitative comparison of depth map rendering on Waymo Open Dataset. Compared to previous methods, our framework provides smooth geometric representations with high level of details.

Tonderski *et al.*, 2024; Chen *et al.*, 2025a; Hess *et al.*, 2024; Zhou *et al.*, 2025; Chen *et al.*, 2025b]. 3DGS provides a better trade-off between photorealism, physical accuracy and computational efficiency. However, the conventional 3DGS based on EWA [Zwicker *et al.*, 2001] formulation introduces challenges for automotive scenarios limiting the ability to render accurately complex sensors [Huang *et al.*, 2025]. The promising alternative 3DGUT [Wu *et al.*, 2025] provides flexibility to render arbitrary cameras without strong approximation. In this work we introduced the sensor simulation framework based on 3DGUT [Wu *et al.*, 2025]. In contrast to previous 3DGS-based methods [Hess *et al.*, 2024; Chen *et al.*, 2025b], our framework provides a unified rendering formulation for rolling-shutter camera and LiDAR sensors, resulting in improved cross-sensor consistency.

3 Method

We introduce XSIM – the sensor simulation framework for autonomous driving. Our framework is based on 3DGUT formulation that we overview in Section 3.1. In Section 3.2 we introduce extended 3D Gaussian representation to alleviate geometry and color distribution mismatches. Finally, we describe our rolling shutter sensor modeling approach along with the proposed phase mode mechanism in Section 3.3.

3.1 Gaussian Splatting Preliminary

Scene representation 3DGS [Kerbl *et al.*, 2023; Zwicker *et al.*, 2001; Wu *et al.*, 2025] represents an arbitrary scene as a set of transparent Gaussian particles. Each particle is parameterized by its mean position $\boldsymbol{\mu} \in \mathbb{R}^3$ and a shape encoded by a covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{3\times3}$. The contribution of a particle is defined by a Gaussian kernel function:

$$\rho(\boldsymbol{x}) = \sigma \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) \right) \quad (1)$$

Covariances $\boldsymbol{\Sigma} = \boldsymbol{R}\boldsymbol{S}\boldsymbol{S}^\top\boldsymbol{R}^\top$ in practice are represented and optimized via decoupled scaling vector $\boldsymbol{s} \in \mathbb{R}^3$ and rotation quaternion $\boldsymbol{q} \in \mathbb{R}^4$. Each particle is associated with opacity value $\sigma \in \mathbb{R}$, diffuse color $c_d \in \mathbb{R}^3$, and view-dependent appearance feature vector $c_s \in \mathbb{R}^f$. While original 3DGS [Kerbl *et al.*, 2023] uses spherical harmonics for encoding view-dependent appearance and evaluate them into color before volumetric integration, recent autonomous driving simulation frameworks [Tonderski *et al.*, 2024; Hess *et al.*, 2024] render both c_d and c_s features into an image. We follow their approach, and use small trainable network consisting of few convolution layers to decode RGB color.

Dynamic scenes are typically represented as the graph [Chen *et al.*, 2025b] with static and dynamic actor nodes, consisting of individual Gaussian particles. Dynamic actors are associated with their bounding boxes and an optimizable trajectory (sequence of SE3 poses). As pedestrians and cyclists represent a vulnerable road users category, we follow [Chen *et al.*, 2025b] and represent humans as SMPL [Loper *et al.*, 2015] bodies to improve reconstruction quality and provide better controllability.

Unscented transform. A key stage that enables efficient rendering in Gaussian splatting is tiling, in which particles are assigned to screen-space tiles based on the 2D projection of their shape. The EWA splatting formulation relies on a linearized projection approximation based on a single point, which makes rendering highly non-linear cameras (e.g. with rolling shutter or optical distortions) challenging. Alternatively, 3DGUT projects particles onto an arbitrary camera via Unscented Transform (UT). Given the mean and covariance of a particle in 3D, UT constructs a set of sigma points, which are individually projected through the camera model. The resulting 2D Gaussian conics are then approximated by weighting projected points. As a result, UT enables tiling for a wide range of complex camera models without requiring any modifications. Finally, particles assigned to each tile are sorted by depth to ensure correct ordering during volumetric rendering.

Volumetric integration. The final stage of rendering is rasterization, which performs volumetric integration. In EWA

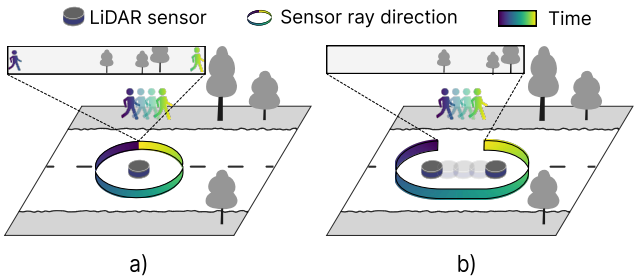

Figure 4. LiDAR discontinuities occurring near azimuth border. a) Even when sensor is stationary and covers exactly 360 degrees, time discontinuity due to the rolling shutter may lead to objects observed twice. b) Ego movement in combination with rolling shutter leads to spatial discontinuities

splatting, 2D Gaussian conics are used explicitly to compute particle responses during volumetric rendering, introducing projection approximation errors into the rendering process. In contrast, 3DGUT uses 2D conics only for tiling, while volumetric rendering is performed by computing particle responses directly in 3D space. Specifically, for a given camera ray with origin $\boldsymbol{o} \in \mathbb{R}^3$ and direction $\boldsymbol{d} \in \mathbb{R}^3$, the particle is evaluated at the point of maximum response along the ray:

$$\boldsymbol{x}_{\text{max}} = \boldsymbol{o} + \tau_{\text{max}}\boldsymbol{d}, \quad \tau_{\text{max}} = \frac{\boldsymbol{d}^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - \boldsymbol{o})}{\boldsymbol{d}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{d}} \tag{2}$$

Given the maximum response point, the overall volumetric integration follows standard formulation with $\alpha_i = \rho_i(\boldsymbol{x}_{\text{max}})$:

$$T_i = \prod_{j<i}(1-\alpha_j), \quad \boldsymbol{c} = \sum_i \boldsymbol{c}_i \alpha_i T_i \tag{3}$$

Similarly, for depth and range image rendering we use the same equation as above, but replace particle colors $\boldsymbol{c}_i$ with maximum response τ_{max} distances along the ray.

3.2 Extended 3D Gaussian Representation

While using geometric guidance is generally beneficial for 3DGS scene reconstuction, in some cases the two sensing modalities may impose different requirements on opacity modeling. Accurate geometric modelling requires representing surfaces with precisely located opaque gaussians, as guided by LiDAR measurements. In contrast, appearance modelling must account for view-dependent effects such as specular reflections and translucency, which often require multiple semi-transparent Gaussians along a viewing ray. Furthermore, surface opacity can be wavelength-dependent, as materials such as glass exhibit different transparency properties for LiDAR and visible light. To accommodate these effects, we augment each Gaussian with separate opacity parameters: σ_c for camera and σ_L for LiDAR rendering, which are jointly optimized within the unified representation. These two opacity values are then regularized during optimization to ensure consistency. We experimentally demonstrate that this extended representation effectively resolves distribution mismatches and improves both camera and LiDAR modelling for the novel-view synthesis problem.

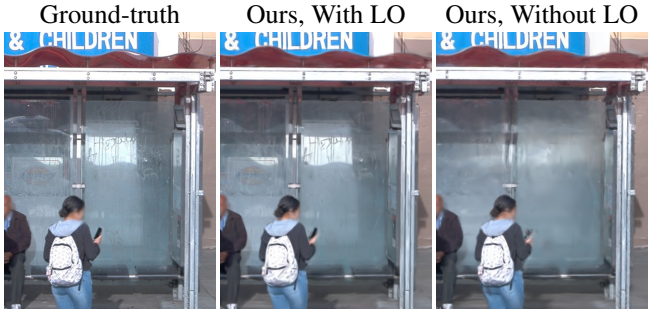

Figure 5. Modeling LiDAR opacity (LO) separately resolves geometry and color distributions mismatch, and increases quality of appearance modeling for translucent surfaces and specular reflections.

3.3 Sensor Modeling

Autonomous driving perception and planning algorithms primarily rely on two sensor modalities: cameras and LiDARs. Both sensors typically operate in rolling shutter mode, in which sensor readings are acquired sequentially over time in a row-by-row fashion. As highlited by multiple previous works [Tonderski *et al.*, 2024; Hess *et al.*, 2024], due to high velocities experienced in autonomous driving scenarios, modeling rolling shutter is essential for accurate reconstruction. Whereas previous work [Hess *et al.*, 2024] addressed rolling-shutter effects using separate, sensor-specific models, we describe generalized rolling-shutter modeling approach. We also identify spherical cameras (i.e. LiDARs) as an edge case of 3DGUT approach and propose phase modeling mechanism to mitigate arising rendering issues.

General rolling shutter modeling. Due to the rolling-shutter mechanism, sensor readings are not captured instantaneously, and the observation time varies across pixels. Consequently, each image-space point (u, v) is associated with a distinct capture time $\tau(u, v)$. While in practice the rolling shutter time is linear and aligned with either horizontal or vertical image axis, we can express it as:

$$\tau(u, v) = \tau_{\text{start}} + u\tau_u + v\tau_v \tag{4}$$

where τ_u, τ_v define the scan direction and speed, and denote middle of exposure time as $\tau_{\text{mid}} = \tau(0.5, 0.5)$.

As time progresses during image acquisition, both the sensor and the scene evolve. We assume that, over the duration of capture, the camera and all dynamic actors move with constant linear and angular velocities. Under this assumption, the position of a 3D world point $\boldsymbol{x}_w \in \mathbb{R}^3$ at time η is given by:

$$\boldsymbol{x}_w(\eta) = \boldsymbol{x}_w(\tau_{\text{mid}}) + (\boldsymbol{v}_a + \boldsymbol{w}_a \times \boldsymbol{r})\eta \tag{5}$$

where $\boldsymbol{v}_a \in \mathbb{R}^3$ and $\boldsymbol{w}_a \in \mathbb{R}^3$ denote linear and angular velocities of the dynamic actor, respectively, and $\boldsymbol{r} \in \mathbb{R}^3$ is radius vector defined by point position in actor coordinates.

Similarly, the camera pose evolves over time according to constant-velocity motion. Let $\boldsymbol{q}(\eta) \in \mathbb{R}^4$ and $\boldsymbol{t}(\eta) \in \mathbb{R}^3$ denote the camera orientation and position at time η. Camera movement is then modeled as:

$$\boldsymbol{q}(\eta) = e^{\boldsymbol{w}_c \eta/2} \otimes \boldsymbol{q}(\tau_{\text{mid}}), \quad \boldsymbol{t}(\eta) = \boldsymbol{t}(\tau_{\text{mid}}) + \eta \boldsymbol{v}_c \tag{6}$$

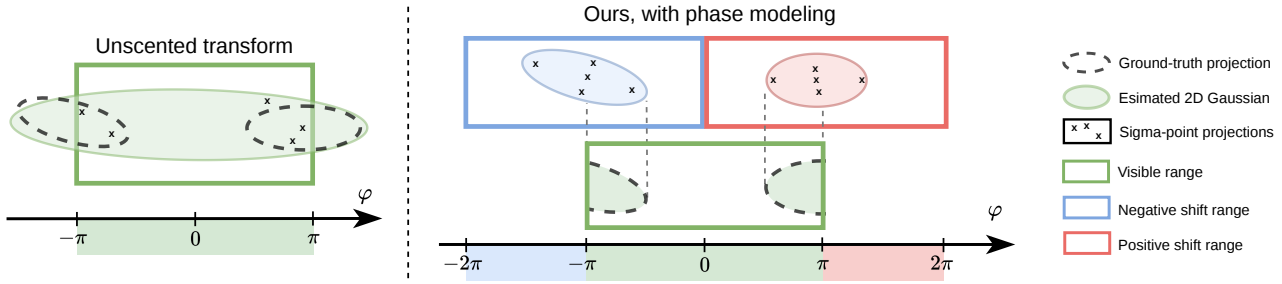

Figure 6. Single Gaussian particle spanning the azimuth boundaries ($\varphi = \pm\pi$) of rolling-shutter spherical camera projects into two separate 2D gaussians with different 2D covariances. Unscented Transform provides unimodal posterior approximation of particle projection, leading to a overly large projections with incorrect shapes. Our phase modeling mechanism enables bimodal 2D gaussian projection by considering two additional projections shifted by $\pm\pi$ from visible range. By performing projections shifted by half period, we handle particles which sigma points fall into multiple azimuth periods.

where $\boldsymbol{v}_c$ and $\boldsymbol{w}_c$ are the camera's linear and angular velocities, and $\otimes$ denotes the Hamilton product. Given the arbitrary static camera projection function $\pi(\cdot) : \mathbb{R}^3 \to [0,1]^2$ and known point time η projection of world point $x_w \in \mathbb{R}^3$ into image coordinates (u, v) has the closed form:

$$\begin{aligned} \boldsymbol{x}_c(\eta) &= \boldsymbol{q}^{-1}(\eta) \otimes (\boldsymbol{x}_w(\eta) - \boldsymbol{t}(\eta)) \otimes \boldsymbol{q}(\eta), \\ u(\eta), v(\eta) &= \pi(\boldsymbol{x}_c(\eta)), \end{aligned} \tag{7}$$

However, since the observation time η of the point is typically not known prior to projection, the projection problem formulates as estimating world time η which is consistent with image space point observation time:

$$\eta = \tau(u(\eta), v(\eta)) \tag{8}$$

Although this dependency is nonlinear and does not admit a closed-form solution, it can be efficiently solved using iterative methods. Following [Sun *et al.*, 2020], we use the Newton–Raphson method and observe that only a few iterations are sufficient in practice.

Phase modeling. Our framework builds upon 3DGUT splatting, which leverages the Unscented Transform (UT) to approximate the projection of a 3D Gaussian. The UT performs a posterior approximation under the assumption that the projection of a 3D Gaussian can be adequately represented by a single 2D Gaussian. While this assumption holds in many practical settings, it breaks down in scenarios involving spherical rolling-shutter sensors. In particular, spinning spherical LiDAR sensors may observe a single Gaussian particle multiple times when it spans the azimuthal boundary or ignore it depending on the world dynamics (see Fig. 4). Moreover, this can lead to a same particle projecting into two separate shapes near boundaries with different covariances and depths (see ground-truth projections in Fig. 6). In such cases, the projected distribution becomes multimodal, violating the assumptions inherent to the UT.

To address this limitation, we introduce a phase modeling mechanism for spherical camera rendering. Under the spherical projection, the azimuth is a periodic function defined as:

$$\phi = \operatorname{atan2}(y, x) + 2\pi k, \quad k \in \mathbb{Z} \tag{9}$$

While conventional static spherical mappings retain only the principal solution ($k = 0$), we explicitly account for phase wrapping by considering $k \in \{-1, 0, +1\}$. Since additional projections may differ in covariance and depth to camera, in addition to the standard central projection, we perform auxiliary projections shifted by $\pm\pi$ (see Fig. 6). For each interval, we constrain projections of sigma points into it by initializing τ in Eq. 8 with an interval middle of exposure time, and shift projected point azimuths by $2\pi k$ if they fall outside of interval. 2D Gaussian conics and depth are individually estimated for each interval based on projected sigma points, and valid projections are passed to the next tiling stage. Compared to original 3DGUT projection, our mechanism results in accurate tiling with no false tile-particle intersections and less depth sorting errors for particles near azimuth boundaries.

3.4 Camera and LiDAR supervision

Our scene representation consisting of multiple object nodes is optimized simultaneously from driving logs by randomly sampling images and closest by time LiDAR sweeps at each iteration. We supervise it using combination of losses:

$$\mathcal{L} = \underbrace{\lambda\mathcal{L}_1 + (1-\lambda)L_{\text{SSIM}}}_{\text{camera guidance}} + \underbrace{\mathcal{L}_{\text{depth}}}_{\text{LiDAR}} + \mathcal{L}_{\text{opacity}} + \mathcal{L}_{reg} \tag{10}$$

Following common practice we set $\lambda = 0.2$. LiDAR guidance loss is $\mathcal{L}_1$ loss between rendered and ground-truth ray lengths. To ensure consistency between optimized opacities we impose $L_{\text{opacity}} = \sum_i |\sigma_{c,i} - \sigma_{L,i}|$ regularization. Details on other regularization terms $\mathcal{L}_{reg}$ are listed in appendix.

4 Experiments

Implementation details. We implement camera and LiDAR rendering using custom CUDA kernels. A unified rendering pipeline allows sharing rasterization forward and backward passes across all camera models. To handle non-uniform LiDAR beam angles during tiling, particle assignment iterates over elevation tile boundaries [Hess *et al.*, 2024]. Human modeling follows OmniRE [Chen *et al.*, 2025b], using their deformable and SMPL-based scene nodes. We adopt the specular Gaussian configuration and CNN post-processing from [Hess *et al.*, 2024; Tonderski *et*

Dataset	Method	Conference	Reconstruction				Novel-view synthesis			
			PSNR↑	SSIM↑	LPIPS↓	CD↓	PSNR↑	SSIM↑	LPIPS↓	CD↓
Waymo (12 scenes)	PVG	Arxiv23	25,02	0,8005	0,4408	82,11	24,29	0,7864	0,4451	68,27
	StreetGS	ECCV24	25,45	0,8123	0,3111	16,16	24,41	0,7827	0,3198	16,87
	OmniRE	ICLR25	25,94	0,8159	0,3049	15,68	24,60	0,7765	0,3207	13,76
	HUGS	CVPR24	26,90	0,8513	0,3351	44,58	25,95	0,8267	0,3337	37,60
	EmerNerf	ICLR24	27,15	0,8056	0,4620	0,71	26,12	0,7962	0,4575	2,54
	SplatAD	CVPR25	27,74	0,8650	0,2807	0,82	27,06	0,8492	0,2807	0,82
	XSIM, Ours	**–**	**30,75**	**0,9030**	**0,2228**	**0,08**	**29,80**	**0,8904**	**0,2236**	**0,18**
Argoverse (10 scenes)	PVG	Arxiv23	23,78	0,7164	0,4840	31,36	22,93	0,7031	0,4908	27,20
	StreetGS	ECCV24	23,85	0,7223	0,3824	21,62	22,37	0,6975	0,3806	21,01
	OmniRE	ICLR25	23,97	0,7230	0,3822	21,87	22,44	0,6975	0,3815	22,02
	UniSim	CVPR23	23,04	0,6697	0,3986	25,58	23,06	0,6694	0,3962	26,24
	NeuRad	CVPR24	26,46	0,7271	0,3045	2,43	26,49	0,7271	0,3044	2,63
	SplatAD	CVPR25	28,71	0,8333	0,2653	2,78	28,40	0,8258	0,2706	2,68
	XSIM, Ours	**–**	**29,44**	**0,8431**	**0,2563**	**0,57**	**29,44**	**0,8423**	**0,2514**	**1,26**
Pandaset (10 scenes)	PVG	Arxiv23	23,62	0,7066	0,4405	101,33	22,81	0,6885	0,4537	121,54
	StreetGS	ECCV24	23,70	0,7192	0,3206	18,68	22,53	0,6866	0,3249	19,90
	OmniRE	ICLR25	23,73	0,7196	0,3246	21,02	22,58	0,6884	0,3262	18,99
	UniSim	CVPR23	23,62	0,6953	0,3291	10,46	23,45	0,6910	0,3300	9,68
	NeuRad	CVPR24	26,54	0,7675	0,2386	1,45	26,05	0,7589	0,2418	1,65
	SplatAD	CVPR25	28,69	0,8759	**0,1853**	1,54	26,77	0,8044	**0,1904**	1,69
	XSIM, Ours	**–**	**29,05**	**0,8839**	0,1872	**0,20**	**27,00**	**0,8055**	0,1944	**1,23**

Table 1. Quantitative results on three datasets under scene reconstruction and novel-view synthesis scenarios. We report RGB image quality metrics (PSNR, SSIM, LPIPS) and LiDAR reconstruction accuracy measured by Chamfer Distance (CD). Our framework achieves state-of-the-art performance across all datasets and scenarios, with particularly large error reductions for LiDAR rendering. On PandaSet, LPIPS remains competitive and ranks second, with a minor gap to the best-performing method.

al., 2024]. Hyperparameters largely follow SplatAD [Hess *et al.*, 2024], while Gaussian splitting and densification use the 3DGUT [Wu *et al.*, 2025] strategy with minor modifications. Full hyperparameter details are provided in the appendix.

Datasets. To evaluate our framework we use three popular datasets – Waymo Open Dataset [Sun *et al.*, 2020], Argoverse2 [Wilson *et al.*, 2021] and PandaSet [Xiao *et al.*, 2021]. Following OmniRE [Chen *et al.*, 2025b], we use 12 scenes from Waymo, featuring ego-vehicle motion, dynamic and diverse classes of vehicles and pedestrians. As for Argoverse2 and PandaSet we adopt SplatAD [Hess *et al.*, 2024] partition without modifications. Both training and evaluation are performed using full-resolution images and point clouds.

Baselines. For experimental evaluation we choose a wide range of baselines, featuring both recent NeRF-based methods (UniSim [Yang *et al.*, 2023], NeuRAD [Tonderski *et al.*, 2024], EmerNerf [Yang *et al.*, 2024]) and 3DGS-based PVG [Chen *et al.*, 2023], StreetGaussians [Yan *et al.*, 2024], OmniRe [Chen *et al.*, 2025b], HUGS [Zhou *et al.*, 2024b; Zhou *et al.*, 2024a] and SplatAD [Hess *et al.*, 2024]. For PVG, StreetGaussians and OmniRe we use implementation based on `drivestudio`. As UniSim has no official repository, we use reimplementation by `neurad-studio`.

Evaluation metrics. For rendered image quality assessment we use standard novel-view synthesis metrics – PSNR ↑, SSIM ↑, and LPIPS↓. To measure LiDAR simulation quality we compute Chamfer Distance (CD↓) metric between rendered and ground-truth point clouds.

4.1 Image rendering

Scene Reconstruction. Following common practice, we evaluate the upper bound of reconstruction and modeling capacity of frameworks by reconstructing scenes using all available RGB images and LiDAR sweeps. Quantitative results on three datasets are reported in Table 1. On the Waymo dataset, XSIM achieves substantial improvements over the previous state of the art, SplatAD, with gains of +3.01 PSNR and +3.8% SSIM, while reducing LPIPS by 20.6%. Across all three datasets, XSIM consistently demonstrates superior RGB image reconstruction quality compared to prior baselines. On PandaSet, performance remains competitive across all metrics, with LPIPS showing a minor performance drop.

Novel-view synthesis. As our framework targets simulation of novel scenarios and trajectories, we evaluate render-

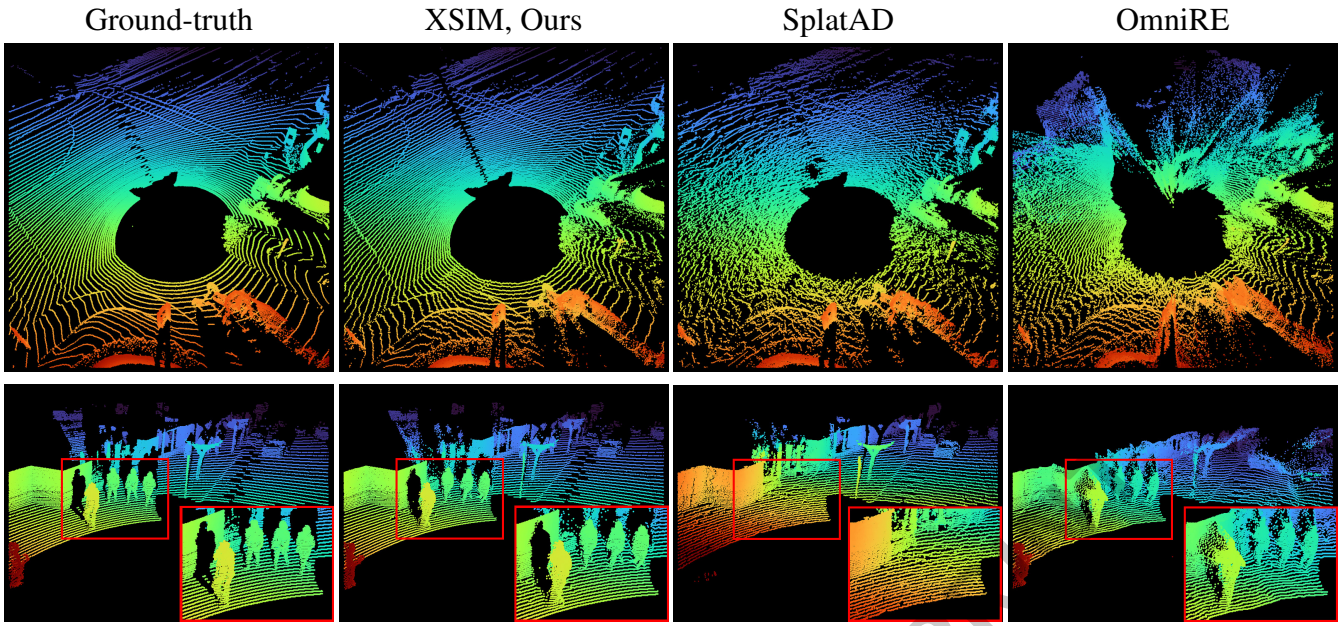

Figure 7. LiDAR Rendering. Comparison of LiDAR point clouds rendered by different methods. Previous approaches exhibit distorted scan-line patterns and incomplete geometry, including vulnerable road users such as pedestrians.

ing quality on views unseen during reconstruction (Table 1). We train on every second sensor frame and evaluate on the remaining frames. In this setting, XSIM achieves gains of +2.74 PSNR on Waymo and +1.04 PSNR on Argoverse over previous state-of-the-art method. We further assess extrapolation capability in Figure 2 by rendering ego-vehicle cameras under a lateral shift of 3 meters. Compared to prior methods, our framework produces more consistent renderings.

4.2 LiDAR and depth rendering

We evaluate LiDAR rendering quality on three datasets in both scene reconstruction and novel-view synthesis settings using Chamfer Distance (CD) (Table 1). XSIM achieves state-of-the-art results on all datasets, with an x8.8 CD error reduction on Waymo reconstruction and an x4.5 error reduction on Waymo NVS. As shown in Figure 7, XSIM better preserves characteristic LiDAR ring patterns and accurately renders pedestrians. We further assess geometric quality via RGB-camera depth rendering (Figure 3), where our method produces smooth, dense depth maps.

4.3 Ablation study

We illustrate the effect of our phase modeling mechanism via synthetic example (fig. 1) with a single car mesh captured by rolling shutter LiDAR camera. As the LiDAR moves along the vehicle, it becomes visible twice near the azimuthal boundaries of the range image due to rolling shutter. While 3DGUT default projection produces artifacts at the boundaries, our framework with phase modeling precisely reconstructs and renders the scene. We further ablate our framework components in a novel-view synthesis setting on six scenes from Waymo dataset by individually disabling features (Tab. 2). Camera rolling-shutter modeling (`c` vs. `a`) leads to significant gains in PSNR. Removing the separate LiDAR opacity parameter (`c` vs. `a`) degrades both image and LiDAR metrics, with qualitative effects shown in Figure 5. Modeling LiDAR rolling shutter (`d` vs. `c`) with our phase modeling mechanism (`e` vs. `c`) further improves image and LiDAR rendering demonstrating effectiveness of our contributions.

Component	PSNR↑	SSIM↑	LPIPS↓	CD↓
(a) XSIM, full	**30,03**	**0,8945**	**0,2122**	**0,21**
(b) – Camera rolling shutter	28,95	0,8761	0,2407	0,21
(c) – Lidar opacity	29,55	0,8888	0,2189	0,25
(d) – Lidar rolling shutter	29,05	0,8816	0,2296	0,31
(e) – Phase modelling	29,32	0,8824	0,2245	0,28

Table 2. Ablation studies on novel-view synthesis on Waymo dataset (half of split). Each row corresponds to disabling of a single component relative to the parent configuration.

5 Conclusion

In this paper, we presented XSIM, a unified sensor simulation framework for autonomous driving that extends 3DGUT splatting with generalized rolling-shutter modeling, enabling modeling of complex sensors in dynamic environments in a unified manner. Our phase modeling mechanism enables robust rendering for spherical rolling-shutter sensors by accounting for azimuthal discontinuities. Extensive experiments on three autonomous driving benchmarks demonstrate that our approach consistently improves geometric accuracy and photorealism, outperforming strong previous baselines.

Acknowledgements

This work was supported by the The Ministry of Economic Development of the Russian Federation in accordance with the subsidy agreement (agreement identifier 000000C313925P4H0002; grant No 139-15-2025-012).

References

[Adamkiewicz *et al.*, 2022] Michal Adamkiewicz, Timothy Chen, Adam Caccavale, Rachel Gardner, Preston Culbertson, Jeannette Bohg, and Mac Schwager. Vision-Only Robot Navigation in a Neural Radiance World. *IEEE Robotics and Automation Letters (RA-L)*, 7(2):4606–4613, April 2022. website: https://mikh3x4.github.io/nerf-navigation/.

[Barron *et al.*, 2023] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *ICCV*, 2023.

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

[Chen *et al.*, 2025a] Yun Chen, Matthew Haines, Jingkang Wang, Krzysztof Baron-Lis, Sivabalan Manivasagam, Ze Yang, and Raquel Urtasun. Salf: Sparse local fields for multi-sensor rendering in real-time. *arXiv preprint arxiv:2507.18713*, 2025.

[Chen *et al.*, 2025b] Ziyu Chen, Jiawei Yang, Jiahui Huang, Riccardo de Lutio, Janick Martinez Esturo, Boris Ivanovic, Or Litany, Zan Gojcic, Sanja Fidler, Marco Pavone, Li Song, and Yue Wang. Omnire: Omni urban scene reconstruction. In *The Thirteenth International Conference on Learning Representations*, 2025.

[Fu *et al.*, 2022] Xiao Fu, Shangzhan Zhang, Tianrun Chen, Yichong Lu, Lanyun Zhu, Xiaowei Zhou, Andreas Geiger, and Yiyi Liao. Panoptic nerf: 3d-to-2d label transfer for panoptic urban scene segmentation. In *International Conference on 3D Vision (3DV)*, 2022.

[Go *et al.*, 2025] Hyojun Go, Byeongjun Park, Jiho Jang, Jin-Young Kim, Soonwoo Kwon, and Changick Kim. Splatflow: Multi-view rectified flow model for 3d gaussian splatting synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21524–21536, June 2025.

[Hess *et al.*, 2024] Georg Hess, Carl Lindström, Maryam Fatemi, Christoffer Petersson, and Lennart Svensson. Splatad: Real-time lidar and camera rendering with 3d gaussian splatting for autonomous driving. *arXiv preprint arXiv:2411.16816*, 2024.

[Huang *et al.*, 2023] Shengyu Huang, Zan Gojcic, Zian Wang, Francis Williams, Yoni Kasten, Sanja Fidler, Konrad Schindler, and Or Litany. Neural lidar fields for novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 18236–18246, October 2023.

[Huang *et al.*, 2025] Letian Huang, Jiayang Bai, Jie Guo, Yuanqi Li, and Yanwen Guo. On the error analysis of 3d gaussian splatting and an optimal projection strategy. In *Computer Vision – ECCV 2024*, pages 247–263, Cham, 2025. Springer Nature Switzerland.

[Jiang *et al.*, 2025] Junzhe Jiang, Nan Song, Jingyu Li, Xiatian Zhu, and Li Zhang. Realengine: Simulating autonomous driving in realistic context. *arXiv preprint arXiv:2505.16902*, 2025.

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

[Kundu *et al.*, 2022] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J. Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12871–12881, June 2022.

[Ljungbergh *et al.*, 2025] William Ljungbergh, Adam Tonderski, Joakim Johnander, Holger Caesar, Kalle Åström, Michael Felsberg, and Christoffer Petersson. Neuroncap: Photorealistic closed-loop safety testing for autonomous driving. In *European Conference on Computer Vision*, pages 161–177. Springer, 2025.

[Loper *et al.*, 2015] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: a skinned multi-person linear model. *ACM Trans. Graph.*, 34(6), October 2015.

[Mildenhall *et al.*, 2020] 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. In *ECCV*, 2020.

[Moenne-Loccoz *et al.*, 2024] Nicolas Moenne-Loccoz, Ashkan Mirzaei, Or Perel, Riccardo de Lutio, Janick Martinez Esturo, Gavriel State, Sanja Fidler, Nicholas Sharp, and Zan Gojcic. 3d gaussian ray tracing: Fast tracing of particle scenes. *ACM Transactions on Graphics and SIGGRAPH Asia*, 2024.

[Müller *et al.*, 2022] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022.

[Ost *et al.*, 2021] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2856–2865, June 2021.

[Peng *et al.*, 2025] Chensheng Peng, Chengwei Zhang, Yixiao Wang, Chenfeng Xu, Yichen Xie, Wenzhao Zheng, Kurt Keutzer, Masayoshi Tomizuka, and Wei Zhan. Desire-gs: 4d street gaussians for static-dynamic decomposition and surface reconstruction for urban driving scenes. In *Proceedings of the IEEE/CVF Conference on*

Computer Vision and Pattern Recognition (CVPR), pages 6782–6791, June 2025.

[Sun *et al.*, 2020] 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, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[Tonderski *et al.*, 2024] Adam Tonderski, Carl Lindström, Georg Hess, William Ljungbergh, Lennart Svensson, and Christoffer Petersson. Neurad: Neural rendering for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14895–14904, 2024.

[Wilson *et al.*, 2021] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021.

[Wu *et al.*, 2023] 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. In Lu Fang, Jian Pei, Guangtao Zhai, and Ruiping Wang, editors, *CICAI (1)*, volume 14473 of *Lecture Notes in Computer Science*, pages 3–15. Springer, 2023.

[Wu *et al.*, 2024] Hanfeng Wu, Xingxing Zuo, Stefan Leutenegger, Or Litany, Konrad Schindler, and Shengyu Huang. Dynamic lidar re-simulation using compositional neural fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19988–19998, June 2024.

[Wu *et al.*, 2025] Qi Wu, Janick Martinez Esturo, Ashkan Mirzaei, Nicolas Moenne-Loccoz, and Zan Gojcic. 3dgut: Enabling distorted cameras and secondary rays in gaussian splatting. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[Xiao *et al.*, 2021] Pengchuan Xiao, Zhenlei Shao, Steven Hao, Zishuo Zhang, Xiaolin Chai, Judy Jiao, Zesong Li, Jian Wu, Kai Sun, Kun Jiang, Yunlong Wang, and Diange Yang. Pandaset: Advanced sensor suite dataset for autonomous driving. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3095–3101, 2021.

[Xie *et al.*, 2023] Ziyang Xie, Junge Zhang, Wenye Li, Feihu Zhang, and Li Zhang. S-neRF: Neural radiance fields for street views. In *The Eleventh International Conference on Learning Representations*, 2023.

[Yan *et al.*, 2024] Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In *ECCV*, 2024.

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

[Yang *et al.*, 2024] 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. In *The Twelfth International Conference on Learning Representations*, 2024.

[Yuan *et al.*, 2024] Tianyuan Yuan, Yucheng Mao, Jiawei Yang, Yicheng Liu, Yue Wang, and Hang Zhao. Presight: Enhancing autonomous vehicle perception with city-scale nerf priors. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXXVII*, page 323–339, 2024.

[Zhang *et al.*, 2024] Baowen Zhang, Chuan Fang, Rakesh Shrestha, Yixun Liang, Xiaoxiao Long, and Ping Tan. Rade-gs: Rasterizing depth in gaussian splatting, 2024.

[Zhou *et al.*, 2024a] Hongyu Zhou, Longzhong Lin, Jiabao Wang, Yichong Lu, Dongfeng Bai, Bingbing Liu, Yue Wang, Andreas Geiger, and Yiyi Liao. Hugsim: A real-time, photo-realistic and closed-loop simulator for autonomous driving. *arXiv preprint arXiv:2412.01718*, 2024.

[Zhou *et al.*, 2024b] Hongyu Zhou, Jiahao Shao, Lu Xu, Dongfeng Bai, Weichao Qiu, Bingbing Liu, Yue Wang, Andreas Geiger, and Yiyi Liao. Hugs: Holistic urban 3d scene understanding via gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21336–21345, June 2024.

[Zhou *et al.*, 2024c] Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21634–21643, June 2024.

[Zhou *et al.*, 2025] Chenxu Zhou, Lvchang Fu, Sida Peng, Yunzhi Yan, Zhanhua Zhang, Yong Chen, Jiazhi Xia, and Xiaowei Zhou. LiDAR-RT: Gaussian-based ray tracing for dynamic lidar re-simulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[Zwicker *et al.*, 2001] Matthias Zwicker, Hanspeter Pfister, Jeroen van Baar, and Markus Gross. Ewa volume splatting. In *Proceedings of the Conference on Visualization '01*, VIS '01, page 29–36, 2001.