DecoupleGS: Interactive 3D Gaussian Splatting for End-to-End Autonomous Driving Testing

Siying Li[1,2], Ying Ni[1,2†], Jie Sun[1,2], Jian Sun[1,2], and Haotian Shi[1,2†]

[1] College of Transportation, Tongji University, Shanghai 201804, China
[2] Key Laboratory of Road and Traffic Engineering, Ministry of Education, Shanghai 201804, China
{siyingli,ying_ni,jie_sun,sunjian,shihaotian95}@tongji.edu.cn

Abstract. End-to-end (E2E) autonomous driving algorithms require rigorous closed-loop validation in simulation environments offering high visual fidelity, strong interactivity, and real-time performance. Existing approaches, from game engines to static neural rendering, inherently trade off these requirements and struggle with the dynamic scene composition essential for E2E testing. To bridge this gap, we propose a novel decoupled 3D Gaussian Splatting (3DGS) framework tailored for large-scale E2E evaluation. We fundamentally decompose scenes into a high-fidelity static background and manipulable dynamic agents using an object-centric canonical representation. To resolve resulting representational conflicts, we introduce three targeted modules: (1) asset compression via perceptual pruning and vector quantization for real-time traffic rendering; (2) map-guided geometric registration leveraging semantic topology to strictly align trajectories; and (3) proxy-based relighting transferring ambient illumination for seamless photometric integration. Extensive experiments demonstrate that DecoupleGS achieves a balanced fidelity-efficiency trade-off, improves metric and photometric consistency, and provides a practical closed-loop sensor simulation platform for E2E autonomous driving evaluation.

Keywords: Autonomous Driving · 3D Gaussian Splatting · End-to-End Testing · Sensor Simulation

1 Introduction

Autonomous driving (AD) is rapidly transitioning from classical modular pipelines to End-to-End (E2E) learning paradigms that map raw sensor observations directly to control commands [6, 8, 32]. While E2E algorithms reduce error accumulation [7,16,17], their safety and driving policies must be rigorously optimized across a broad, demanding set of scenarios. Since real-world data collection for such safety-critical scenarios is prohibitively costly and hazardous, high-fidelity virtual simulation has become an indispensable data engine for the continuous testing and optimization of E2E algorithms.

† Corresponding authors.

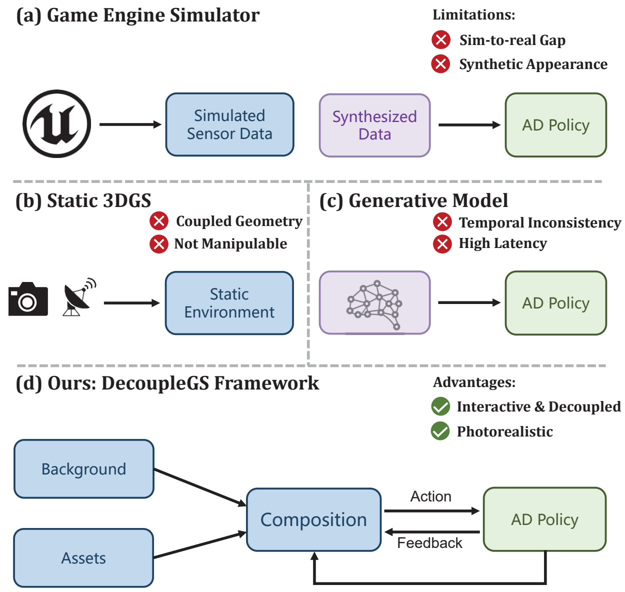

Fig. 1: Comparison of autonomous driving simulation paradigms. Traditional approaches face distinct limitations: (a) Game engine simulators suffer from sim-to-real gaps and synthetic appearances; (b) Generative models struggle with temporal inconsistencies and high inference latency; (c) Static 3DGS representations couple geometry and illumination, making dynamic manipulation impossible. In contrast, (d) our DecoupleGS framework introduces a modular composition of backgrounds and compressed assets, uniquely enabling an interactive, photorealistic, and closed-loop testing environment for AD policies.

For E2E testing, a practical closed-loop simulation platform must satisfy three key requirements: high visual and geometric fidelity, strong multi-agent interactivity, and sufficient real-time throughput [2, 9, 22, 30, 31]. However, existing sensor simulation approaches inherently trade off these requirements. Traditional game engines suffer from sim-to-real domain gaps, generative 2D models lack strict multi-view 3D consistency for spatial-temporal control loops, and static neural rendering methods lack the modularity to dynamically manipulate traffic flows.

Consequently, current methodologies cannot meet the integrated demands of a large-scale, rigorous testing pipeline. This gap stems from representational conflicts when seamlessly separating persistent static infrastructure from dynamic traffic participants. Merging these distinct entities introduces severe bottlenecks: storing and rendering multiple high-fidelity vehicles exhausts memory (efficiency conflict); unstructured neural coordinates fail to align with semantic metric maps (geometric conflict); and entangled environment lighting prevents dynamic assets from inheriting consistent illumination (photometric conflict).

To resolve these challenges, we propose DecoupleGS, a novel decoupled 3D Gaussian Splatting (3DGS) framework. The core task of this paper is to provide a comprehensive sensor simulation framework for E2E algorithms, achieving interactive, high-fidelity closed-loop evaluation that enables rigorous safety validation and dynamic multi-agent testing at scale. As compared in Fig. 1, while existing simulation paradigms suffer from the sim-to-real gap, temporal inconsistency, or coupled geometry, DecoupleGS uniquely bridges these gaps by organically composing persistent backgrounds and manipulable assets. We utilize an object-centric canonical representation to decompose the scene into a persistent high-fidelity static background and independent, compact 3DGS dynamic agents. Our pipeline integrates three targeted modules: an asset compression module utilizing perceptual redundancy pruning and vector quantization to mitigate memory bottlenecks; a map-guided geometric registration module leveraging semantic topology to strictly align agent trajectories; and a proxy-based relighting mechanism transferring ambient illumination for seamless photometric integration.

The main contributions of this study are threefold. First, we present a holistic 3DGS framework that integrates neural reconstruction fidelity with the interactivity and throughput required for large-scale closed-loop E2E testing. Second, we systematically address fundamental efficiency, geometry, and photometry conflicts by developing solver-agnostic modules for vehicle compression, map-guided registration, and relighting. Third, we validate that these modules enable plug-and-play insertion of multiple vehicles at interactive rates, establishing a robust and efficient data engine for E2E algorithms.

2 Related Work

2.1 Traditional Simulators and 2D Generative Methods

Game-engine-based simulators provide mature scene graphs, physically-based rendering primitives, and real-time rasterization pipelines, which yield strong interactivity and efficient runtime performance [2, 9, 22, 30, 31]. Their principal limitation for testing lies in the engineering cost of producing extensive collections of high-fidelity, diverse assets, and the reliance on hand-crafted materials and approximate physics, which often preserve a non-negligible visual or sensor-level domain gap from recorded real scenes. Conversely, generative 2D image synthesis—particularly latent diffusion models—has recently achieved impressive photorealism for scene synthesis and controllable simulation [15,24,29,35,39,42], but these methods are fundamentally 2D. They do not provide an intrinsic, multi-view, physically consistent 3D representation and therefore struggle with high inference latency and temporal coherence under the high frame-rate demands of control loops.

2.2 Neural 3D Reconstruction and Rendering

By contrast, neural 3D reconstruction and neural rendering methods—most notably NeRF variants and the recent 3DGS paradigm—reconstruct geometry and view-dependent appearance directly in a 3D coordinate frame, which reduces certain sim-to-real gaps and enables novel-view synthesis [1, 5, 12, 19, 26, 27, 41]. While recent frameworks advance closed-loop safety testing and dynamic scene composition [4, 14, 25, 40, 43, 45, 46], they still face bottlenecks for interactive E2E testing: many rely on pre-rendered alternatives, incorporate lighting into the reconstruction, or generally lack an efficient, map-aware modular decomposition that cleanly separates persistent infrastructure from movable traffic participants [13, 33, 34]. Consequently, naively reusing reconstructed vehicle instances across scenarios or moving them along arbitrary trajectories produces geometric and photometric inconsistencies.

2.3 Research Gaps for Interactive Neural Scene Composition

To address these challenges, recent attempts to make neural reconstructions interactive or to insert dynamic vehicles into neural scenes have advanced the state of the art [21,28,36,44] but expose three concrete research gaps that directly impede closed-loop E2E evaluation.

Efficiency and Scalability. High-fidelity vehicle models derived from 3DGS or dense Gaussian representations typically contain a large number of primitives. Although 3DGS substantially accelerates rendering compared with early implicit NeRF-based methods, the composition of multiple detailed vehicle instances remains computationally demanding, often leading to severe frame-rate degradation when many agents co-exist in a scene. This limitation arises because most reconstruction-oriented pipelines prioritize offline visual fidelity and lack runtime-oriented compression or pruning mechanisms, which are essential for maintaining interactive performance in large-scale multi-agent simulation [40, 46].

Geometric Alignment and Metric Consistency. Neural reconstructions are typically represented in unstructured learned coordinates, whereas planning and control operate in metric map space, leading to alignment inconsistencies in closed-loop simulation. Existing object-centric methods [28, 44] focus on single-object editing or require per-scene optimization and therefore do not yet provide robust, map-aware registration for long-horizon multi-agent sequences.

Photometric Integration and Relighting. In neural scene representations, lighting is often entangled with geometry and appearance, preventing inserted vehicles from automatically inheriting consistent ambient illumination, shadows, and environmental reflections. Although recent relighting and compositing pipelines demonstrate promising capabilities for object-level relighting and shadow synthesis [47], they typically incur substantial computational overhead or assume a limited number of edited objects and fixed camera configurations. These constraints hinder their applicability to large-scale, interactive multi-vehicle simulation.

3 Method

To enable large-scale closed-loop E2E evaluation, we aim to compose a photo-realistic yet interactive simulation where (i) the static infrastructure remains persistent across time, and (ii) a variable number of traffic agents can be instantiated, moved, relit, and rendered in response to planner-driven scene states. As illustrated in Fig. 2, DecoupleGS is organized around a decoupled neural scene representation and three tightly coupled functional modules. The static background and canonical dynamic assets are represented in separate coordinate spaces, while asset compression, map-guided registration, and proxy-based relighting respectively address the efficiency, geometric, and photometric conflicts that arise during dynamic scene composition. The transformed and relit primitives are finally composited through a unified rasterizer to produce multi-view observations for closed-loop E2E testing.

3.1 Decoupling and Unified Rendering

Applying 3DGS to interactive E2E testing is fundamentally bottlenecked by its inherent assumption of static scenes. We introduce a dual-stream architecture that represents the background and dynamic agents in separate coordinate spaces, mathematically fusing them prior to rasterization without the need for network inference.

Object-Centric Canonical Decomposition. Formally, the global scene Ω at timestamp t is defined as the spatial union of a time-invariant background field $\mathcal{S}_{bg}$ and a set of K dynamic agents:

$$\Omega(t) = \mathcal{S}_{bg} \cup \left(\bigcup_{k=1}^{K} \mathcal{T}_k(t) \circ \mathcal{V}_k \right) \tag{1}$$

The background $\mathcal{S}_{bg}$ consists of Gaussian primitives anchored in the world coordinate system (WCS). Conversely, each dynamic agent k is parameterized as a canonical Gaussian volume $\mathcal{V}_k$ within a standardized local coordinate system (LCS). Given the agent's 6-DoF rigid transform $\mathcal{T}_k(t) = [\mathbf{R}_k \mid \mathbf{t}_k] \in SE(3)$ mapping LCS to WCS, we apply the transformation to the spatial attributes (means μ and covariances Σ) of all primitives within $\mathcal{V}_k$:

$$\mu_i^{(W)} = \mathbf{R}_k \mu_i^{(L)} + \mathbf{t}_k, \quad \Sigma_i^{(W)} = \mathbf{R}_k \Sigma_i^{(L)} \mathbf{R}_k^\top \tag{2}$$

Crucially, 3DGS encodes view-dependent appearance via Spherical Harmonics (SH). When the geometry undergoes a rotation $\mathbf{R}_k$, the intrinsic SH coefficients $c_i^{(L)}$ must be correspondingly rotated to preserve correct view-dependent reflections relative to the new orientation. We apply the Wigner D-matrix $\mathbf{D}(\mathbf{R}_k)$ to rigorously align the SH bands:

$$c_i^{(W)} = \mathbf{D}(\mathbf{R}_k) c_i^{(L)} \tag{3}$$

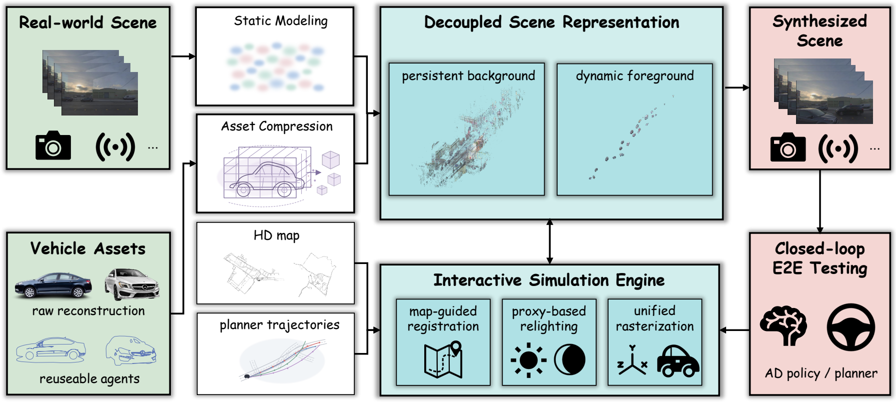

Fig. 2: Overview of the DecoupleGS framework. DecoupleGS decouples persistent static backgrounds from canonical dynamic assets, and integrates asset compression, map-guided registration, proxy-based relighting, and unified rasterization to support interactive closed-loop E2E evaluation.

While this mathematically preserves the asset's intrinsic specularities under rigid motion, resolving the ultimate photometric mismatch with the new background illumination is deferred to our proxy-based relighting module (Sec. 3.4).

Unified Rasterization and Runtime Optimization. To ensure physically accurate occlusion, all primitives from $\Omega(t)$ are merged into a unified aggregate list $\mathcal{P}$. For each pixel u, primitives overlapping u are depth-sorted relative to the novel camera center and composited via continuous front-to-back α-blending. To maintain the robust interactive frame rates required for large-scale E2E testing, we integrate several hardware-level optimizations: (a) conservative frustum culling to aggressively bound $\mathcal{P}$; (b) decoupling the sorting radix, where the static background splats are cached and incrementally reused; and (c) computing batched spatial updates exclusively for the moving canonical volumes.

3.2 Asset Compression

Directly rendering high-dimensional raw 3DGS assets for dense traffic is computationally prohibitive. As detailed in Fig. 3, we introduce a semantic-aware compression pipeline consisting of importance scoring, pruning, and vector quantization to produce compact canonical assets suitable for real-time multi-agent composition.

Explicit Importance Scoring and Pruning. Instead of relying on black-box neural networks, we compute an explicit scalar importance score s_i for each primitive g_i that summarizes its visibility, photometric contribution, and local geometric saliency:

$$s_i = w_{\text{vis}} V_i + w_{\text{col}} D_i + w_{\text{ent}} H_i \tag{4}$$

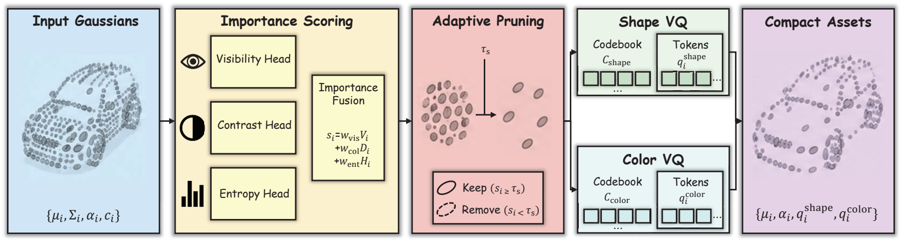

Fig. 3: Illustration of the semantic-aware asset compression module. Canonical vehicle Gaussians undergo explicit importance scoring (V_i, D_i, H_i), adaptive pruning via threshold τ_s, and high-dimensional attribute vector quantization to form compact canonical volumes.

Here, the visibility estimate V_i is the expected opacity contribution over sampled training views; the color contrast term D_i measures perceptual distinctiveness versus the local background; and the entropy term H_i captures texture complexity inside the primitive neighborhood. Primitives with s_i below a strict threshold τ_s (empirically set to 0.005) are pruned. This operation significantly reduces the primitive count, concentrating computational resources on essential structural details like wheels and chassis boundaries. The weighting coefficients w_{vis}, w_{col}, and w_{ent} are empirically determined via grid search to optimally balance geometric preservation and color fidelity, demonstrating robust generalization across different vehicle types.

Attribute Vector Quantization. To further minimize the memory footprint, Vector Quantization (VQ) is applied to the high-dimensional attributes of retained primitives. The covariance Σ_i and color coefficients c_i are replaced by their nearest entries in small, learnable codebooks $\mathcal{C}_{shape}$ and $\mathcal{C}_{color}$:

$$\Sigma_i \approx \mathcal{C}_{\text{shape}}[q_i^{\text{shape}}], \quad c_i \approx \mathcal{C}_{\text{color}}[q_i^{\text{color}}] \tag{5}$$

This discretization replaces heavy floating-point tensors with compact integer indices q_i. Specifically, we utilize attribute-specific codebooks (e.g., $K_{\text{color}} = 1024$ for SH color coefficients and $K_{\text{shape}} = 512$ for covariance attributes) optimized via Exponential Moving Average (EMA) during compression.

3.3 Map-Guided Geometric Registration

Integrating dynamic agents requires strict geometric alignment. It is crucial to note that for the ego-vehicle evaluated in closed-loop E2E testing, its trajectory is driven entirely by the unconstrained relative pose updates from the planner to preserve maneuvers like lane changes. The sequence-level registration described below, and illustrated in Fig. 4, is specifically designed to accurately inject background traffic agents into the reconstructed shared coordinate system.

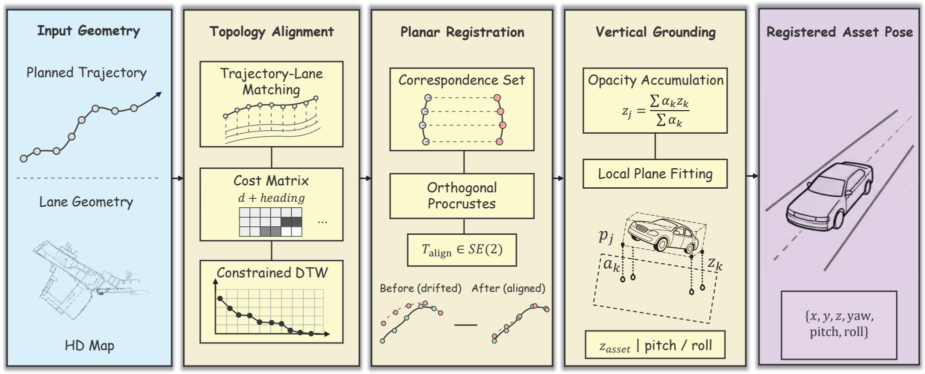

Fig. 4: The map-guided geometric registration pipeline. It enforces strict metric alignment via trajectory-to-lane DTW sequence alignment, SE(2) Procrustes 2D transformation, and robust opacity-accumulated 3D vertical grounding.

Semantic Topology and Planar Alignment. We obtain vectorized lane centerlines $\mathcal{L}$ from dataset-provided HD maps when available, and use MapTRv2 [23] to extract lane topology otherwise. Given a background agent's 2D trajectory $\mathcal{T} = \{p_t\}_{t=1}^{T}$ and the corresponding lane geometry, we build a cost matrix that penalizes both Euclidean distance and heading mismatch. We utilize constrained Dynamic Time Warping (DTW) to find the optimal warping path establishing trajectory-to-lane correspondences. Subsequently, an Orthogonal Procrustes analysis computes the globally optimal 2D rigid transform $T_{align} \in SE(2)$ to eliminate systemic lateral drift.

Vertical Grounding via Opacity Accumulation. Since 3DGS lacks a continuous surface mesh, standard ray-casting fails. We employ opacity-weighted depth accumulation to ground the vehicles. For anchor points p_j defined at the bottom corners of the asset's bounding box, the vertical height z_j is computed as the opacity-weighted average of the intersected background gaussians within a vertical column $\mathcal{N}$:

$$z_j = \frac{\sum_{k \in \mathcal{N}} \alpha_k z_k}{\sum_{k \in \mathcal{N}} \alpha_k} \tag{6}$$

A local ground plane is then fitted using least-squares minimization over the anchors $\{z_j\}$. The final vertical position z_{asset} and attitude (pitch and roll) of the asset are derived from the estimated plane normal, ensuring that the wheels maintain physically plausible contact with the terrain.

3.4 Relighting

To resolve the photometric conflict where inserted assets retain intrinsic illumination that mismatches the novel environment, we develop a lightweight, proxy-based relighting module that operates without online neural network inference.

Local Probe Sampling and Linear SH Transfer. We treat the background SH coefficients as a dense field of light probes. For a vehicle located at x, we construct a local ambient descriptor $\mathcal{L}(x)$ by aggregating the SH coefficients of neighboring background gaussians c_g, weighted by their visibility V_g and spatial distance:

$$\mathcal{L}(x) = \frac{\sum_{g \in \mathcal{N}(x)} w_g c_g}{\sum_{g \in \mathcal{N}(x)} w_g}, \quad w_g = \exp\left(-\frac{||x - \mu_g||^2}{2\sigma_p^2}\right) V_g \tag{7}$$

To transfer this environmental lighting, we apply a pre-calibrated linear modulation operator to the canonical vehicle's SH coefficients c_{can}:

$$c_{out} = \mathcal{T}(\mathcal{L}(x)) \otimes c_{can} + b(\mathcal{L}(x)) \tag{8}$$

where $\mathcal{T}(\cdot)$ and $b(\cdot)$ are efficient transformation matrices. Rather than relying on online neural network inference, these operators are derived offline by solving a least-squares regression over a synthetic dataset of canonical assets rendered under diverse pre-captured environmental probes. This analytical calibration completely avoids per-frame MLP inference while successfully rebaking the assets with the local lighting context.

Proxy-based Contact Shadows. To anchor the object visually, we synthesize contact shadows on the estimated local ground plane. We project a parametric shadow mask $M_{shadow}(u)$ scaled by the object footprint. Crucially, the intensity of the background gaussians under this mask is modulated dynamically by the dominant light intensity I_{dom} extracted from the DC component of the ambient descriptor $\mathcal{L}(x)$:

$$c_{\text{final}}(u) = c_{\text{ground}}(u) \cdot (1 - \lambda \cdot I_{\text{dom}} \cdot M_{\text{shadow}}(u)) \tag{9}$$

This efficient approximation ensures that shadows become appropriately faint in overcast conditions or intense under direct sunlight, completing the photometric integration seamlessly.

4 Experiments

This section evaluates DecoupleGS from four complementary perspectives: rendering efficiency, geometric and photometric fidelity, open-loop sim-to-real consistency, and closed-loop E2E testing. We further ablate the proposed compression, registration, and relighting modules to verify how each component resolves the efficiency, geometric, and photometric conflicts in dynamic neural scene composition.

4.1 Experimental Setup

Datasets and assets. We utilize nuScenes [3] and PandaSet [37] for diverse background scenarios, and source manipulable canonical vehicles from 3DRealCar [10]. We curate 15 dynamic clips from nuScenes and 10 sequences from

PandaSet, covering challenging illumination conditions such as dusk, night, and overcast scenes. Dynamic objects are removed using Segment Anything Model (SAM) [20] masks with morphological dilation to suppress boundary artifacts. We sample 20 vehicle assets from 3DRealCar, covering sedans, SUVs, and large commercial vehicles.

Baselines and metrics. For rendering efficiency and compression, we compare against Plenoxels [12], Vanilla 3DGS [19], and LightGaussian [11]. For simulator-level fidelity and closed-loop testing, we compare with HUGSIM [45], OASim [38], and RealEngine [18]. Downstream E2E policy testing is conducted with UniAD [16] and VAD [17]. Rendering fidelity is measured by PSNR-Veh, PSNR-All, SSIM, and LPIPS, while runtime performance is evaluated using FPS and VRAM. Geometric consistency is quantified by Trajectory Average Displacement Error (ADE) and Ground Penetration Rate (GPR). Photometric realism is evaluated using Masked PSNR, Peak Intensity Error (PIE), and Peak Angular Error (PAE). Open-loop planner consistency is measured by mADE and minTTC, while closed-loop E2E testing adopts Driving Score (DS), Success Rate (SR), Route Completion (RC), and minTTC.

Implementation and closed-loop protocol. All experiments are conducted on a single NVIDIA RTX 4090 GPU with 24 GB VRAM using custom CUDA kernels. The static background is optimized for 30k iterations, and the asset compression threshold τ_s is set to 0.005. All closed-loop evaluations are fully interactive. At each simulation step, DecoupleGS renders multi-view images for the tested planner, the E2E model predicts the ego trajectory, and background agents react online via coupled IDM and MOBIL behavioral models. The global scene state is then synchronized and re-rendered for the next step.

4.2 Rendering Efficiency and Scalability

Asset compression. Table 1 reports results under three traffic densities: Size S with 1–2 vehicles, Size M with 3–5 vehicles, and Size L with 6–10+ vehicles. Vanilla 3DGS preserves the highest rendering fidelity but becomes inefficient as traffic density increases, while LightGaussian achieves the most aggressive compression and fastest rendering at the cost of lower scene fidelity. DecoupleGS is designed for a different objective: it preserves the high-fidelity static background and compresses only dynamic assets, yielding the most balanced fidelity-efficiency trade-off. As shown in the final column, among the compared methods, DecoupleGS is the only one that consistently remains within the top-2 across all fidelity and efficiency metrics under every traffic density, which is more suitable for interactive E2E testing than optimizing a single isolated metric.

Scalability. We further conduct a stress test by increasing the number of dynamic agents to 50. As shown in Fig. 5, Vanilla 3DGS suffers rapid memory growth and encounters out-of-memory failures before reaching dense traffic. In contrast, DecoupleGS relies on a persistent background and compact VQ-indexed assets, leading to approximately linear memory growth. It maintains usable throughput even under extreme multi-agent settings, which is essential for large-scale closed-loop policy evaluation.

Table 1: Asset compression results under different traffic densities. Best values are bolded, second-best values are shaded in gray, and DecoupleGS rows are highlighted. The final column summarizes whether a method consistently remains top-2 across all fidelity and efficiency metrics.

Size	Methods	Fidelity				Efficiency		Top-2 Consist.
		PSNR-Veh ↑	PSNR-All ↑	SSIM ↑	LPIPS ↓	FPS ↑	VRAM ↓	
S	Plenoxels [12]	21.45	23.06	0.795	0.510	8.5	2.5GB	No
	Vanilla 3DGS [19]	**28.52**	**29.41**	**0.903**	**0.243**	55.4	1.2GB	No
	LightGaussian [11]	26.21	27.50	0.865	0.281	**75.2**	**180MB**	No
	DecoupleGS (Ours)	28.10	29.25	0.898	0.252	68.5	850MB	**Yes**
M	Plenoxels [12]	20.12	23.08	0.626	0.463	6.2	3.1GB	No
	Vanilla 3DGS [19]	**27.05**	**27.21**	**0.815**	**0.214**	32.5	2.1GB	No
	LightGaussian [11]	25.14	26.40	0.781	0.250	**58.6**	**240MB**	No
	DecoupleGS (Ours)	26.85	27.12	0.805	0.218	52.4	950MB	**Yes**
L	Plenoxels [12]	18.55	21.08	0.719	0.379	3.5	4.8GB	No
	Vanilla 3DGS [19]	**25.50**	**25.14**	**0.785**	**0.255**	12.1	3.8GB	No
	LightGaussian [11]	23.52	24.10	0.720	0.295	**45.0**	**320MB**	No
	DecoupleGS (Ours)	25.24	25.05	0.772	0.260	38.5	1.1GB	**Yes**

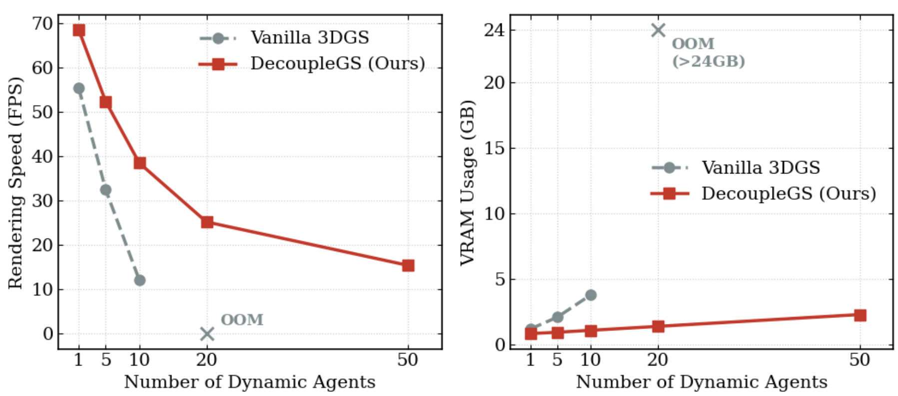

Fig. 5: Scalability evaluation. DecoupleGS avoids the OOM bottlenecks of Vanilla 3DGS, demonstrating stable VRAM growth and usable frame rates up to 50 agents.

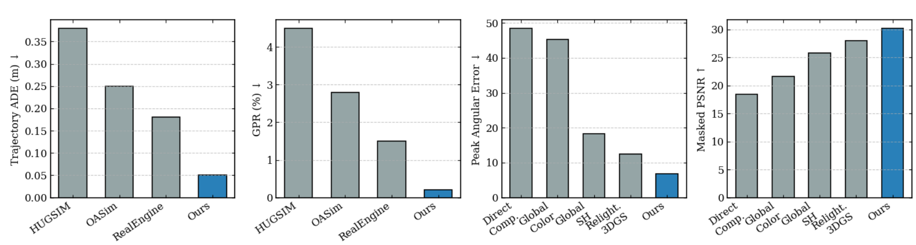

Fig. 6: Quantitative comparisons of geometric registration and photometric realism against baseline methods. DecoupleGS achieves lower trajectory drift, fewer ground-penetration artifacts, and more consistent relighting.

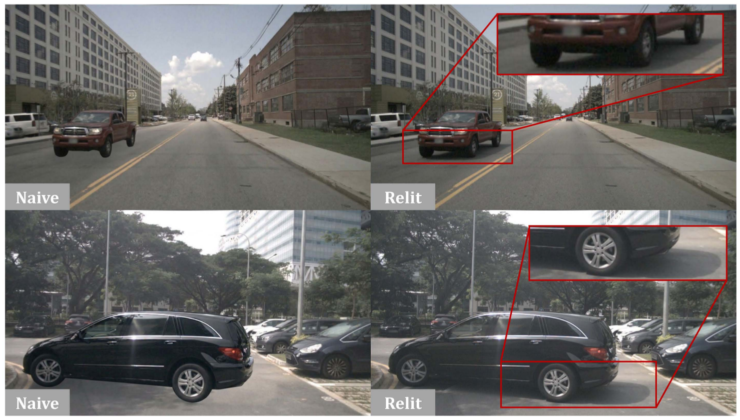

Fig. 7: Qualitative comparison of the relighting module, demonstrating seamless integration of dynamic assets under various illuminations.

Table 2: Open-loop sim-to-real consistency and closed-loop E2E simulator comparison. Open-loop results measure UniAD behavior under rendered observations, using real images as the reference input. Bold values indicate the best result among simulated inputs. Closed-loop scores are normalized to $[0, 1]$ for consistency with Table 3.

Setting	Metric	Real Input	Ours	HUGSIM [45]	RealEngine [18]	OASim [38]
Open-loop	mADE (m) ↓	0.76	**0.82**	0.94	1.15	1.02
	minTTC (s) ↑	3.5	**3.3**	2.8	2.4	2.9
Closed-loop	Driving Score ↑	–	**0.884**	0.765	0.682	0.748
	Route Completion ↑	–	**0.956**	0.814	0.795	0.851
	minTTC (s) ↑	–	**3.3**	2.3	2.6	2.8
	Rendering FPS ↑	–	**45**	12	32	18

4.3 Geometric and Photometric Fidelity

Map-guided geometric consistency. Fig. 6 evaluates the metric alignment of inserted agents. Compared with HUGSIM and RealEngine, our map-guided registration significantly reduces trajectory drift and ground penetration. By combining trajectory-to-lane sequence alignment, SE(2) planar correction, and opacity-accumulated vertical grounding, DecoupleGS achieves accurate lane-level placement while maintaining physically plausible contact with the road surface.

Photometric realism. The same figure also reports photometric realism under held-out vehicle re-insertion. Our proxy-based relighting obtains the lowest PAE and competitive Masked PSNR by transferring local ambient illumination to canonical assets without online neural inference. As shown in Fig. 7, inserted vehicles inherit scene-dependent lighting and contact shadows, avoiding the common mismatch between foreground appearance and background illumination.

Table 3: Closed-loop E2E AD testing across scenario difficulty levels. Values are Mean ± Std over 50 independent episodes.

Difficulty	Method	DS	SR	RC	minTTC
Easy	UniAD [16]	0.725±0.04	0.850±0.03	0.780±0.05	0.812±0.06
	VAD [17]	0.680±0.05	0.815±0.04	0.720±0.06	0.785±0.05
Medium	UniAD [16]	0.540±0.06	0.710±0.05	0.610±0.07	0.650±0.08
	VAD [17]	0.450±0.07	0.630±0.06	0.520±0.08	0.580±0.07
Hard	UniAD [16]	0.380±0.08	0.520±0.07	0.450±0.09	0.480±0.10
	VAD [17]	0.250±0.09	0.410±0.08	0.320±0.10	0.390±0.09
Extreme	UniAD [16]	0.195±0.07	0.310±0.09	0.250±0.08	0.280±0.08
	VAD [17]	0.120±0.05	0.220±0.07	0.180±0.06	0.210±0.06

Fig. 8: Representative camera views of UniAD evaluated across generated scenarios of increasing difficulty, from Easy to Extreme.

4.4 Open-loop Sim-to-real Consistency

A reliable simulator should not only render visually plausible images but also preserve the behavior of downstream planners. We therefore conduct an open-loop sim-to-real consistency test using UniAD. Real images serve as the reference input, while different simulators provide rendered observations for the same driving scenes. As shown in Table 2, DecoupleGS most closely matches the real-data behavior in both mADE and minTTC, indicating a smaller sim-to-real behavioral gap than existing neural simulators.

4.5 Closed-loop E2E Testing

Simulator-level comparison. Beyond open-loop consistency, Table 2 compares different simulators under interactive closed-loop testing. DecoupleGS achieves the highest normalized Driving Score and Route Completion while maintaining

Table 4: System-level ablation study evaluated on medium-density scenarios. Values in parentheses indicate changes relative to the full framework.

Methods	FPS ↑	Trajectory ADE ↓	Peak Angular Error ↓
w/o asset compression	8.5 (↓26.9)	0.05 (=)	6.8 (=)
w/o map-guided reg.	36.2 (↑0.8)	0.48 (↑0.43)	6.8 (=)
w/o relighting	38.5 (↑3.1)	0.05 (=)	48.5 (↑41.7)
Full Framework (Ours)	**35.4**	**0.05**	**6.8**

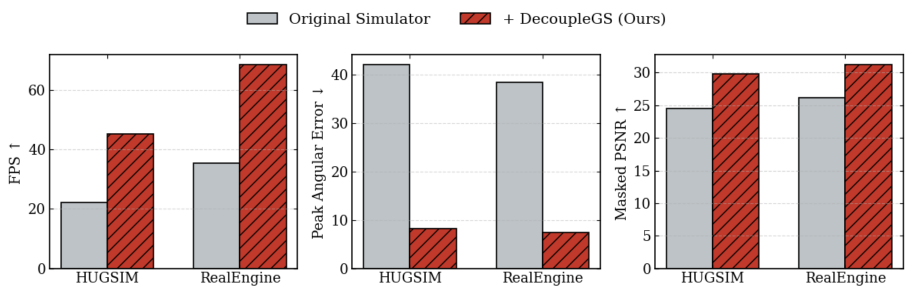

Fig. 9: Plug-and-play capability. Integrating our modules into existing simulators improves rendering speed and photometric accuracy.

the largest safety margin measured by minTTC. It also runs at 45 FPS, substantially faster than HUGSIM and OASim. These results demonstrate that DecoupleGS is not merely a rendering module, but a practical closed-loop sensor simulation engine for E2E policy evaluation.

Difficulty-based stress testing. We further evaluate UniAD and VAD across four scenario difficulty levels, as shown in Table 3, with representative closed-loop visualizations provided in Fig. 8. The difficulty is controlled by traffic density, illumination, and background-agent aggressiveness. Easy scenarios contain sparse traffic and clear daytime illumination. Medium scenarios introduce moderate density and standard urban maneuvers. Hard scenarios include dense traffic and challenging illumination such as dusk or heavy overcast. Extreme scenarios further introduce adversarial lighting and safety-critical behaviors such as sudden braking and aggressive cut-ins.

Both planners degrade as the scenario difficulty increases. UniAD maintains a relatively high completion rate in Easy scenarios but suffers under severe occlusion and dense interactions, while VAD is more sensitive to dense multi-agent constraints, causing a steeper drop in Driving Score under Hard and Extreme settings. The qualitative examples in Fig. 8 further illustrate that DecoupleGS can synthesize visually diverse and increasingly challenging closed-loop scenarios, including dense traffic, low-light conditions, and safety-critical interactions. These results show that DecoupleGS can generate structured and controllable stress tests that expose planner vulnerabilities.

4.6 Ablation Studies

Table 4 decomposes the effect of each core module under medium-density scenarios. Without asset compression, throughput collapses from 35.4 FPS to 8.5 FPS, making interactive multi-agent simulation impractical. Without map-guided registration, ADE increases from 0.05 m to 0.48 m, showing that neural coordinates alone do not guarantee metric-consistent placement. Without relighting, PAE increases from 6.8° to 48.5°, exposing severe foreground-background illumination mismatch. The full framework therefore offers the best balanced trade-off across efficiency, geometric alignment, and photometric consistency.

Plug-and-play capability. We further validate that the proposed modules are solver-agnostic. As shown in Fig. 9, integrating DecoupleGS components into existing simulators improves runtime and photometric consistency, indicating that our compression, registration, and relighting modules can serve as general-purpose building blocks for interactive neural driving simulation.

4.7 Limitations

While DecoupleGS enables efficient and consistent closed-loop testing, it still has limitations. First, its proxy-based relighting favors runtime efficiency over physically exact global illumination, and may struggle with complex multi-bounce lighting, specular effects, and cast shadows. Second, map-guided registration relies on reliable semantic topology, making multi-level roads, overpasses, under-bridge regions, and poorly detected lanes more challenging. Third, the current framework mainly targets rigid traffic agents and RGB simulation. Future work will explore non-rigid foregrounds, full background compression, richer interactive behaviors, and unified RGB-LiDAR sensor simulation.

5 Conclusion

This paper presents DecoupleGS, a decoupled 3DGS framework for interactive E2E autonomous driving testing. By separating persistent backgrounds from manipulable agents, DecoupleGS addresses efficiency, geometric, and photometric conflicts through asset compression, map-guided registration, and proxy-based relighting. Experiments demonstrate a balanced fidelity-efficiency trade-off, improved sim-to-real consistency, and fully interactive closed-loop policy evaluation. Future work will extend DecoupleGS toward richer illumination, non-rigid agents, and unified RGB-LiDAR simulation.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 52502514) and the Fundamental Research Funds for the Central Universities.

References

1. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In: IEEE Conf. Comput. Vis. Pattern Recog. (2022)
2. Behrisch, M., Bieker, L., Erdmann, J., Krajzewicz, D.: SUMO: Simulation of urban mobility. In: Int. Conf. Advances Syst. Simul. (2011)
3. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuScenes: A multimodal dataset for autonomous driving. In: IEEE Conf. Comput. Vis. Pattern Recog. (2020)
4. Cao, W., Hallgarten, M., Li, T., Dauner, D., Gu, X., Wang, C., Miron, Y., Aiello, M., Li, H., Gilitschenski, I., Ivanovic, B., Pavone, M., Geiger, A., Chitta, K.: Pseudo-simulation for autonomous driving. In: Conf. Robot Learn. Proceedings of Machine Learning Research, vol. 305, pp. 4709–4722. PMLR (2025)
5. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: TensoRF: Tensorial radiance fields. In: Eur. Conf. Comput. Vis. (2022)
6. Chen, L., Wu, P., Chitta, K., Jaeger, B., Geiger, A., Li, H.: End-to-end autonomous driving: Challenges and frontiers. IEEE Trans. Pattern Anal. Mach. Intell. **46**(12), 10164–10183 (2024)
7. Chitta, K., Prakash, A., Jaeger, B., Yu, Z., Renz, K., Geiger, A.: Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. IEEE Trans. Pattern Anal. Mach. Intell. **45**, 12878–12895 (2023)
8. Codevilla, F., Müller, M., López, A., Koltun, V., Dosovitskiy, A.: End-to-end driving via conditional imitation learning. In: IEEE Int. Conf. Robot. Autom. (2018)
9. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: Conf. Robot Learn. (2017)
10. Du, X., Wang, Y., Sun, H., Wu, Z., Sheng, H., Wang, S., Ying, J., Lu, M., Zhu, T., Zhan, K., Yu, X.: 3DRealCar: An in-the-wild RGB-D car dataset with 360-degree views. In: Int. Conf. Comput. Vis. pp. 26488–26498 (2025)
11. Fan, Z., Wang, K., Wen, K., Zhu, Z., Xu, D., Wang, Z.: LightGaussian: Unbounded 3D gaussian compression with 15x reduction and 200+ FPS. In: Adv. Neural Inform. Process. Syst. (2024). `https://doi.org/10.52202/079017-4447`
12. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: IEEE Conf. Comput. Vis. Pattern Recog. (2022)
13. Fridovich-Keil, S., et al.: K-planes: Explicit radiance fields in space, time, and appearance. In: IEEE Conf. Comput. Vis. Pattern Recog. (2023)
14. Ge, J., Liu, Z., Fan, L., Jiang, Y., Su, J., Li, Y., Zhang, Z., Chen, S.: Unraveling the effects of synthetic data on end-to-end autonomous driving. In: Int. Conf. Comput. Vis. pp. 28859–28869 (2025)
15. Hu, A., Russell, L., Yeo, H., et al.: GAIA-1: A generative world model for autonomous driving. arXiv preprint arXiv:2309.17080 (2023)
16. Hu, Y., Yang, J., Chen, L., Li, K., Sima, C., Zhu, X., Chai, S., Du, S., Lin, T., Wang, W., et al.: Planning-oriented autonomous driving. In: IEEE Conf. Comput. Vis. Pattern Recog. (2023)
17. Jiang, B., Chen, S., Xu, Q., Liao, B., Chen, J., Zhou, H., Zhang, Q., Liu, W., Huang, C., Wang, X.: VAD: Vectorized scene representation for efficient autonomous driving. In: Int. Conf. Comput. Vis. (2023)
18. Jiang, J., Song, N., Li, J., Zhu, X., Zhang, L.: RealEngine: Simulating autonomous driving in realistic context. arXiv preprint arXiv:2505.16902 (2025)

19. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3D gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4) (2023)
20. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollar, P., Girshick, R.: Segment anything. In: Int. Conf. Comput. Vis. pp. 4015–4026 (2023)
21. Kundu, A., et al.: Panoptic neural fields: A semantic object-aware neural scene representation. In: IEEE Conf. Comput. Vis. Pattern Recog. (2022)
22. Li, Q., Peng, Z., Feng, L., Zhang, Q., Xue, Z., Zhou, B.: Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. IEEE Trans. Pattern Anal. Mach. Intell. **45**(3), 3461–3475 (2023)
23. Liao, B., Chen, S., Zhang, Y., et al.: MapTRv2: An end-to-end framework for online vectorized HD map construction. Int. J. Comput. Vis. **133**(3), 1352–1374 (2025)
24. Liu, Z., Fu, D., Cai, P., Wang, L., Liu, Y., Ren, Y., Shi, B., Wang, J.: SymDrive: Realistic and controllable driving simulator via symmetric auto-regressive online restoration. arXiv preprint arXiv:2512.21618 (2025)
25. Ljungbergh, W., Tonderski, A., Johnander, J., Caesar, H., Åström, K., Felsberg, M., Petersson, C.: NeuroNCAP: Photorealistic closed-loop safety testing for autonomous driving. In: Eur. Conf. Comput. Vis. pp. 161–177 (2024)
26. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: Representing scenes as neural radiance fields for view synthesis. In: Eur. Conf. Comput. Vis. (2020)
27. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans. Graph. **41**(4) (2022)
28. Ost, J., et al.: Neural scene graphs for dynamic scenes. In: IEEE Conf. Comput. Vis. Pattern Recog. (2021)
29. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: IEEE Conf. Comput. Vis. Pattern Recog. (2022)
30. Rong, G., Shin, B.H., Tabatabaee, H., Lu, Q., Lemke, S., Možeiko, M., Boise, E., Uhm, G., Gerow, M., Mehta, S., Agafonov, E., Kim, T.H., Sterner, E., Ushiroda, K., Reyes, M., Zelenkovsky, D., Kim, S.: LGSVL simulator: A high fidelity simulator for autonomous driving. In: IEEE Intell. Transp. Syst. Conf. pp. 1–6 (2020). `https://doi.org/10.1109/ITSC45102.2020.9294422`
31. Shah, S., Dey, D., Lovett, C., Kapoor, A.: AirSim: High-fidelity visual and physical simulation for autonomous vehicles. In: Field and Service Robotics (2018)
32. Tampuu, A., Matiisen, T., Semikin, M., Fishman, D., Muhammad, N.: A survey of end-to-end driving: Architectures and training methods. IEEE Trans. Neural Netw. Learn. Syst. **33**(4), 1364–1384 (2022)
33. Tancik, M., et al.: Block-nerf: Scalable large scene neural view synthesis. In: IEEE Conf. Comput. Vis. Pattern Recog. (2022)
34. Turki, H., et al.: Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In: IEEE Conf. Comput. Vis. Pattern Recog. (2022)
35. Wang, X., Zhu, Z., Huang, G., Chen, X., Zhu, J., Lu, J.: DriveDreamer: Towards real-world-drive world models for autonomous driving. In: Eur. Conf. Comput. Vis. pp. 55–72 (2024). `https://doi.org/10.1007/978-3-031-73195-2_4`
36. Wu, Z., Liu, T., Luo, L., Zhong, Z., Chen, J., Xiao, H., Hou, C., Lou, H., Chen, Y., Yang, R., et al.: MARS: An instance-aware, modular and realistic simulator for autonomous driving. In: Artificial Intelligence. Lecture Notes in Computer Science, vol. 14473, pp. 3–15 (2023). `https://doi.org/10.1007/978-981-99-8850-1_1`

37. Xiao, P., Shao, Z., Hao, S., Zhang, Z., Chai, X., Jiao, J., Li, Z., Wu, J., Sun, K., Jiang, K., Wang, Y., Yang, D.: PandaSet: Advanced sensor suite dataset for autonomous driving. In: IEEE Intell. Transp. Syst. Conf. (2021)
38. Yan, G., Pi, J., Guo, J., Luo, Z., Dou, M., Deng, N., Huang, Q., Fu, D., Wen, L., Cai, P., Gao, X., Cai, X., Zhang, B., Yang, X., Bai, Y., Zhou, H., Shi, B.: OASim: An open and adaptive simulator based on neural rendering for autonomous driving. arXiv preprint arXiv:2402.03830 (2024)
39. Yan, T., Wu, D., Han, W., Jiang, J., Zhou, X., Zhan, K., Xu, C.Z., Shen, J.: DrivingSphere: Building a high-fidelity 4D world for closed-loop simulation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 27531–27541 (2025)
40. Yan, Y., Lin, H., Zhou, C., Wang, W., Sun, H., Zhan, K., Lang, X., Zhou, X., Peng, S.: Street Gaussians: Modeling dynamic urban scenes with gaussian splatting. In: Eur. Conf. Comput. Vis. (2024)
41. Yan, Z., Low, W.F., Chen, Y., Lee, G.H.: Multi-scale 3D gaussian splatting for anti-aliased rendering. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 20923–20931 (2024)
42. Yang, X., Wen, L., Wei, T., Ma, Y., Mei, J., Li, X., Lei, W., Fu, D., Cai, P., Dou, M., He, L., Liu, Y., Shi, B., Qiao, Y.: DriveArena: A closed-loop generative simulation platform for autonomous driving. In: Int. Conf. Comput. Vis. pp. 26933–26943 (2025)
43. Yang, Z., Chen, Y., Wang, J., Manivasagam, S., Ma, W.C., Yang, A.J., Urtasun, R.: UniSim: A neural closed-loop sensor simulator. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 1389–1399 (2023)
44. Yu, H.X., et al.: Unsupervised discovery of object radiance fields. In: Int. Conf. Learn. Represent. (2022)
45. Zhou, H., Lin, L., Wang, J., Lu, Y., Bai, D., Liu, B., Wang, Y., Geiger, A., Liao, Y.: HUGSIM: A real-time, photo-realistic and closed-loop simulator for autonomous driving. IEEE Trans. Pattern Anal. Mach. Intell. **48**(4), 4673–4691 (2026). `https://doi.org/10.1109/TPAMI.2025.3647952`
46. Zhou, X., Lin, Z., Shan, X., Wang, Y., Sun, D., Yang, M.H.: DrivingGaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 21634–21643 (2024)
47. Zhu, X., Yi, R., Wen, X., Zhu, C., Xu, K.: Relighting scenes with object insertions in neural radiance fields. IEEE Trans. Circuit Syst. Video Technol. **35**(7), 6787–6802 (2025)