PADSIM: A PHOTOREALISTIC CLOSED-LOOP SIMULATOR WITH CONTROLLABLE PEDESTRIANS FOR AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Closed-loop simulation is essential for evaluating autonomous driving systems (ADS) in safety-critical urban scenarios. However, existing simulators primarily focus on vehicle-centric behaviors and largely overlook the impact of pedestrian dynamics on decision-making. To address this limitation, we present **PADSIM**, a photorealistic closed-loop simulator that enables controllable and interaction-consistent pedestrian behaviors in dynamic environments. PADSIM supports flexible placement and motion control of pedestrians while ensuring physically plausible and collision-free interactions with surrounding agents. At the core of our system is an Interaction-Driven Motion Generation (IDMG) pipeline, which produces scene-aware pedestrian trajectories by explicitly modeling multi-agent interactions and incorporating physical constraints. To maintain visual realism, we further introduce a Global-Local Illumination-Consistent Synthesis (GLICS) strategy that preserves human–scene appearance consistency by leveraging environmental lighting cues without additional optimization or training. By tightly coupling motion generation, rendering, and policy inference in a recurrent loop, PADSIM enables reactive simulation beyond open-loop replay. In addition, we propose a system-level complexity metric to quantify the risk and difficulty of generated scenarios from the ego vehicle's perspective. Experiments on real-world datasets demonstrate that PADSIM supports realistic augmentation of pedestrian behaviors, facilitates policy-driven scenario evolution, and enables more comprehensive evaluation of autonomous driving systems under safety-critical human–vehicle interactions.

1 INTRODUCTION

Autonomous driving systems (ADS) are required to operate reliably and safely within complex and dynamic traffic environments, where pedestrian behavior frequently constitutes a critical factor in ego-vehicle decision-making. Although real-world evaluation provides high-fidelity data, it is costly, risky, and insufficient for covering rare safety-critical interactions Dosovitskiy et al. (2017); Li et al. (2022). Driving scenario simulation has therefore become a mainstream paradigm for scalable ADS evaluation. However, credible pedestrian-centered simulation requires more than visual realism: pedestrians must maintain consistent movement, appearance, and interaction with the surrounding 3D scene and traffic agents.

Existing simulation platforms still provide limited support for constructing safety-critical scenarios involving rich pedestrian interaction behaviors. Traditional simulators such as CARLA and MetaDrive Dosovitskiy et al. (2017); Li et al. (2022) offer controllable agents but lack the realism of reconstructed real-world scenes, while neural rendering-based simulators, including NeRF Mildenhall et al. (2021) and 3DGS Kerbl et al. (2023), improve visual fidelity but suffer from either low efficiency Yang et al. (2023); Ljungbergh et al. (2024) or limited support for complex multi-pedestrian dynamics Zhou et al. (2025); Jiang et al. (2025); Ge et al. (2025). As summarized in Table 1, existing systems exhibit clear trade-offs across key capabilities, and none provides a unified solution that balances realism, controllability, dynamic interaction, and efficiency. This limitation makes them insufficient for evaluating long-tail safety-critical scenarios involving diverse pedestrian behaviors.

Pedestrian-centric closed-loop simulation requires more than augmenting reconstructed 3D scenes with additional human agents; it demands that placed pedestrians be integrated as physically and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



Figure 1: Comparison of simulation results between PADSIM and HUGSIM. (a) Autonomous driving scenarios simulated by PADSIM, featuring both pedestrians and vehicles with consistent lighting; (b) scenarios generated by HUGSIM, containing only vehicles and exhibiting inconsistent lighting.

Table 1: Comparison of Autonomous Driving Simulation Platforms. We compare various features including closed-loop evaluation (C-L), multi-view image consistency (MV-C), photorealistic rendering (P-R), multi-person generation (MP-G), dynamic scene planning (DS-P), human motion driven (HM-D), sudden pedestrian (S-P) and real-time simulation (Real-Time) capabilities.

No.	Name	Rendering	C-L	P-R	MV-C	MP-G	DS-P	HM-D	S-P	Real-time
1	CARLA Dosovitskiy et al. (2017)	Virtual	✓	✗	✓	✗	✗	✗	✗	✓
2	MateDrive Li et al. (2022)	Virtual	✓	✗	✓	✗	✗	✗	✗	✓
3	NeuroNCAP Ljungbergh et al. (2024)	NeRF	✓	✓	✓	✗	✗	✗	✗	✗
4	UniSim Yang et al. (2023)	NeRF	✓	✓	✓	✗	✗	✗	✗	✗
5	DriveArena Yang et al. (2025)	Diffusion	✓	✓	✗	✗	✗	✗	✗	✗
6	NAVSIM v2 Cao et al. (2025)	3DGS	Pseudo	✓	✓	✗	✗	✗	✗	✓
7	HUGSIM Zhou et al. (2025)	3DGS	✓	✓	✓	✗	✗	✗	✗	✓
8	RealEngine Jiang et al. (2025)	3DGS	✓	✓	✓	✗	✗	✗	✗	✓
9	SceneCrafter Zhu et al. (2025)	3DGS	✓	✓	✓	✗	✗	✗	✗	✓
10	AlpaSim NVIDIA et al. (2025)	3DGS	✓	✓	✓	✗	✗	✗	✗	✓
11	**OURS**	**3DGS**	✓	✓	✓	✓	✓	✓	✓	✓

visually coherent participants in the evolving traffic environment. This entails jointly modeling interaction-aware motion, which respects walkable regions and surrounding agents, and illumination-consistent appearance, which aligns placed pedestrians with scene geometry and lighting. Existing methods fail to guarantee motion compatibility and illumination consistency in dynamic scenarios, which motivates the development of PADSIM, as illustrated in Figure 1. While some of these aspects have been studied in isolation, they are not directly compatible with our target setting. For instance, RESCUE Liu et al. (2025) and PACER Rempe et al. (2023) can generate plausible crowd motion in environments with static obstacles, but they lack the ability to adapt to the highly dynamic and interactive nature of autonomous driving scenarios. Existing relighting methods are limited by their inability to operate directly on 3D representations, lack of end-to-end reasoning, and high computational cost, which hinder their integration into the simulator.

We propose **PADSIM**, a real-time 3DGS-based closed-loop simulation framework for autonomous driving that models pedestrian-induced effects. Given a reconstructed dynamic scene, configurable pedestrian behaviors, and a driving policy, PADSIM generates physically and visually realistic simulation sequences with dynamic scenario evolution. It contains an **Interaction-Driven Motion Generation (IDMG)** pipeline for controllable and scene-aware pedestrian trajectories and a **Global-Local Illumination-Consistent Synthesis (GLICS)** strategy for appearance consistency. Our system models pedestrian interactions to drive motion consistent with the dynamic environment, while environmental lighting is leveraged to estimate pedestrian lighting attribute without additional optimization or training, ensuring visual consistency with the scene. By coupling pedestrian evolution, rendering, and ego-policy inference in a closed-loop, PADSIM supports reactive simulation beyond open-loop replay.

The main contributions are as follows:

- We propose **PADSIM**, a 3DGS-based closed-loop simulation framework that enables dynamic pedestrian augmentation in dynamic scenes, facilitating human-centric autonomous driving evaluation and providing a foundation for analyzing the impact of pedestrian behaviors on AD decision-making.
- We introduce an **Interaction-Driven Motion Generation** pipeline to synthesize controllable pedestrian behaviors that respect walkable geometry and surrounding agents, enabling interactive scenarios beyond data replay.
- We propose a **Global-Local Illumination-Consistent Synthesis** strategy to improve human–scene appearance consistency for dynamic pedestrian placement under diverse illumination.
- We propose a **dynamic scenario complexity evaluation framework** to quantify the impact of pedestrian augmentation on safety-critical interactions from kinematic and risk perspectives.

2 Related Work

2.1 Closed-loop Simulation Platforms

Autonomous driving evaluation has evolved from open-loop benchmarks to closed-loop simulation. Open-loop methods such as UniAD Hu et al. (2023) evaluate perception, prediction, and planning on offline expert trajectories, but cannot capture feedback effects from ego-agent interactions. Classical simulators such as CARLA Dosovitskiy et al. (2017) and MetaDrive Li et al. (2022) provide controllable closed-loop environments, but rely on manually designed scenes that differ from real-world data. Recent neural rendering-based platforms, including UniSim Yang et al. (2023), NeuronCap Ljungbergh et al. (2024), HUGSIM Zhou et al. (2025), RealEngine Jiang et al. (2025), SceneCrafter Ge et al. (2025), AlpaSim Wang et al. (2025), and NAVSIM v2 Cao et al. (2025), improve scene realism using reconstructed or rendered real-world scenes. However, they provide limited support for controllable multi-pedestrian behaviors in dynamic closed-loop environments. Additionally, diffusion-based simulators including DrivingSphere Yan et al. (2025) and DriveArena Yang et al. (2025) suffer from inherent limitations in multi-view consistency. Our work targets scene-aware pedestrian augmentation for pedestrian-centric closed-loop simulation.

2.2 Autonomous Driving Scenario Synthesis

Autonomous driving scenario synthesis can be broadly divided into diffusion-based generation and 3D-driven editing. Diffusion-based methods such as DriveDreamer-2 Zhao et al. (2025) and MagicDrive-V2 Gao et al. (2025a) generate driving videos from text, trajectories, or layouts, but often suffer from limited multi-view consistency and physical plausibility. 3D-driven methods such as PACER+ Wang et al. (2024), ChatSim Wei et al. (2024b), ChatDyn Wei et al. (2024a), SIMSplat Park et al. (2025), and CRUISE Liu et al. (2025) offer stronger geometric consistency and controllability, yet still struggle to jointly support realistic rendering, efficient simulation, and flexible pedestrian-level editing. This limits their applicability to the pedestrian-centric closed-loop simulator.

3 Method

We present **PADSIM**, a physically grounded and photorealistic framework for *closed-loop simulation of coupled pedestrian-vehicle-background scenario in real-world driving scenes*. Existing simulation platforms typically emphasize these functional modules: they generate plausible vehicle trajectories, support realistic rendering, and enable safety evaluation, yet overlook controllable and scene-consistent pedestrian modeling within reconstructed real scenes. PADSIM is designed to jointly achieve: **(I) physically plausible and interaction-aware multi-pedestrian behaviors**, **(II) geometry- and illumination-consistent integration with the scene**, and **(III) online feedback between scene evolution and ego-vehicle control**.

As illustrated in Figure 2, PADSIM builds on a real-time unified 3DGS representation for autonomous driving and explicitly models pedestrian-induced dynamics. At each time step t, the dynamic scene is represented as

$$\begin{aligned} \mathcal{G}^t &= \left(\mathcal{G}_{\mathrm{s}}, \mathcal{G}_{\mathrm{p}}^t, \mathcal{G}_{\mathrm{o}}^t\right), \\ \mathcal{G}_{\mathrm{p}}^t &= \bigcup_i \Phi\left(\mathcal{A}_{\mathrm{p},i}, \Theta_i^t, \mathbf{T}_i^t\right), \quad \mathcal{A}_{\mathrm{p},i} \in \mathcal{A}_{\mathrm{p}}. \end{aligned} \tag{1}$$

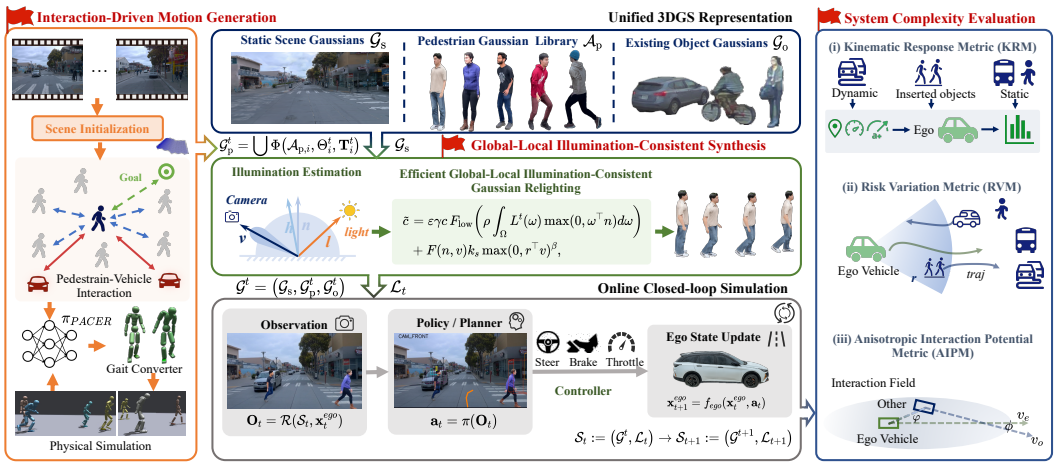

Figure 2: Overview of PADSIM. The framework generates interaction-aware pedestrian motion, converts the motion into dynamic Gaussian primitives, and integrates them with the static 3DGS scene. The unified scene is rendered into observations for the driving policy, whose control actions further affect subsequent pedestrian behaviors, forming a closed-loop simulation process.

Here, $\mathcal{G}_s$, $\mathcal{G}_p^t$, and $\mathcal{G}_o^t$ denote static scene Gaussians, time-varying pedestrian Gaussians, and existing-object Gaussians (e.g. vehicles), respectively. The pedestrian Gaussian library $\mathcal{A}_p$ is constructed from pedestrian videos following Moon et al. (2024). The pedestrian component $\mathcal{G}_p^t$ is obtained by selecting pedestrian Gaussian avatars $\mathcal{A}_{p,i}$ from this library and driving them with the generated pose Θ_i^t and global translation $\mathbf{T}_i^t$ through $\Phi(\cdot)$, while $\mathcal{G}_s$ and $\mathcal{G}_o^t$ are obtained from scene video reconstruction or collected scene data.

Built upon this representation, PADSIM generates physically plausible and visually consistent simulation sequences under evolving traffic scenarios. The framework consists of an **Interaction-Driven Motion Generation (IDMG)** pipeline, which produces controllable and scene-aware pedestrian trajectories and determines the temporal evolution of $\mathcal{G}_p^t$; a **Global-Local Illumination-Consistent Synthesis (GLICS)** strategy, which preserves appearance consistency across dynamic views; and an online closed-loop simulation process, which jointly incorporates dynamic scene states and ego-vehicle feedback for validating autonomous driving policies and planners. System-level complexity metrics are further used to quantify kinematic responses, risk evolution, and dynamic risk distribution.

3.1 Interaction-Driven Motion Generation

This module generates pedestrian motion that is both goal-directed and reactive to surrounding traffic, and defines the temporal evolution of $\mathcal{G}_p^t$.

Scene initialization. Given an input driving scene video, we first reconstruct the walkable ground geometry to support physically grounded pedestrian insertion. Specifically, we use Mask2Former Cheng et al. (2022) to extract road regions and RoMe Mei et al. (2024) to reconstruct a geometrically consistent ground mesh, which is transformed into a unified world coordinate system using camera intrinsics and extrinsics. Pedestrians are then initialized by sampling spawn locations $\mathbf{x}_i^0$ and goal locations $\mathbf{x}_i^{goal}$ on the reconstructed walkable surface.

Interaction-aware motion model. RESCUE Liu et al. (2025) fails to comprehensively capture the complex interaction distributions between pedestrians and agents in dynamic scenes. It only accounts for the influence of static obstacles and other generated pedestrians, while neglecting the intricacies of dynamic traffic flow. In contrast, our approach explicitly models the interaction between generated pedestrians and other traffic agents, and leverages the interaction as control signal to drive pedestrian motion. At each timestep t, the control signal of pedestrian i is defined as

$$\mathbf{u}_i^t = \mathbf{f}_i^{goal} + \sum_{k \neq i} \mathbf{f}_{ik}^{ped} + \sum_j \mathbf{f}_{ij}^{veh}, \tag{2}$$

where $\mathbf{f}_i^{goal}$ is the goal-directed driving force, $\mathbf{f}_{ik}^{ped}$ models interactions between pedestrian i and other pedestrian k, and $\mathbf{f}_{ij}^{veh}$ captures interactions between pedestrian i and vehicle j. The formulations of

$\mathbf{f}_i^{goal}$ and $\mathbf{f}_{ik}^{ped}$ follow standard and extended social force models Helbing & Molnar (1995); Helbing et al. (2000); Liu et al. (2025) and are provided in the Appendix B.2. For interactions with vehicles, we instead approximate each vehicle as a dynamic circular agent and model the interaction as an exponential repulsive force:

$$\mathbf{f}_{ij}^{veh} = A_i \exp\left(\frac{\lambda_{veh}(r_i + r_j) - d_{ij}}{B_i}\right) \mathbf{n}_{ij}, \tag{3}$$

where A_i controls the interaction magnitude, B_i the spatial decay, r_i and r_j the effective radii, $d_{ij} = \|\mathbf{x}_i^t - \mathbf{x}_j^t\|_2$, and $\mathbf{n}_{ij} = (\mathbf{x}_i^t - \mathbf{x}_j^t)/d_{ij}$ is the unit direction vector from vehicle j to pedestrian i. We use a larger λ_{veh} than in pedestrian–pedestrian interactions to account for stronger and earlier avoidance of vehicles.

Intent-preserving collision avoidance. To preserve interaction intent while enforcing local safety, we combine the interaction-aware update with ORCA-based Van Den Berg et al. (2011) projection:

$$\hat{\mathbf{v}}_i^{t+1} = \Pi_{\mathcal{V}_i^{free}}\left(\mathbf{v}_i^t + \Delta t\, \mathbf{u}_i^t\right), \qquad \tau_{gs}^{i,t+1} = \tau_{gs}^{i,t} + \Delta t\, \hat{\mathbf{v}}_i^{t+1}, \tag{4}$$

where $\mathbf{v}_i^t$ denotes the current velocity of pedestrian i, $\Pi_{\mathcal{V}_i^{free}}(\cdot)$ projects the desired velocity onto the collision-free velocity set defined by ORCA, and $\tau_{gs}^{i,t}$ denotes the 2D trajectory point of pedestrian i at time t. This preserves interaction-aware behavioral intent while preventing short-horizon collisions.

From trajectories to dynamic Gaussians. We adopt an existing control policy π_{PACER} Rempe et al. (2023) to follow the 2D pedestrian trajectory τ_{gs}^i and drive a humanoid model with the kinematic structure of SMPL Loper et al. (2023). Conditioned on the current scene state $\mathcal{S}_t$ and the target trajectory τ_{gs}^i, the policy outputs the pedestrian action a_{p}^t, yielding SMPL-based motion sequences with body pose Θ_i^t and global translation $\mathbf{T}_i^t$. As defined in Eq. equation 1, these motions drive the selected pedestrian Gaussian avatar $\mathcal{A}_{\text{p},i}$ to form the dynamic pedestrian component $\mathcal{G}_{\text{p}}^t$, which is directly compatible with rendering, illumination modeling, and closed-loop scene simulation.

3.2 Global-Local Illumination-Consistent Synthesis

Realistic pedestrian insertion requires not only geometric alignment but also appearance consistency under scene illumination. Direct image-space compositing or per-view correction Liang et al. (2025); Gao et al. (2025b); He et al. (2025) may disturb the original illumination distribution and introduce inconsistent illumination variations across camera views. In addition, some works Zhu et al. (2025) generate scenes by separately rendering the background and inserted objects and then compositing them. This will disrupt the occlusion relationship between pedestrians and existing dynamic agents for our task. To resolve these issues, PADSIM performs illumination-aware synthesis directly in the Gaussian domain, enabling inserted pedestrians to adapt to the surrounding lighting, and generating motion-coordinated and illumination-consistent 3DGS-based 3D scenarios before multi-view rendering.

Scene illumination estimation. We estimate the frame-level illumination state from the reconstructed 3DGS scene using the spherical harmonics (SH) of order $L = 3$. At time step t, the directional illumination is approximated as

$$L^t(\omega) = \sum_{l=0}^{L} \sum_{m=-l}^{l} c_{lm}^t Y_{lm}(\omega), \tag{5}$$

where Y_{lm} denotes the real-valued SH basis function for direction ω, and c_{lm}^t is the corresponding lighting coefficient. The coefficients are obtained by weighted least-squares fitting to approximate the scene radiance distribution. We define the frame-level illumination state as

$$\mathcal{L}^t = \left\{ c_{lm}^t \mid 0 \le l \le L,\ -l \le m \le l \right\}, \tag{6}$$

which summarizes the overall lighting condition of the current Gaussian scene and provides a shared illumination prior for all components in $\mathcal{G}^t$.

Efficient global-local illumination-consistent gaussian relighting. Our model combines a Lambertian diffuse term for global illumination and a view-dependent specular term for local appearance details. Inspired by the classic Lambertian model, we compute diffuse reflection from surface normals and the estimated incident illumination. A low-frequency filtering operation is further introduced to

improve robustness and suppress noise from sampling and normal estimation. Since the diffuse-only branch tends to produce overly uniform appearances and cannot capture view-dependent effects, we additionally adopt a Fresnel-based specular term to model local view variations. For each pedestrian Gaussian, we estimate a local surface normal using weighted PCA with outward-facing correction. Given the estimated normal n and the frame-level illumination state $\mathcal{L}^t$, we relight the Gaussian color by combining low-frequency diffuse irradiance and view-dependent specular reflection:

$$\tilde{c} = \varepsilon\gamma c\, F_{\text{low}} \left(\rho \int_{\Omega} L^t(\omega) \max(0, \omega^\top n) d\omega \right) + F(n, v) k_s \max(0, r^\top v)^\beta, \tag{7}$$

where c is the original avatar color, ρ is the diffuse albedo, ε and γ control exposure and environmental illumination intensity, and F_{low} suppresses high-frequency illumination noise. The low-frequency irradiance term serves as a shadow-aware illumination cue by attenuating pedestrian Gaussians in weakly illuminated regions, while the specular term models view-dependent highlights. The reflection direction is defined as $r = 2(n^\top \omega)n - \omega$, and $F(n, v) = F_0 + (1 - F_0)(1 - n^\top v)^5$ denotes the Fresnel term with base reflectance F_0. By applying relighting before rendering, PADSIM produces illumination-consistent pedestrian insertions across viewpoints without requiring per-image appearance correction.

3.3 Online Closed-Loop Simulation

PADSIM couples pedestrian generation, illumination-consistent synthesis, and autonomous driving policy evaluation into a recurrent simulation process. At time step t, the scene state is defined as $\mathcal{S}_t := (\mathcal{G}^t, \mathcal{L}_t)$. The ego-vehicle state is denoted by $\mathbf{x}_t^{ego}$, and the closed-loop simulation evolves as

$$(\mathcal{S}_t, \mathbf{x}_t^{ego}) \rightarrow \mathbf{O}_t \rightarrow \mathbf{a}_t \rightarrow (\mathcal{S}_{t+1}, \mathbf{x}_{t+1}^{ego}), \tag{8}$$

where the rendered observation $\mathbf{O}_t$ and the policy action $\mathbf{a}_t$ are computed as

$$\mathbf{O}_t = \mathcal{R}(\mathcal{S}_t, \mathbf{x}_t^{ego}), \quad \mathbf{a}_t = \pi(\mathbf{O}_t). \tag{9}$$

Here, $\mathcal{R}$ denotes the Gaussian renderer and π represents the autonomous driving policy. The predicted action, typically consisting of low-level control commands (e.g., steer, brake, and throttle), is then executed by the ego-vehicle dynamics to obtain the next ego state:

$$\mathbf{x}_{t+1}^{ego} = f_{ego}(\mathbf{x}_t^{ego}, \mathbf{a}_t), \tag{10}$$

where f_{ego} denotes the ego-vehicle dynamics. The next Gaussian scene and scene state are formulated as

$$\mathcal{G}^{t+1} = \left(\mathcal{G}_{\text{s}}, \mathcal{G}_{\text{p}}^{t+1}, \mathcal{G}_{\text{o}}^{t+1}\right), \qquad \mathcal{S}_{t+1} := \left(\mathcal{G}^{t+1}, \mathcal{L}_{t+1}\right), \tag{11}$$

where $\mathcal{G}_{\text{p}}^{t+1}$ is generated by the interaction-driven motion generation module in Section 3.1, $\mathcal{G}_{\text{o}}^{t+1}$ follows the underlying traffic dynamics and $\mathcal{L}_{t+1}$ is estimated by the illumination-consistent synthesis module in Section 3.2. In this recurrent process, ego actions influence subsequent pedestrian interactions, while the updated dynamic scene changes future observations and planning decisions, enabling closed-loop validation beyond open-loop replay.

3.4 Scenario Complexity Metrics

To quantitatively evaluate the complexity of generated scenarios, we use three complementary metrics: the Kinematic Response Metric (KRM), the Risk Variation Metric (RVM), and the Anisotropic Interaction Potential Metric (AIPM). Their weighted combination defines the overall scenario complexity at time t:

$$Q(t) = \alpha_k Q_k(t) + \alpha_r Q_r(t) + \alpha_p Q_p(t), \tag{12}$$

where $Q_k(t)$, $Q_r(t)$, and $Q_p(t)$ denote KRM, RVM, and AIPM, respectively, and α_k, α_r, and α_p balance their contributions. We report the sequence-level score by averaging $Q(t)$ over all frames.

Kinematic Response Metric (KRM). KRM measures local motion intensity and proximity-induced response of the ego vehicle to nearby dynamic and static objects:

$$Q_k(t) = \sum_{i \in \mathcal{N}_d} \left(\frac{|\Delta v_i(t)|}{v_{lim}} + \beta_k \frac{|a_i(t)|}{a_{lim}} + \frac{d_{safe}^d - d_i}{d_{safe}^d} \right) + \sum_{i \in \mathcal{N}_s} \frac{d_{safe}^s - d_i}{d_{safe}^s}, \tag{13}$$

where $\mathcal{N}_d$ and $\mathcal{N}_s$ denote the sets of dynamic and static objects within the ego safety range, d_i is the distance from the ego vehicle to object i, d_{safe}^d and d_{safe}^s are the corresponding safety distances, v_{lim} and a_{lim} are the maximum velocity and acceleration limits, and β_k is a weighting coefficient.

Table 2: Comparison of the ACR and WACR between our method and RESCUE in different scenarios.

Metrics	Methods	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	**Avg.**
ACR	RESCUE	4.53	6.09	1.52	2.65	1.85	8.06	10.36	30.92	4.29	4.90	4.16	7.21
	PADSIM	**2.76**	**2.04**	**0.61**	**0.82**	**0.00**	**2.76**	**6.60**	**0.71**	**2.35**	**2.65**	**1.42**	**2.25**
WACR	RESCUE	3.74	4.91	1.28	2.20	1.51	6.69	8.47	25.06	3.51	4.00	3.37	5.89
	PADSIM	**2.33**	**1.71**	**0.53**	**0.69**	**0.00**	**2.33**	**5.38**	**0.61**	**1.94**	**2.16**	**1.18**	**1.71**

Risk Variation Metric (RVM). RVM captures temporal escalation of collision risk by measuring changes in inverse time-to-collision:

$$Q_r(t) = \sum_{i \in \mathcal{N}_d} \left(\frac{1}{TTC_i(t)} - \frac{1}{TTC_i(t - \Delta t)} \right), \tag{14}$$

where $TTC_i(t)$ denotes the time-to-collision between the ego vehicle and dynamic object i at time t.

Anisotropic Interaction Potential Metric (AIPM). AIPM introduces an anisotropic interaction potential field to describe the distribution of dynamic risks around the ego vehicle.:

$$Q_p(t) = \sum_{i \in \mathcal{N}_d} \eta_{dir,i}(t) \exp\left(-r_i^\top(t) \Sigma_i^{-1}(t) r_i(t)\right) \left[1 + k_v \left(\frac{v_i^{rel}(t)}{v_0} \right) \right], \tag{15}$$

where $r_i(t)$ is the relative position between the ego vehicle and object i, $\Sigma_i(t)$ is the anisotropic interaction covariance matrix, $\eta_{dir,i}(t)$ is a directional weighting term, $v_i^{rel}(t)$ is the relative velocity, v_0 is a normalization constant, and k_v controls the velocity-dependent contribution.

Taken together, KRM, RVM, and AIPM characterize complementary aspects of scenario complexity, including local kinematic intensity, temporal risk escalation, and directional interaction concentration.

4 Experiments

4.1 Baselines and Metrics

We summarize the following four aspects of experiments. **(I) Pedestrian motion generation evaluation (Section 4.2)**: we compare the proposed method with state-of-the-art RESCUE in terms of the Average Collision Rate (ACR) and Weighted Average Collision Rate (WACR). **(II) Lighting consistency evaluation (Section 4.3)**: since there is no appropriate comparison method at present, we adopt a groundtruth (GT)-free evaluation method and use three metrics, namely PAA-Y, APF-Y, and NCM-ab Mir et al. (2026), to quantify the lighting consistency of the human and the surrounding scenario before and after relighting. **(III) Autonomous driving algorithm evaluation (Appendix E.3)**: following the evaluation protocols of HUGSIM Zhou et al. (2025), we adopt six metrics including NC, DAC, TTC, COM, Rc, and HD-Score to evaluate three end-to-end autonomous driving algorithms (UniAD Hu et al. (2023), VAD Jiang et al. (2023), LTF Chitta et al. (2022)) and one Vision-Language-Action (VLA)-based driving algorithm Alpamayo-R1 Wang et al. (2025). **(IV) System complexity evaluation (Section 4.4)**: we use the three proposed metrics (KRM, RVM, and AIPM) to evaluate the complexity of PADSIM and RESCUE. More experimental setups are presented in Appendix C, and detailed definitions and formulations are provided in Appendix D.

4.2 Pedestrian Motion Generation Evaluation

Quantitative Analysis. Table 2 quantitatively compares the multi-pedestrian trajectory planning performance of our method and RESCUE across dynamic scenarios. We used eleven representative scenarios to conduct a comprehensive evaluation of both methods. Experimental results demonstrate that our method achieves superior dynamic obstacle avoidance capability over RESCUE, in terms of both scene-wise individual metrics and overall average performance. The RESCUE only focuses on the trajectory rationality among generated agents and collision avoidance with static obstacles. It fails to properly cope with coupled dynamic and static obstacles in complex scenarios, leading to a consistently higher collision rate than our proposed method.

Qualitative Analysis. To further intuitively demonstrate the superiority of our method, we visualize the multi-pedestrian planned trajectories in Figure 3. In this figure, black rectangles denote dynamic and static obstacles within the scene. The yellow curves illustrate the historical trajectories over the past 15 frames and future trajectories over the subsequent 30 frames of these obstacles. Similarly, the

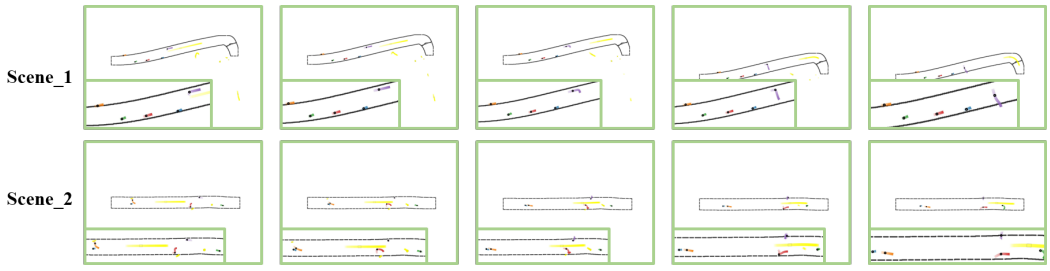

Figure 3: The visualized results of multiple pedestrian trajectories generated in different scenarios.



Figure 4: Visualization of relighting results. (a) rendering results without relighting; (b) rendering results after relighting.

circular markers represent planned pedestrians, and the colored curves depict their respective motion trajectories across the same time span. For each scene sequence, we uniformly sample five frames of planning results for visualization in Figure 3. Furthermore, we also present the visual results of SMPL motion sequences and animated human avatars in Appendix E.1.

Ablation of Motion Generation. As shown in Table 2, removing the dynamic planning mechanism and substituting our approach with the trajectory planning strategy in RESCUE leads to a noticeable increase in the collision rate between pedestrians and other dynamic traffic participants. Experimental results validate that our trajectory generation module serves as a crucial guarantee for maintaining behavioral and contextual compatibility between synthetic pedestrians and other dynamic agents in the scene. In addition, Figure 5 (a) also shows that when our trajectory generation module is removed, obvious penetration occurs in the scene.

4.3 Illumination Consistency Evaluation

Quantitative Analysis. Table 3 reports the quantitative evaluation results of our proposed relighting method across different scenarios. **Orig** denotes the illumination consistency between the human avatar and the original scene, while **Relit** corresponds to that after relighting optimization. Δ indicates the performance gap between **Orig** and **Relit**. Experiment results demonstrate that our relighting method effectively improves the illumination consistency between the human avatar and scenes.

Table 3: The evaluation results of lighting consistency for the proposed relighting method.

	Scene_1			**Scene_2**			**Scene_3**		
	PAA-Y↓	APF-Y↓	NCM-ab↓	PAA-Y↓	APF-Y↓	NCM-ab↓	PAA-Y↓	APF-Y↓	NCM-ab↓
Orig	5.140	0.1184	18.5482	7.3055	0.1015	44.3082	2.6103	0.1119	26.4196
Relit	4.6519	0.0616	14.8736	6.8187	0.0742	26.2716	2.1928	0.0678	13.7464
$\Delta\uparrow$	0.4877	0.0568	3.6746	0.4868	0.0273	18.0366	0.4175	0.0509	12.6732

Qualitative Analysis. To intuitively demonstrate the performance of our relighting method, we adopt different environment maps to relight the same human avatar, as illustrated in Figure 4. Experimental results reveal that despite its structural simplicity, the proposed method can adapt well to diverse illumination conditions. Although there remains room for further improvement in handling subtle illumination variations, the current lightweight version is sufficiently effective for the requirements of autonomous driving evaluation systems.

Table 4: System complexity evaluation results.

KRM			RVM			AIPM			$\mathbf{Q}(\alpha_k, \alpha_r, \alpha_p = 1)$		
Dataset	RESCUE	**PADSIM**	Dataset	RESCUE	**PADSIM**	Dataset	RESCUE	**PADSIM**	Dataset	RESCUE	**PADSIM**
3.0020	3.4462	**3.9649**	1.4969	1.9359	**2.0317**	0.3261	1.9383	**2.7691**	4.8250	7.3204	**8.7657**



Figure 5: Ablation study. (a) Ablation of interaction-driven motion generation; (b) ablation of global-local illumination-consistent synthesis.

Ablation of Illumination Consistency. With our proposed human relighting method, the illumination consistency between the pedestrians and the surrounding environment is significantly improved. As shown in Figure 4 and Figure 5 (b), the inserted pedestrians blend naturally into the background scene. By contrast, without relighting processing, these pedestrians retain their original lighting characteristics. This unnatural inconsistency severely disturbs the perception performance of autonomous driving algorithms and greatly degrades the evaluation reliability of the simulation platform.

4.4 System Complexity Evaluation

Table 4 reports the averaged system complexity over twenty scenes, including the original dataset and the augmented scenarios generated by PADSIM and RESCUE. To provide a more meaningful comparison, we replace HUGSIM with RESCUE as the baseline and evaluate the scenario complexity after augmenting the original dynamic scenes with newly generated pedestrians. Unlike HUGSIM, which removes the original dynamic agents and therefore naturally results in substantially lower system complexity, PADSIM and RESCUE preserves the original dynamic agents. The results of the system complexity of HUGSIM are presented in Appendix E.2. Larger values of KRM, RVM, and AIPM correspond to higher scenario complexity. The increased system complexity further indicates that PADSIM can generate more challenging and risk-intensive scenarios for closed-loop autonomous driving evaluation. Compared with RESCUE, PADSIM introduces stronger interactions between the newly generated pedestrians, the ego vehicle, and the existing traffic participants, resulting in more complex driving situations without removing the original scene dynamics. This conclusion is consistent with the evaluation results in Appendix E.3. Due to the higher complexity and risk level of the synthesized scenarios, autonomous driving algorithms generally exhibit degraded evaluation performance on our simulation platform, further demonstrating the effectiveness of PADSIM in constructing challenging scenarios for closed-loop evaluation.

5 Conclusion

We propose PADSIM, a 3DGS-based closed-loop simulator that integrates both physical and visual realism for autonomous driving simulation. To achieve motion compatibility and illumination consistency between dynamic scenes and multiple pedestrians, we adopt an intention-driven motion generation pipeline and an illumination-consistent pedestrian-scene synthesis strategy. Subsequently, we establish a system-level complexity evaluation framework to evaluate the complexity and risk level of the generated scenarios. Finally, we conducted four sets of experiments, namely motion generation evaluation, illumination consistency evaluation, autonomous driving algorithm evaluation, and system complexity evaluation. The experimental results demonstrate that the scenarios generated by the proposed PADSIM are realistic and rationally constructed, achieve notably higher complexity than existing datasets, and enable flexible modeling of pedestrian impacts on autonomous driving decision-making. This is of profound significance for precise autonomous driving scenario simulation as well as enhancing the reliability and credibility of autonomous driving algorithm evaluation.

AI USE STATEMENT

In this work, we used generative AI tools to polish the manuscript, obtain suggestions for organizing and presenting our writing, and retrieve and discover relevant literature. During the research process, we used AI to identify relevant work as an important reference. After completing the manuscript, we also used AI to correct grammatical errors and improve the clarity and fluency of the writing. We did not use generative AI tools to draft sections of the paper or to conceive or conduct the research. Generating synthetic datasets and proving mathematical claims are not applicable to this work. Two authors reviewed all AI-assisted content for accuracy. We take responsibility for the final content of this work, including any text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, et al. Pseudo-simulation for autonomous driving. *arXiv preprint arXiv:2506.04218*, 2025.

Hongze Chen, Zehong Lin, and Jun Zhang. Gi-gs: Global illumination decomposition on gaussian splatting for inverse rendering. *arXiv preprint arXiv:2410.02619*, 2024.

Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1290–1299, 2022.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE transactions on pattern analysis and machine intelligence*, 45(11):12878–12895, 2022.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pp. 1–16. PMLR, 2017.

Kang Du, Zhihao Liang, Yulin Shen, and Zeyu Wang. Gs-id: Illumination decomposition on gaussian splatting via adaptive light aggregation and diffusion-guided material priors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 26220–26229, 2025.

Ruiyuan Gao, Kai Chen, Bo Xiao, Lanqing Hong, Zhenguo Li, and Qiang Xu. Magicdrive-v2: High-resolution long video generation for autonomous driving with adaptive control. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 28135–28144, 2025a.

Wenshuo Gao, Junyi Fan, Jiangyue Zeng, and Shuai Yang. Flowportal: Residual-corrected flow for training-free video relighting and background replacement. *arXiv preprint arXiv:2511.18346*, 2025b.

Junhao Ge, Zuhong Liu, Longteng Fan, Yifan Jiang, Jiaqi Su, Yiming Li, Zhejun Zhang, and Siheng Chen. Unraveling the effects of synthetic data on end-to-end autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 28859–28869, 2025.

Kai He, Ruofan Liang, Jacob Munkberg, Jon Hasselgren, Nandita Vijaykumar, Alexander Keller, Sanja Fidler, Igor Gilitschenski, Zan Gojcic, and Zian Wang. Unirelight: Learning joint decomposition and synthesis for video relighting. *arXiv preprint arXiv:2506.15673*, 2025.

Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51 (5):4282, 1995.

Dirk Helbing, Illés Farkas, and Tamas Vicsek. Simulating dynamical features of escape panic. *Nature*, 407(6803):487–490, 2000.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 17853–17862, 2023.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8350, 2023.

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

Wei Jiang, Kwang Moo Yi, Golnoosh Samei, Oncel Tuzel, and Anurag Ranjan. Neuman: Neural human radiance field from a single video. In *European Conference on Computer Vision*, pp. 402–418. Springer, 2022.

Napat Karnchanachari, Dimitris Geromichalos, Kok Seang Tan, Nanxiang Li, Christopher Eriksen, Shakiba Yaghoubi, Noushin Mehdipour, Gianmarco Bernasconi, Whye Kit Fong, Yiluan Guo, et al. Towards learning-based planning: The nuplan benchmark for real-world autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 629–636. IEEE, 2024.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*, 45(3):3461–3475, 2022.

Ruofan Liang, Zan Gojcic, Huan Ling, Jacob Munkberg, Jon Hasselgren, Chih-Hao Lin, Jun Gao, Alexander Keller, Nandita Vijaykumar, Sanja Fidler, et al. Diffusion renderer: Neural inverse and forward rendering with video diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 26069–26080, 2025.

Zhihao Liang, Qi Zhang, Ying Feng, Ying Shan, and Kui Jia. Gs-ir: 3d gaussian splatting for inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21644–21653, 2024.

Xiaolin Liu, Tianyi Zhou, Hongbo Kang, Jian Ma, Ziwen Wang, Jing Huang, Wenguo Weng, Yu-Kun Lai, and Kun Li. Rescue: Crowd evacuation simulation via controlling sdm-united characters. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 24955–24964, 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*, pp. 161–177. Springer, 2024.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pp. 851–866. 2023.

Ruohong Mei, Wei Sui, Jiaxin Zhang, Xue Qin, Gang Wang, Tao Peng, Tao Chen, and Cong Yang. Rome: Towards large scale road surface reconstruction via mesh representation. *IEEE Transactions on Intelligent Vehicles*, 9(7):5173–5185, 2024.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Aymen Mir, Riza Alp Guler, Jian Wang, Gerard Pons-Moll, and Bing Zhou. Animated 3dgs avatars in diverse scenes with consistent lighting and shadows. *arXiv preprint arXiv:2601.01660*, 2026.

Gyeongsik Moon, Takaaki Shiratori, and Shunsuke Saito. Expressive whole-body 3d gaussian avatar. In *European Conference on Computer Vision*, pp. 19–35. Springer, 2024.

NVIDIA, Yulong Cao, Riccardo de Lutio, Sanja Fidler, Guillermo Garcia Cobo, Zan Gojcic, Maximilian Igl, Boris Ivanovic, Peter Karkus, Janick Martinez Esturo, Marco Pavone, Aaron Smith, Ellie Tanimura, Michal Tyszkiewicz, Michael Watson, Qi Wu, and Le Zhang. Alpasim: A modular, lightweight, and data-driven research simulator for autonomous driving, October 2025. URL https://github.com/NVlabs/alpasim.

Sung-Yeon Park, Adam Lee, Juanwu Lu, Can Cui, Luyang Jiang, Rohit Gupta, Kyungtae Han, Ahmadreza Moradipari, and Ziran Wang. Simsplat: Predictive driving scene editing with language-aligned 4d gaussian splatting. *arXiv preprint arXiv:2510.02469*, 2025.

Davis Rempe, Zhengyi Luo, Xue Bin Peng, Ye Yuan, Kris Kitani, Karsten Kreis, Sanja Fidler, and Or Litany. Trace and pace: Controllable pedestrian animation via guided trajectory diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13756–13766, 2023.

Kaiyue Shen, Chen Guo, Manuel Kaufmann, Juan Jose Zarate, Julien Valentin, Jie Song, and Otmar Hilliges. X-avatar: Expressive human avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16911–16921, 2023.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2446–2454, 2020.

Jur Van Den Berg, Stephen J Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. In *Robotics research: the 14th international symposium ISRR*, pp. 3–19. Springer, 2011.

Jingbo Wang, Zhengyi Luo, Ye Yuan, Yixuan Li, and Bo Dai. Pacer+: On-demand pedestrian animation controller in driving scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 718–728, 2024.

Yan Wang, Wenjie Luo, Junjie Bai, Yulong Cao, Tong Che, Ke Chen, Yuxiao Chen, Jenna Diamond, Yifan Ding, Wenhao Ding, et al. Alpamayo-r1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail. *arXiv preprint arXiv:2511.00088*, 2025.

Yuxi Wei, Jingbo Wang, Yuwen Du, Dingju Wang, Liang Pan, Chenxin Xu, Yao Feng, Bo Dai, and Siheng Chen. Chatdyn: Language-driven multi-actor dynamics generation in street scenes. *arXiv preprint arXiv:2412.08685*, 2024a.

Yuxi Wei, Zi Wang, Yifan Lu, Chenxin Xu, Changxing Liu, Hao Zhao, Siheng Chen, and Yanfeng Wang. Editable scene simulation for autonomous driving via collaborative llm-agents. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15077–15087, 2024b.

Tianyi Yan, Dongming Wu, Wencheng Han, Junpeng Jiang, Xia Zhou, Kun Zhan, Cheng-zhong Xu, and Jianbing Shen. Drivingsphere: Building a high-fidelity 4d world for closed-loop simulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 27531–27541, 2025.

Xuemeng Yang, Licheng Wen, Tiantian Wei, Yukai Ma, Jianbiao Mei, Xin Li, Wenjie Lei, Daocheng Fu, Pinlong Cai, Min Dou, et al. Drivearena: A closed-loop generative simulation platform for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 26933–26943, 2025.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1389–1399, 2023.

Xiangwen Zhang, Qian Zhang, Longfei Han, Qiang Qu, and Xiaoming Chen. Accidentsim: Generating physically realistic vehicle collision videos from real-world accident reports. *arXiv preprint arXiv:2503.20654*, 2025.

Guosheng Zhao, Xiaofeng Wang, Zheng Zhu, Xinze Chen, Guan Huang, Xiaoyi Bao, and Xingang Wang. Drivedreamer-2: Llm-enhanced world models for diverse driving video generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 10412–10420, 2025.

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. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Zehao Zhu, Yuliang Zou, Chiyu Max Jiang, Bo Sun, Vincent Casser, Xiukun Huang, Jiahao Wang, Zhenpei Yang, Ruiqi Gao, Leonidas Guibas, et al. Scenecrafter: Controllable multi-view driving scene editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6812–6822, 2025.

APPENDIX CONTENTS

A Additional Related Work **14**

B Further Details of the Method **14**

B.1 Overview of the PADSIM Pipeline 14

B.2 Details of Goal-Directed and Pedestrian Interaction Forces 15

B.3 Added Pedestrian Height Estimation 15

C Experimental Setup **16**

D Detailed Evaluation Metrics **17**

D.1 Dynamic Multi-person Planning Metrics 17

D.2 Illumination Consistency Metrics 17

D.3 Closed-loop Autonomous Driving Metrics 18

E More Experiment Results **19**

E.1 Additional Motion Generation Results 19

E.2 System Complexity of HUGSIM 19

E.3 Autonomous Driving Algorithm Evaluation 19

E.4 Real-time Analysis of PADSIM 20

F Limitation Analysis **21**

A ADDITIONAL RELATED WORK

I will add a discussion of related work on relighting here. Maintaining appearance consistency is crucial for inserting dynamic pedestrians into reconstructed 3D scenes. Physics-based inverse rendering methods Liang et al. (2024); Chen et al. (2024); Du et al. (2025) estimate geometry, materials, and illumination for view-consistent relighting, but usually require disentangled scene properties or additional optimization. Diffusion-based relighting methods Liang et al. (2025); Gao et al. (2025b); He et al. (2025) provide flexible image-level editing, but do not directly support controllable and temporally consistent relighting on explicit 3D Gaussian representations. Existing relighting methods are therefore not well suited to Gaussian-based dynamic simulation, where inserted pedestrians must remain consistent with scene geometry, illumination, and temporal evolution.

B FURTHER DETAILS OF THE METHOD

B.1 OVERVIEW OF THE PADSIM PIPELINE

PADSIM comprises two stages: offline pedestrian asset and motion preparation (Algorithm 1) and online closed-loop autonomous driving simulation (Algorithm 2).The offline stage consists of four components: road surface fitting, pedestrian avatar preparation, interaction-aware trajectory planning, and physics-based motion generation and avatar animation. The online stage comprises dynamic scene construction, illumination-consistent rendering, ego-policy inference and state updates, and dynamic-agent updates and simulation logging.

B.2 Details of Goal-Directed and Pedestrian Interaction Forces

In this section, we provide the detailed formulations of the goal-directed force $\mathbf{f}_i^{goal}$ and the pedestrian–pedestrian interaction force $\mathbf{f}_{ik}^{ped}$ used in Eq. equation 2. Our formulation follows the social force model (SFM) Helbing & Molnar (1995); Helbing et al. (2000) with extensions to account for lateral avoidance behaviors in multi-agent scenarios.

Goal-directed force. Each pedestrian is driven toward its assigned goal with a preferred velocity. The goal-directed force is defined as

$$\mathbf{f}_i^{goal} = \frac{v_i^{des}\mathbf{e}_i^t - \mathbf{v}_i^t}{\tau_i}, \tag{16}$$

where $\mathbf{v}_i^t$ is the current velocity, v_i^{des} is the desired walking speed, and τ_i is a relaxation parameter controlling how quickly the pedestrian adapts to the desired velocity. The unit direction toward the goal is given by

$$\mathbf{e}_i^t = \frac{\mathbf{x}_i^{goal} - \mathbf{x}_i^t}{\|\mathbf{x}_i^{goal} - \mathbf{x}_i^t\|_2}. \tag{17}$$

Pedestrian–pedestrian interaction. The interaction between pedestrians consists of a radial repulsive component and a lateral evasive component:

$$\mathbf{f}_{ik}^{ped} = \mathbf{f}_{ik}^{rep} + \mathbf{f}_{ik}^{eva}. \tag{18}$$

The repulsive force prevents collisions and maintains interpersonal distance:

$$\mathbf{f}_{ik}^{rep} = A_i^{ped} \exp\left(\frac{r_i + r_k - d_{ik}^t}{B_i^{ped}}\right) \mathbf{n}_{ik}^t, \tag{19}$$

where A_i^{ped} controls the interaction strength, B_i^{ped} controls the spatial decay, r_i and r_k denote the effective radii of pedestrians, and $d_{ik}^t = \|\mathbf{x}_i^t - \mathbf{x}_k^t\|_2$ is the pairwise distance. The unit direction vector is defined as

$$\mathbf{n}_{ik}^t = \frac{\mathbf{x}_i^t - \mathbf{x}_k^t}{d_{ik}^t}. \tag{20}$$

In addition to radial repulsion, pedestrians exhibit lateral avoidance behaviors when encountering others within their forward perception region. We model this behavior using an evasive force acting perpendicular to the desired motion direction:

$$\mathbf{f}_{ik}^{eva} = \alpha_{ik}\mathbf{p}_i^t, \tag{21}$$

where $\mathbf{p}_i^t$ is a unit vector orthogonal to $\mathbf{e}_i^t$, and α_{ik} is an activation coefficient that depends on the relative configuration between pedestrians (e.g., whether pedestrian k lies within a forward interaction sector of pedestrian i).

This decomposition enables the model to capture both short-range collision avoidance and longer-range path adjustment behaviors, leading to smoother and more realistic multi-agent motion in dynamic environments.

B.3 Added Pedestrian Height Estimation

HUGSIM estimates the agent's ground position using the nearest camera pose and calibrated height. However, this camera-anchored approximation may cause floating or penetration on non-planar terrain and limits long-horizon, large-scale scene generation.

To improve placement accuracy on non-planar surfaces, we propose a terrain-aware grounding method that estimates the local supporting surface directly from semantic scene Gaussians. The comparison results are shown in Figure 6. We first extract the ground Gaussians

$$\mathcal{P}_g = \{\mathbf{p}_i = (x_i, y_i, z_i) \mid c_i = \text{ground}\} \tag{22}$$



Figure 6: Comparison of the results obtained using (a) HUGSIM's height estimation method and (b) PADSIM's height estimation method.

and organize them into a regular grid in the x-z plane. The representative elevation of each grid cell is computed as the median elevation of its contained ground Gaussians, which suppresses isolated outliers and semantic noise.

We also estimate the foot center robustly from the lowest 0.5% of the avatar Gaussians. Since the positive y-axis points downward in our coordinate system, the candidate foot region and its center are defined as

$$\mathcal{F} = \{\mathbf{p}_i \mid y_i \geq Q_{0.995}(\{y_j\})\}, \qquad \mathbf{p}_f = (x_f, y_f, z_f) = \text{median}_{\mathbf{p}_i \in \mathcal{F}}\, \mathbf{p}_i. \tag{23}$$

Given the nearby ground-grid samples $\mathcal{G}_f = \{(x_j, \bar{y}_j, z_j)\}_{j=1}^{N}$, we fit an actor-centered local plane $y = ax + bz + c$ using distance-weighted least squares:

$$(a, b, c) = \arg\min_{a,b,c} \sum_{j=1}^{N} \frac{(ax_j + bz_j + c - \bar{y}_j)^2}{\max((x_j - x_f)^2 + (z_j - z_f)^2, \epsilon^2)}, \qquad \hat{y}_g^{\text{local}} = ax_f + bz_f + c. \tag{24}$$

The local estimate is accepted only when the neighborhood contains sufficient ground observations and the fitted plane has a reasonable residual and slope. Otherwise, we revert to the original camera-based estimate:

$$\hat{y}_g = \begin{cases} \hat{y}_g^{\text{local}}, & N \geq N_{\min},\ e_{\text{RMSE}} \leq \tau_e,\ \sqrt{a^2 + b^2} \leq \tau_s, \\ \hat{y}_g^{\text{cam}}, & \text{otherwise.} \end{cases} \qquad y_i' = y_i + \hat{y}_g - \delta - y_f, \tag{25}$$

where δ is a small clearance offset that alleviates geometric penetration between the feet and the road surface. The same vertical translation is applied to all avatar Gaussians, thereby preserving the pedestrian's geometry, scale, and pose.

This hybrid formulation falls back to camera-based grounding when semantic observations are insufficient, while adapting pedestrian placement to local terrain when reliable geometry is available. The ground grid is constructed only once, resulting in negligible runtime overhead.

C Experimental Setup

We construct simulation assets from both human reconstruction and autonomous driving datasets. For human appearance assets, we use ExAvatar Moon et al. (2024) to reconstruct human 3DGS avatars from monocular videos collected from NeuMan Jiang et al. (2022), X-Humans Shen et al. (2023), and our own high-quality monocular captures. For autonomous driving scenes, we follow the scenario reconstruction pipeline of HUGSIM Zhou et al. (2025) to reconstruct static backgrounds and dynamic traffic participants from the Waymo dataset Sun et al. (2020). The reconstructed

static background, traffic participants, and human avatars are then used as reusable assets in our closed-loop simulator. During simulation, we adopt a Linear Quadratic Regulator (LQR) controller following Karnchanachari et al. (2024); Yang et al. (2025) to convert planned waypoints into low-level control commands. All experiments are conducted on an 8-GPU server equipped with NVIDIA RTX 4090 GPUs. The specific hyperparameter settings are presented in Table 7.

D Detailed Evaluation Metrics

In this section, we provide detailed definitions of the evaluation metrics used in the main paper. We evaluate the proposed simulator from three aspects: dynamic multi-person planning, illumination consistency, and closed-loop autonomous driving performance.

D.1 Dynamic Multi-person Planning Metrics

Following AccidentSim Zhang et al. (2025), we use the Average Collision Rate (ACR) to evaluate the quality of multi-pedestrian trajectory planning in dynamic scenes. ACR measures whether the generated pedestrian trajectories collide with the trajectories or bounding boxes of existing dynamic and static objects. Specifically, each generated pedestrian is represented as a circle centered at its planned trajectory point with a radius of r. A collision is counted if this circle overlaps with the bounding box of another object in the scene.

ACR is defined as:

$$ACR = \frac{\sum_{i=1}^{N_s} \sum_{j=1}^{N_f^i} N_{i,j}^{\text{coll}}}{\sum_{i=1}^{N_s} \left(N_p^i \cdot N_f^i \right)}, \tag{26}$$

where N_s denotes the total number of sequences, N_p^i denotes the number of generated pedestrians in the i-th sequence, and N_f^i denotes the number of frames in the i-th sequence. $N_{i,j}^{\text{coll}}$ represents the number of generated pedestrians that collide with existing objects in the j-th frame of the i-th sequence. A lower ACR indicates better dynamic obstacle avoidance capability.

To further distinguish independent collision events from prolonged contacts, we introduce the Weighted Average Collision Rate (WACR). Unlike ACR, which assigns the same weight to every collision frame, WACR assigns a full weight to the first frame of each continuous collision event and a lower weight to its subsequent consecutive collision frames. In this way, WACR reduces the repeated penalty caused by a single long-lasting contact while still preserving the influence of collision duration.

WACR is defined as:

$$WACR = \frac{\sum_{i=1}^{N_s} \sum_{j=1}^{N_f^i} \sum_{k=1}^{N_p^i} w_{i,j,k}^{coll}}{\sum_{i=1}^{N_s} \left(N_p^i \cdot N_f^i \right)}, \tag{27}$$

where $w_{i,j,k}^{coll}$ denotes the weighted collision indicator of the k-th generated pedestrian in the j-th frame of the i-th sequence. Specifically, $w_{i,j,k}^{coll} = 0$ if no collision occurs, $w_{i,j,k}^{coll} = 1$ if the frame is the first frame of a continuous collision event, and $w_{i,j,k}^{coll} = \lambda$ if the frame belongs to the subsequent consecutive frames of the same collision event. In our experiments, λ is set to 0.8. When a generated pedestrian collides with multiple objects in the same frame, the maximum collision weight is used to avoid repeated counting. A lower WACR indicates better dynamic obstacle avoidance capability with fewer independent collisions and less persistent collision behavior.

D.2 Illumination Consistency Metrics

Following DGSM Mir et al. (2026), we use three metrics to evaluate the illumination consistency between inserted human avatars and the surrounding scene: Probe-Avatar Agreement in Luminance (PAA-Y), Avatar Photometric Fit in Luminance (APF-Y), and Neighborhood Chromaticity Match (NCM-ab). We compute these metrics before and after relighting, and report the absolute reduction as the improvement brought by the relighting module. A larger reduction indicates better improvement in illumination consistency.

Probe-Avatar Agreement in Luminance. PAA-Y measures the L1 distance between the spherical harmonic coefficients of the pedestrian and those of the surrounding scene, reflecting the consistency of perceived illumination lobes:

$$\text{PAA-Y} = \left\| c^{\text{ped}} - c^{\text{scene}} \right\|_1 , \tag{28}$$

where c^{ped} and c^{scene} denote the spherical harmonic coefficients of the pedestrian and scene, respectively.

Avatar Photometric Fit in Luminance. APF-Y measures the average absolute error between the observed luminance of the pedestrian and the luminance predicted by the scene lighting:

$$\text{APF-Y} = \frac{1}{|P|} \sum_{p \in P} \left| Y_p - \hat{Y}_p \right| , \tag{29}$$

where P denotes the set of pedestrian Gaussians, Y_p is the observed luminance of Gaussian p, and $\hat{Y}_p$ is the predicted luminance under the estimated scene illumination.

Neighborhood Chromaticity Match. NCM-ab measures the 1-Wasserstein distance between the chromaticity distributions of the pedestrian and neighboring scene points in the CIE-Lab color space:

$$\text{NCM-ab} = \text{EMD} \left(P_{ab}^{\text{ped}}, P_{ab}^{\text{scene}} \right) , \tag{30}$$

where P_{ab}^{ped} and P_{ab}^{scene} denote the chromaticity distributions of the pedestrian and the surrounding scene points, respectively. EMD denotes the Earth Mover's Distance, which is equivalent to the 1-Wasserstein distance in this setting.

D.3 Closed-loop Autonomous Driving Metrics

Following HUGSIM Zhou et al. (2025), NAVSIM v2 Cao et al. (2025), and DriveArena Yang et al. (2025), we evaluate autonomous driving algorithms using Non-Collision (NC), Drivable Area Compliance (DAC), Time-To-Collision (TTC), Comfort (COM), Route Completion (Rc), and HUGSIM Driving Score (HD-Score).

Non-Collision. NC measures whether the ego vehicle avoids at-fault collisions with surrounding static or dynamic traffic participants. A higher NC score indicates safer driving behavior.

Drivable Area Compliance. DAC measures whether the ego vehicle stays within the drivable area. Following HUGSIM, we compute the ratio of the ego vehicle area located inside the ground point clouds. A violation is counted when this ratio is smaller than 0.3.

Time-To-Collision. TTC measures the estimated time required for two vehicles to collide if they continue moving at their current velocities along their current trajectories. A higher TTC score indicates lower collision risk.

Comfort. COM evaluates the driving smoothness and stability of the ego vehicle. It considers longitudinal acceleration, lateral acceleration, yaw rate, yaw acceleration, and longitudinal jerk. A higher COM score indicates more comfortable and stable driving behavior.

Route Completion. Rc measures the driving progress of the ego vehicle before encountering critical failures, including collisions, off-road driving, large deviations from the predefined route, or exceeding the time limit.

HUGSIM Driving Score. HD-Score provides an overall closed-loop evaluation of autonomous driving performance. At each timestamp, the score is defined as:

$$HDScore_t = \left(\prod_{m \in \{NC, COM\}} score_m \right) \times \left(\frac{\sum_{w \in \{TTC, COM\}} weight_w \times score_w}{\sum_{w \in \{TTC, COM\}} weight_w} \right) . \tag{31}$$

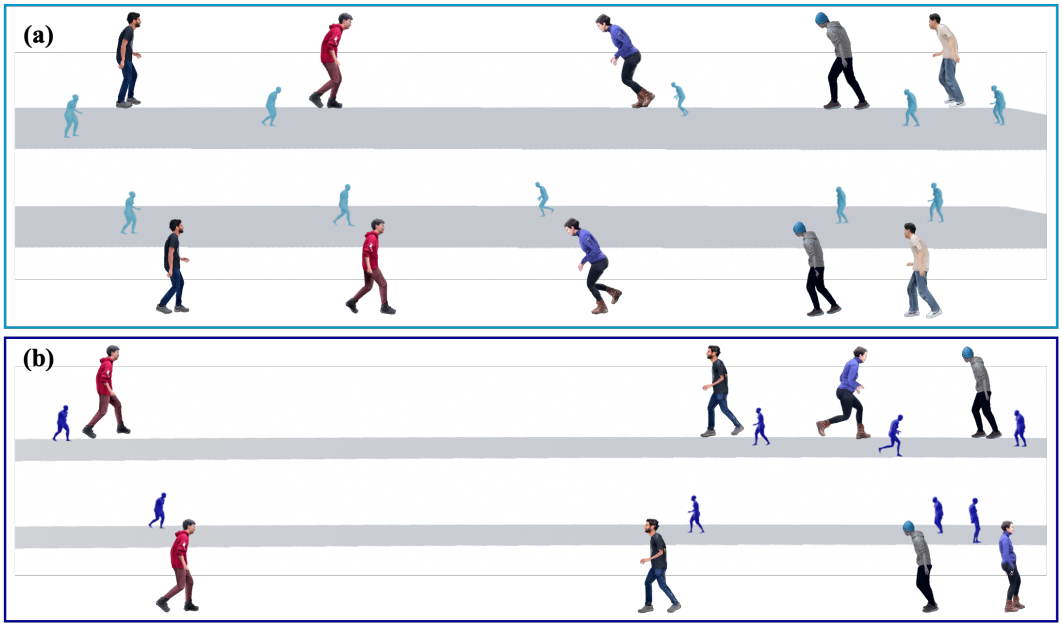

Figure 7: Additional visualization of generated SMPL motion sequences and animated human avatars. The gray region denotes the fitted ground area. (a) and (b) show the generated results in different scenarios.

The overall HDScore is then computed as:

$$HDScore = R_c \times \frac{\sum_{t=0}^{T} HDScore_t}{T}, \tag{32}$$

where R_c denotes the route completion score and T is the total number of evaluated timestamps.

E More Experiment Results

E.1 Additional Motion Generation Results

In this section, we provide additional qualitative results of the proposed motion generation pipeline. The planned pedestrian trajectories are first converted into physically plausible SMPL motion sequences through the physical simulator. Then, the generated SMPL sequences are used to animate reconstructed human avatars. As shown in Figure 7, our method can generate multiple pedestrian motions with behavioral diversity and physical feasibility under different traffic scenarios. Furthermore, we have presented more results of multi-pedestrian trajectory planning in Figure 8.

E.2 System Complexity of HUGSIM

Although HUGSIM is expected to exhibit relatively low system complexity because it removes the original dynamic agents from the scene, we still provide a quantitative evaluation on one representative scenario for completeness, as shown in Table 5. Compared with the original dataset, HUGSIM produces substantially lower KRM, RVM and AIPM values, indicating reduced system complexity after the removal of existing dynamic agents. This observation further supports our decision to use RESCUE instead of HUGSIM as the main baseline for system complexity evaluation, since RESCUE preserves the original dynamic content and therefore provides a more meaningful comparison with PADSIM.

E.3 Autonomous Driving Algorithm Evaluation

The synthetic scenes constructed in this work are ultimately designed for the evaluation of existing autonomous driving algorithms. Based on real autonomous driving datasets, we generate diverse

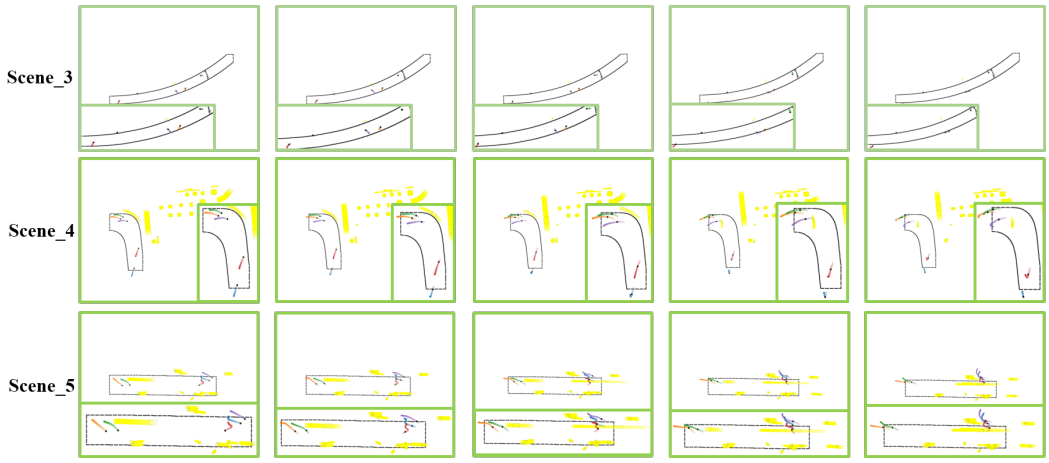

Figure 8: Additional visualization of generated multi-pedestrian trajectory planning results.

Table 5: System complexity evaluation results.

KRM			RVM			AIPM			Q ($\alpha_k, \alpha_r, \alpha_p = 1$)		
Dataset	HUGSIM	PADSIM	Dataset	HUGSIM	PADSIM	Dataset	HUGSIM	PADSIM	Dataset	HUGSIM	PADSIM
4.5828	0.6991	5.2418	2.513	0.0086	4.603	0.00035	0.0687	4.6935	7.0993	0.7764	14.5383

pedestrians and composite them into existing realistic scenes to achieve excellent compatibility with dynamic environment. Table 6 summarizes the quantitative evaluation results of different autonomous driving algorithms across multiple synthetic scenarios. As observed from the experimental result, autonomous driving algorithms achieve generally low overall evaluation scores on our simulation platform. Meanwhile, the autonomous vehicle travels only a very short distance within our constructed scenarios and soon collide with surrounding dynamic traffic participants. This phenomenon indicates that the generated pedestrians greatly interfere with the decision-making process of autonomous driving algorithms. Figure 9 shows the visualization results of RGB observations, perception results, and planning trajectories, predicted BEV information during the evaluation process of the UniAD. In addition, our simulation platform can also synthesize some specific scenarios, such as traffic police scenario and sudden pedestrian appearance scenario, as shown in Figure 10.

Table 6: Evaluation results of the AD algorithms on PADSIM across different scenarios.

Methods	Scene_1						Scene_2					
	NC↑	DAC↑	TTC↑	COM ↑	Rc↑	HD-Score↑	NC↑	DAC↑	TTC↑	COM↑	Rc↑	HD-Score↑
UniAD	1.0000	0.7273	0.7879	0.5152	0.5604	0.2669	1.0000	0.6667	0.7500	0.7917	0.1005	0.0443
VAD_base	0.6818	0.5909	0.4545	0.9545	0.6709	0.2396	0.4375	0.2500	0.4375	0.9375	0.1021	0.0064
VAD_tiny	0.4286	0.2143	0.4286	1.0000	0.5526	0.1184	0.5385	0.0000	0.5385	1.0000	0.1154	0.0000
LTF	1.0000	0.3571	1.0000	1.0000	0.4441	0.1586	1.0000	0.3333	1.0000	1.0000	0.0560	0.0187
AR1	1.0000	0.7222	1.0000	0.8889	0.4208	0.2905	1.0000	0.7797	0.9831	0.6441	1.0000	0.7046

E.4 Real-time Analysis of PADSIM

In our current pipeline, both pedestrian motion generation and Gaussian asset preparation—including human avatar driving and normal computation—are completed in advance and therefore do not introduce additional runtime overhead during closed-loop simulation. Compared with HUGSIM, the additional computational cost mainly comes from loading the pedestrian Gaussian assets (approximately **0.8s**), estimating the environmental illumination (approximately **0.45s**), relighting the human avatars (approximately **0.7s**), and image refinement (approximately **0.07s** per rendered image). Therefore, the additional computational overhead is limited, and PADSIM fully satisfies the real-time requirements of closed-loop simulation.

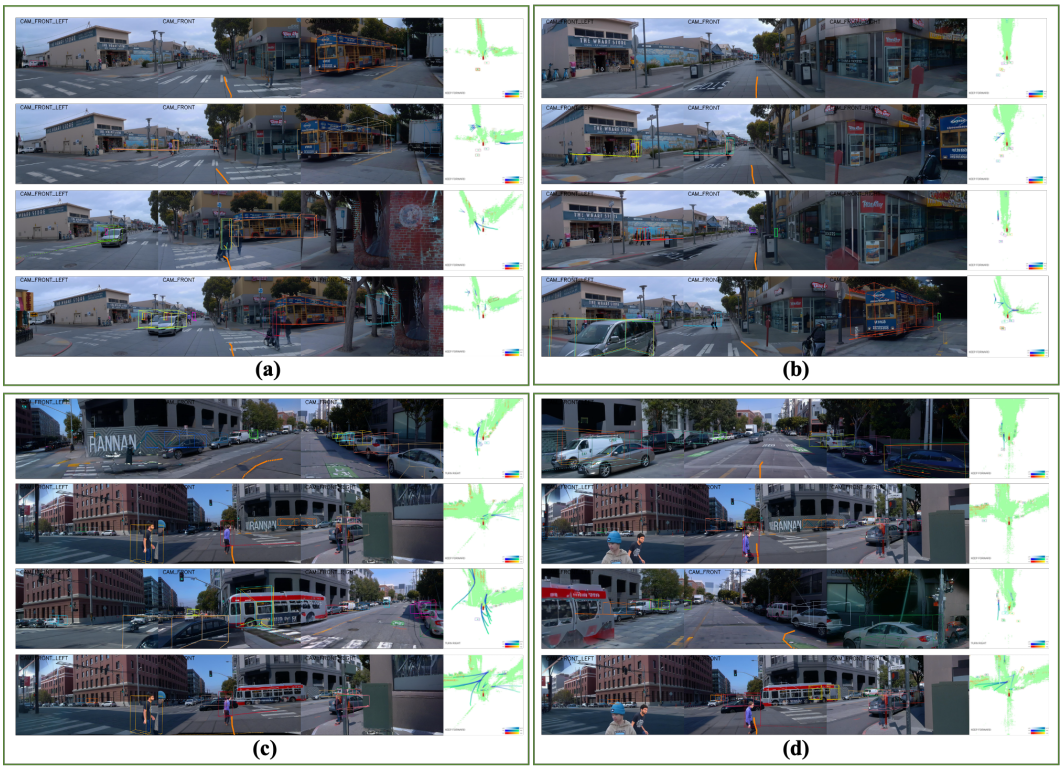

Figure 9: Visualization results of RGB observations, perception results, and planning trajectories, predicted BEV (Bird's Eye View) information during the UniAD evaluation process. In each subfigure, the four rows from top to bottom show: a scene without dynamic agents, a scene containing only our inserted pedestrians, a scene containing only the original dynamic agents, and a scene containing both the original dynamic agents and our inserted pedestrians.

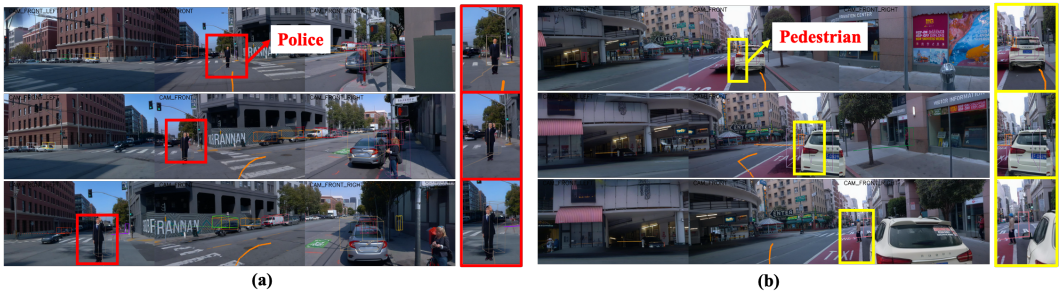

Figure 10: Special scenario display. (a) traffic police scenario; (b) sudden pedestrian appearance scenario.

F Limitation Analysis

Although we have innovatively incorporated pedestrians into a 3DGS-based closed-loop simulator for autonomous driving, a noticeable gap still exists between current closed-loop simulators and real-world autonomous driving scenarios. First, pedestrian behaviors remain relatively simple and lack interactive actions, such as gesturing toward ego vehicles. Second, social interactions among pedestrians are not modeled, e.g., conversations or greetings. Third, interactions among agents are still at a preliminary stage, where agents do not exhibit realistic decision-making or reasoning capabilities. This paper is merely our initial exploration of the closed-loop simulator for autonomous driving. We plan to further explore these challenges in future work.

Table 7: Hyperparameter settings for the proposed complexity metrics.

Metric	**Hyperparameter**	**Symbol**	**Value**
KRM	Dynamic safety distance	d^{d}_{safe}	$10.0\,\text{m}$
	Static safety distance	d^{s}_{safe}	$6.0\,\text{m}$
	Velocity limit	v_{lim}	$10.0\,\text{m/s}$
	Acceleration limit	a_{lim}	$4.0\,\text{m/s}^2$
	Acceleration weight	β_k	1.0
RVM	Maximum interaction distance	$d_{\max}$	$10.0\,\text{m}$
	Collision-distance threshold	d_{col}	$2.0\,\text{m}$
AIPM	Longitudinal scale	$\sigma_{\parallel}$	$2.0 + 0.4v_{\text{ego}}\,\text{m}$
	Lateral scale	$\sigma_{\perp}$	$1.0\,\text{m}$
	Velocity normalization	v_0	$2.0\,\text{m/s}$
	Velocity weight	k_v	1.0
	Maximum interaction distance	d_{neighbor}	$10.0\,\text{m}$
	Directional-mode weights	(α, β, γ)	$(0.30,\ 0.35,\ 0.35)$
Complexity	Metric fusion weights	$(\alpha_k, \alpha_r, \alpha_p)$	$(1,\ 1,\ 1)$

Algorithm 1 Offline Pedestrian Asset and Motion Preparation

Require: Driving-scene video $\mathcal{V}_s$; camera parameters $\{K, R, T\}$; human videos $\{\mathcal{V}_{p,i}\}$; static obstacles $\mathcal{B}_s$; dynamic-agent trajectories $\{\tau_j^{\text{obj}}\}$; number of pedestrians N_p; simulation step Δt

Ensure: Road surface mesh $\mathcal{M}_{\text{road}}$; pedestrian avatar library $\mathcal{A}_p$; planned pedestrian trajectories $\{\tau_i\}$; time-varying pedestrian Gaussians $\{G_p^{(t)}\}$; corresponding normals $\{N_p^{(t)}\}$

1: Road Surface Fitting

Extract the road mask $M_{\text{road}} \leftarrow \text{Mask2Former}(\mathcal{V}_s)$

Reconstruct the road surface mesh:

$$\mathcal{M}_{\text{road}} \leftarrow \text{RoMe}\left(\mathcal{V}_s, M_{\text{road}}, K, R, T\right)$$

Transform $\mathcal{M}_{\text{road}}$ into the unified world coordinate system

2: Pedestrian Avatar Preparation

for each human video $\mathcal{V}_{p,i}$ **do**
    Reconstruct the canonical pedestrian Gaussian avatar $A_{p,i} \leftarrow \text{ExAvatar}(\mathcal{V}_{p,i})$
    Precompute its Gaussian normals $N_{p,i} \leftarrow \text{EstimateNormals}(A_{p,i})$
    Add $(A_{p,i}, N_{p,i})$ to the avatar library $\mathcal{A}_p$
end for

3: Interaction-Aware Trajectory Planning

for each pedestrian $i = 1, \ldots, N_p$ **do**
    Sample the initial position $x_i^{(0)}$ and destination x_i^{goal} on $\mathcal{M}_{\text{road}}$
end for
for each planning step t **do**
    for each pedestrian i **do**
        Compute the interaction-aware control:

$$u_i^{(t)} \leftarrow f_i^{\text{goal}} + \sum_{k \neq i} f_{ik}^{\text{ped}} + \sum_j f_{ij}^{\text{veh}}$$

        Project the desired velocity onto the collision-free set:

$$\widehat{v}_i^{(t+1)} \leftarrow \Pi_{\mathcal{V}_i^{\text{free}}}\left(v_i^{(t)} + \Delta t\, u_i^{(t)}\right)$$

        Update the planned trajectory point $\tau_i^{(t+1)} \leftarrow \tau_i^{(t)} + \Delta t\, \widehat{v}_i^{(t+1)}$
    end for
end for

4: Physics-Based Motion and Avatar Animation

for each pedestrian i **do**
    Select a pedestrian avatar $(A_{p,i}, N_{p,i}) \in \mathcal{A}_p$
    Track τ_i using PACER in the physical simulator:

$$\{\Theta_i^{(t)}, T_i^{(t)}\}_{t=0}^{N_i} \leftarrow \text{PhysicalMotion}\left(\pi_{\text{PACER}}, \tau_i, \mathcal{M}_{\text{road}}\right)$$

    for each motion frame t **do**
        Animate the pedestrian Gaussian avatar $G_{p,i}^{(t)} \leftarrow \Phi(A_{p,i}, \Theta_i^{(t)}, T_i^{(t)})$
        Transform the precomputed normals $N_{p,i}^{(t)} \leftarrow \Psi(N_{p,i}, \Theta_i^{(t)}, T_i^{(t)})$
    end for
end for

Aggregate all animated pedestrians $G_p^{(t)} \leftarrow \bigcup_i G_{p,i}^{(t)}$

Aggregate their transformed normals $N_p^{(t)} \leftarrow \bigcup_i N_{p,i}^{(t)}$

Save $\mathcal{M}_{\text{road}}$, $\mathcal{A}_p$, $\{\tau_i\}$, $\{G_p^{(t)}\}$, and $\{N_p^{(t)}\}$

return $\mathcal{M}_{\text{road}}$, $\mathcal{A}_p$, $\{\tau_i\}$, $\{G_p^{(t)}\}$, and $\{N_p^{(t)}\}$

Algorithm 2 Online Closed-Loop Autonomous Driving Simulation

Require: Static scene Gaussians G_s; pedestrian avatar library $\mathcal{A}_p$; dynamic object Gaussians $G_o^{(0)}$; pedestrian states $\{\xi_i^{(0)}\}$; ego state $x_{\text{ego}}^{(0)}$; E2E driving policy π; pedestrian motion generator IDMG; object dynamics f_{obj}; cameras $\mathcal{C}$; simulation step Δt; maximum step $N_{\max}$

Ensure: Closed-loop records and metrics: NC, DAC, TTC, COM, Rc, and HD-Score

Load G_s, $G_o^{(0)}$, and pedestrian avatars $\mathcal{A}_p$
Initialize $t \leftarrow 0$ and launch the E2E driving policy π
while $t < N_{\max}$ and no terminal condition **do**
1: Dynamic Scene Construction
    Generate pedestrian states using interaction-driven motion:

$$\{\xi_i^{(t)}\}_i \leftarrow \text{IDMG}\left(\{\xi_i^{(t-1)}\}_i, G_o^{(t)}, x_{\text{ego}}^{(t)}, \Delta t\right)$$

    Animate selected pedestrian avatars:

$$G_p^{(t)} \leftarrow \bigcup_i \Phi\left(A_{p,i}, \Theta_i^{(t)}, T_i^{(t)}\right)$$

    Construct the dynamic Gaussian scene:

$$G^{(t)} \leftarrow G_s \cup G_p^{(t)} \cup G_o^{(t)}$$

2: Illumination-Consistent Rendering
    Estimate the frame-level SH illumination $L^{(t)} \leftarrow \text{EstimateIllumination}(G^{(t)})$
    Relight pedestrian Gaussians:

$$\widetilde{G}_p^{(t)} \leftarrow \text{GLICS}\left(G_p^{(t)}, L^{(t)}\right)$$

    Form the relit scene $\widetilde{G}^{(t)} \leftarrow G_s \cup \widetilde{G}_p^{(t)} \cup G_o^{(t)}$
    for each camera $c \in \mathcal{C}$ **do**
        Render observation $(I_{t,c}, D_{t,c}) \leftarrow \mathcal{R}(\widetilde{G}^{(t)}, x_{\text{ego}}^{(t)}, c)$
    end for
    Construct the multi-view observation $\mathcal{O}_t \leftarrow \{I_{t,c}, D_{t,c}\}_{c\in\mathcal{C}}$
3: Ego Policy Inference and State Update
    Predict the ego trajectory $\mathcal{T}_{\text{ego}}^{(t)} \leftarrow \pi(\mathcal{O}_t, \text{Command}_t)$
    if $\mathcal{T}_{\text{ego}}^{(t)}$ is invalid **then**
        break
    end if
    Convert the trajectory into controls $a_t \leftarrow \text{LQR}(\mathcal{T}_{\text{ego}}^{(t)}, x_{\text{ego}}^{(t)})$
    Update the ego state $x_{\text{ego}}^{(t+1)} \leftarrow f_{\text{ego}}(x_{\text{ego}}^{(t)}, a_t)$
4: Dynamic Agent Update and Logging
    Update dynamic object Gaussians $G_o^{(t+1)} \leftarrow f_{\text{obj}}(G_o^{(t)}, x_{\text{ego}}^{(t+1)}, \Delta t)$
    Compute route completion Rc_t and route deviation
    Record ego state, pedestrian and object states, planned trajectory, collisions, and Rc_t
    if collision, route deviation, or $Rc_t \geq 1$ **then**
        break
    end if
    $t \leftarrow t + 1$
end while
Calculate NC, DAC, TTC, and COM from simulation records
Calculate route completion $Rc \leftarrow \max_t(Rc_t)$
Calculate HD-Score using Eqs. (26)–(27)
Save rendered videos, simulation records, and evaluation results
return NC, DAC, TTC, COM, Rc, and HD-Score