Adversarial Generation and Collaborative Evolution of Safety-Critical Scenarios for Autonomous Vehicles

Anonymous Author(s)
Affiliation
Address
email

Abstract

The generation of safety-critical scenarios in simulation has become increasingly crucial for safety evaluation in autonomous vehicles (AV) prior to road deployment. However, current approaches largely rely on predefined threat patterns or rule-based strategies, which limit their ability to expose diverse and unforeseen failure modes. To overcome these, we propose SCENGE, a framework that exposes AV safety vulnerabilities by combining adversarial threat generation and collaborative trajectory evolution. Given a simple prompt of a benign scene, it first performs *Meta-Scenario Generation*, where a large language model (LLM), grounded in structured driving knowledge (*e.g.*, traffic regulations, real-world accident records), infers an adversarial agent whose behavior poses a threat to the ego vehicle. This agent is embedded into a meta-scenario specified in executable simulation code, supporting precise control over scene composition and agent dynamics. Subsequently, *Complex Scenario Evolution* augments the Meta-Scenario by introducing background vehicles with collaborative risky trajectories. It constructs an adversarial collaborator graph to identify key agents whose trajectories will be perturbed temporally and spatially, which will induce coordinated deviations in agent behavior and increase the likelihood of collision by intensifying interaction complexity. Extensive experiments conducted on multiple reinforcement learning (RL) based AV models show that SCENGE uncovers more severe collision cases (+31.96%) on average than SoTA baselines. Additionally, we validate the efficacy of SCENGE on large model based; we further observe that adversarial training on our scenarios improves the robustness of RL-based models under safety-critical conditions. Our SCENGE can generate hundreds of adversarial variants per scene, covering diverse agent interactions and failure modes, facilitating the safety evaluation of AD systems. Our codes can be found at https://scenge.github.io.

1 Introduction

Over the past decade, autonomous vehicles (AV) have advanced significantly [1, 2, 3, 4]. As these systems approach widespread deployment, ensuring their safety and reliability has become critical. Simulation-based testing [5, 6] offers a controlled, reusable, and cost-effective way of evaluating behavior under various conditions, particularly safety-critical scenarios that probe the safety capacity of AV. Therefore, how to generate safety-critical scenarios to effectively and efficiently reveal the safety flaws of AV instead of manually crafting them has attracted significant attention and extensive research [7, 8, 9, 10, 11].

However, most existing methods are constrained by predefined threat templates [11, 12, 13] or rule-based strategies [7, 8, 9, 10] based on typical driving situations and expert experience; therefore, they fail to capture the wide variety of edge cases that can pose significant risks to system safety, showing in weak *risk exposure* abilities. These limitations compromise the comprehensiveness of simulation-based testing, restricting its capacity to assess the safety and reliability of AV.

To address these limitations, we propose SCENGE, a two-stage framework that exposes safety vulnerabilities in AV by performing adversarial threat generation and collaborative trajectory evolution. Given a simple natural language description of a benign driving scene, SCENGE first proposes *Meta-Scenario Generation*, which prompts an LLM to infer a safety-critical scenario in which an adversarial agent threatens the safe operation of the ego vehicle. Here, we construct a structured driving knowledge base using retrieval augmented generation (RAG) [14], incorporating traffic regulations, driver qualification standards, and realistic pre-crash scenarios. The LLM is guided to generate violations of these safety principles, enabling the generated scenarios to intentionally reflect safety violations and edge cases in traffic environments. The generated meta-scenario with a single adversarial agent is expressed in an executable programming language (*i.e.*, Scenic [15, 16]), which can be directly executed within the CARLA simulator [17]. Subsequently, *Complex Scenario Evolution* increases the threat level of the meta-scenario by introducing additional collaborative background vehicles to create a more complex and dynamic environment. In particular, an adversarial collaboration graph is constructed to model interactions among agents and identify key background vehicles that are most influential to the collision outcome in the meta-scenario. Trajectory-level perturbations are applied to these selected background vehicles temporally and spatially coherently. Rather than causing direct collisions, these coordinated modifications intensify interaction complexity, increasing the likelihood of a collision between the ego vehicle and the adversarial agent.

Extensive experiments on multiple RL-based AV models demonstrate that SCENGE uncovers more severe collision cases (+31.96%) on average than state-of-the-art baselines. We further evaluate SCENGE on a large vision-language model (VLM)-based AV system and observe that the generated scenarios lead to consistent reductions in driving score, indicating its effectiveness in challenging high-level semantic reasoning components. Moreover, adversarial training on these scenarios improves the robustness of RL-based models under safety-critical conditions, suggesting that the scenarios generated by SCENGE effectively reveal critical failure modes in AV models. Overall, our SCENGE can generate hundreds of adversarial variants from a single benign scene description, encompassing various agent interaction patterns and safety violation types. This generation capability improves test coverage and enables a structured and repeatable evaluation process. Our main **contributions** are:

- We propose SCENGE, a two-stage framework that combines adversarial threat generation and collaborative trajectory evolution to expose safety vulnerabilities in AV systems.
- We proposed two components: Meta-Scenario Generation, which generates a richly detailed meta-scenario in a programming language, by driving safety knowledge priors augmented LLMs' reasoning; Complex Scenario Evolution, which enhances the threats by perturbing the trajectory of selected background vehicles in an Adversarial Collaborator Graph.
- Extensive experiments conducted on diverse RL-based AV models show the effectiveness of SCENGE (+31.96% collision rate on average) compared to state-of-the-art baselines.

2 Related Work

Simulation-Based Testing for AV. Simulation-based testing has become a mainstream approach for evaluating AV, offering a cost-effective and controlled environment for assessing performance across a wide range of driving conditions in the simulation environment, such as CARLA [17], MetaDrive [18], LimSim [19], *etc*. One of the key advantages of simulation is its ability to recreate complex, rare, and potentially dangerous driving scenarios that are difficult to replicate in real-world testing. Unlike physical testing, where extreme conditions may be costly or risky, digital simulations can model many traffic scenarios, weather conditions, and vehicle interactions, making them an invaluable tool for evaluating AV. Another significant benefit is the ability to test the system in a controlled and repeatable manner. In contrast to the real-world testing, simulations provide a standardized environment where variables (*e.g.*, weather, traffic density, road conditions) can be precisely manipulated, allowing for detailed analysis of the performance and behavior of AV. This consistency in testing is critical for identifying performance weaknesses and ensuring that the system operates within safety parameters.

Safety-Critical Scenario Generation. The generation of safety-critical scenarios plays a crucial role in evaluating the robustness and safety of AV systems, especially under rare or extreme conditions that challenge decision-making. Existing approaches can be broadly categorized into three types. *Generative models* [7, 20, 21, 22, 23] learn from real-world driving data to create realistic and diverse traffic situations. *Optimization-based methods* [24, 8, 25, 26] synthesize targeted scenarios via tailored objective functions. *Semantic-driven methods* [27, 28, 13, 29, 30], particularly those leveraging world models, aim to incorporate high-level contextual knowledge for controllable scenario creation. Recent methods such as ChatSUMO [31] and Chat2Scenario [32] generate scenarios from language or log data, while LEADE [33] and D2RL [34] enhance coverage through semantic replay or trajectory compression. Although these approaches improve scenario diversity, they remain constrained by existing data distributions and do not produce new safety-critical threats.

Prior methods have primarily focused on generating rule-violating behaviors from individual agents or replaying observed high-risk patterns, limiting their ability to reveal novel, compound failure modes. While recent LLM-based approaches improve semantic coverage, they typically lack fine-grained control over multi-agent interactions. These **limitations** motivate SCENGE, which address semantic novelty and emergent risk through structured generation and collaborative perturbation.

3 SCENGE Approach

SCENGE is designed to target AV system failures stemming from explicit rule violations and subtle multi-agent interactions, uncovering failure cases typically overlooked by template-based or single-agent approaches. Specifically, SCENGE first synthesizes a rule-violating adversarial agent via knowledge-guided language reasoning. Then it perturbs key background vehicles identified through an Adversarial Collaborator Graph to induce interaction-driven planning failures. This approach enables the generation of highly plausible, safety-critical scenarios that reveal system-level vulnerabilities in AV systems. An overview is illustrated in Fig. 1.

3.1 Problem Definition

Let $\mathcal{S}_{\mathrm{meta}} = \{\mathbf{a}_{\mathrm{ego}}, \mathbf{a}_{\mathrm{adv}} \mid \mathrm{R}, \mathrm{L}\}$ denote the **meta scenario**, which includes an ego vehicle $\mathbf{a}_{\mathrm{ego}}$ and a single adversarial agent $\mathbf{a}_{\mathrm{adv}}$ operating within an environmental context defined by road type R and traffic light state L. The adversarial agent is further specified by semantic properties $(\mathrm{c}, \mathrm{p}, \mathrm{b})$, denoting its type, position, and behavior, respectively. To construct such scenarios, we start from a benign natural language description Φ_{base}, augmented by a fixed instruction prompt Φ_{inst} to induce safety-violating behavior. A retrieval function f_R selects relevant entries from a knowledge base $\mathbb{D}$, and the resulting context is used by an LLM f_{LLM} to produce semantic descriptions $\langle \Phi_{\mathrm{c}}, \Phi_{\mathrm{p}}, \Phi_{\mathrm{b}}, \Phi_{\mathrm{R}}, \Phi_{\mathrm{L}} \rangle$ for adversarial agent and environment properties. These are subsequently parsed into structured values $(\mathrm{c}, \mathrm{p}, \mathrm{b}, \mathrm{R}, \mathrm{L})$ and instantiated in Scenic to define $\mathcal{S}_{\mathrm{meta}}$.

We define the **adversarial scenario** as $\mathcal{S}_{adv} = \mathcal{S}_{meta} \cup \{\mathbf{a}_1, \ldots, \mathbf{a}_N\}$, where N denotes the number of background vehicles. Each background vehicle $\mathbf{a}_i$ follows a trajectory $\tau_i = \{(x_t, y_t)\}_{t=0}^{T}$, representing its simulated coordinates over T frames, where $i \in \{1, \ldots, N\}$. A subset of background vehicles, indexed by $K \subset \{1, \ldots, N\}$, is selected for perturbation. Their trajectory segments are optimized to induce collaborative risky behaviors that increase the threat level of $\mathcal{S}_{\mathrm{meta}}$. Specifically, for each selected vehicle $\mathbf{a}_{i^\star}$ with $i^\star \in K$, we identify a keyframe $t^\star_{i^\star}$ as the most influential frame, and define the corresponding perturbable segment $\tilde{\tau}_{i^\star} \subset \tau_{i^\star}$ as a temporal window centered at this keyframe. These segments are perturbed by optimizing the objective function $\mathcal{L}$, yielding the final adversarial scenario $\mathcal{S}_{adv}$ with optimized segments $\{\tilde{\tau}^\star_{i^\star}\}_{i^\star \in K}$.

3.2 Meta-Scenario Generation

Given a benign scenario description Φ_{base}, which typically specifies a normal traffic situation without threats (*e.g.*, `the ego car is driving across the corner`), our goal is to construct a meta-scenario $\mathcal{S}_{\mathrm{meta}}$ in which $\mathbf{a}_{\mathrm{adv}}$ introduces a safety-critical threat. The process comprises two main components: ❶ constructing a structured driving knowledge base via RAG, and ❷ generating an executable scenario description using an LLM informed by that prior.

Safety Driving Knowledge Construction. The knowledge base $\mathbb{D} = \{\mathbf{D}_r, \mathbf{D}_l, \mathbf{D}_c\}$ consists of three components, each representing a distinct aspect of driving knowledge essential for simulating normative and adversarial traffic behavior. ❶ $\mathbf{D}_r$ contains 27 driving regulations segmented from official manuals in the USA, Germany, and China, covering behaviors such as lane merging, overtaking, and

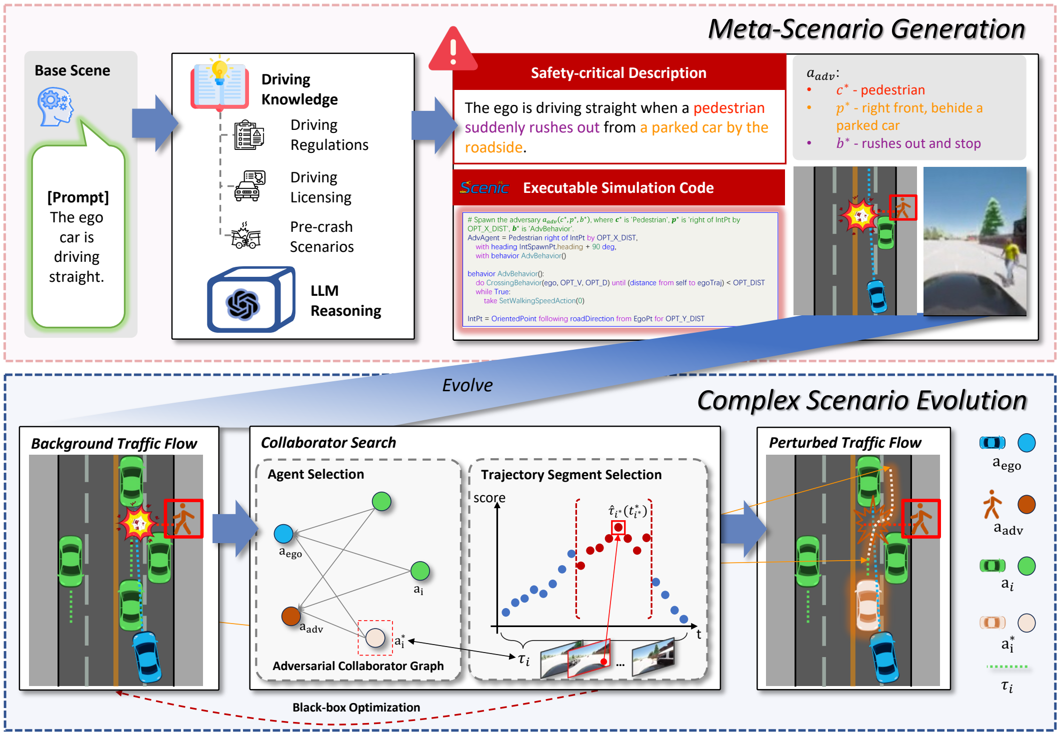


Figure 1: *Framework overview*. Given a simple prompt of a benign scene, SCENGE first performs Meta-Scenario Generation, where an LLM is prompted to generate an executable meta-scenario, grounded in violations of established driving safety knowledge prior. Subsequently, Complex Scenario Evolution constructs an Adversarial Collaborator Graph to identify key agents within the complex traffic environment and perturb their trajectories to maximize adversarial impact.

other maneuvers. ❷ $\mathbf{D}_l$ includes 100 standardized driver's license test questions and answers that assess traffic rule knowledge, situational awareness, and safe behavior selection. Together, $\mathbf{D}_r$ and $\mathbf{D}_l$ provide normative behavioral priors intentionally violated to construct safety-critical adversarial behaviors. In contrast, ❸ $\mathbf{D}_c$ comprises 14 pre-crash scenarios drawn from taxonomies in the NHTSA Pre-Crash Typology Report [35] (*e.g.*, unprotected left turns, red-light violations), providing concrete adversarial patterns for scenario construction. Collectively, these components inform the synthesis of plausible threat scenarios and support the generation of critical adversarial conditions.

LLM-Driven Scenario Generation. Given a base description Φ_{base} of a benign driving scenario, the LLM is prompted to generate a detailed, safety-critical scenario by introducing one main adversarial agent $\mathbf{a}_{\text{adv}}$ into the scenario. It infers the agent's properties and the associated environmental context through in-context learning [36]. However, simply adopting an LLM may lead to unsafe or unrealistic critical scenarios; thus, we ground the reasoning process in structured driving knowledge. To this end, relevant knowledge is retrieved from the database $\mathbb{D}$ and combined with the instruction prompt Φ_{inst} to form the input to the LLM f_{LLM}, see the Supplementary Material for details on Φ_{inst}. The generation process is formalized as:

$$\langle \Phi_{\text{c}}, \Phi_{\text{p}}, \Phi_{\text{b}}, \Phi_{\text{R}}, \Phi_{\text{L}} \rangle = f_{\text{LLM}} \left(\mathbf{a}_{\text{ego}}, \Phi_{\text{base}}, f_R(\mathbb{D}, \Phi_{\text{base}}) \mid \Phi_{\text{inst}} \right), \tag{3.1}$$

where each Φ_* represents a natural language description of a scenario element, including the adversarial agent's properties and environmental context. The instruction prompt Φ_{inst} explicitly guides the model to generate rule-violating yet plausible actions, grounded in retrieved safety knowledge. Although expressed in textual form, the generation is controlled through few-shot prompting and slot-based templates, ensuring the outputs remain semantically structured and scenario-compatible.

The generated descriptions are then parsed into structured values $(\text{c}, \text{p}, \text{b}, \text{R}, \text{L})$ and populated into a predefined Scenic template. This template encodes scenario-level semantics while enforcing syntactic and physical constraints, bridging language-driven generation and executable simulation. The resulting program is run in the simulator to instantiate $\mathcal{S}_{\text{meta}}$.

3.3 Complex Scenario Evolution

Building on the generated meta-scenario, Complex Scenario Evolution enhances its complexity by introducing background vehicles $\{\mathbf{a}_1, \ldots, \mathbf{a}_N\}$ with collaborative risky trajectories. To that end, their interactions with $\mathbf{a}_{\text{adv}}$ and $\mathbf{a}_{\text{ego}}$ are adjusted to create a more challenging scenario for the AV. This process comprises two main components: ❶ *Collaborator Search*, which identifies the background vehicles that can most amplify the adversarial nature of the scenario, and ❷ *Trajectory Perturbation*, which adjusts the selected vehicles to maximize the adversarial impact.

Collaborator Search. To identify influential background vehicles, we construct an Adversarial Collaborator Graph $\boldsymbol{G}$, where each node corresponds to an agent in the scenario, and the edges reflect directional behavioral relevance, particularly emphasizing the impact of background vehicles on the ego vehicle and adversarial agent. This graph is derived from a frame-wise attention matrix $\boldsymbol{M}_{\text{att}}$ that models trajectory-level dependencies using ego and adversarial trajectories as queries and background trajectories as keys. Specifically:

$$\boldsymbol{M}_{\text{att}} = \text{softmax}\left(\frac{(\tau_{\text{ego}}, \tau_{\text{adv}}) \cdot (\tau_1, \ldots, \tau_N)^\top}{\sqrt{d}} + \boldsymbol{M}_{\text{mask}} + \log \boldsymbol{M}_{\text{decay}}\right), \tag{3.2}$$

where d is the dimension of τ, $\boldsymbol{M}_{\text{mask}}$ enforces causality by preventing attention to future frames, and $\boldsymbol{M}_{\text{decay}}$ introduces a temporal decay bias to emphasize recent interactions. Further details of $\boldsymbol{M}_{\text{att}}$, $\boldsymbol{M}_{\text{mask}}$, and $\boldsymbol{M}_{\text{decay}}$ are provided in the Supplementary Material.

Based on $\boldsymbol{M}_{\text{att}}$, we perform *Collaborator Search* in two stages. First, we aggregate attention scores across frames to estimate the overall influence of each background vehicle from the ego vehicle and adversarial agent, and identify the Top-k most relevant vehicles indexed by K. Then, for each $i^\star \in K$, we locate the keyframe $t^\star_{i^\star}$ receiving the highest attention for vehicle $\mathbf{a}_{i^\star}$, and extract a local temporal window $\tilde{\tau}_{i^\star}$ centered at keyframe as its perturbable trajectory segment. These segments serve as the input to the subsequent trajectory perturbation module.

Trajectory Perturbation. Given the selected collaborators indexed by K and their perturbable segments $\{\tilde{\tau}_{i^\star}\}_{i^\star \in K}$, we optimize these trajectories to maximize the adversarial impact on the ego vehicle. This is formulated as the following objective:

$$\{\tilde{\tau}^\star_{i^\star}\}_{i^\star \in K} = \underset{\{\tilde{\tau}_{i^\star}\}_{i^\star \in K}}{\arg\max}\ \mathcal{L}(\tilde{\tau}_{\text{ego}}, \tilde{\tau}_{\text{adv}}, \{\tilde{\tau}_{i^\star}\}_{i^\star \in K}), \tag{3.3}$$

The optimization follows an iterative, gradient-based procedure. Specifically, for each perturbable segment, we compute the gradient of $\mathcal{L}$ *w.r.t* the trajectory coordinates and update them in the direction that increases the loss. Each update step uses a small, fixed step size and is projected back to the feasible space to ensure realism. The process continues until convergence or a predefined number of steps is reached.

$$\mathcal{L} = \lambda_1 \underbrace{\|\tilde{\tau}_{i^\star} - \tilde{\tau}_{\text{ego}}\|_2}_{\mathcal{L}_{\text{ego}}} + \lambda_2 \underbrace{\|(\tilde{\tau}_{i^\star} - \tilde{\tau}_{\text{ego}}) \times (\tilde{\tau}_{\text{adv}} - \tilde{\tau}_{\text{ego}})\|_\perp}_{\mathcal{L}_{\text{occ}}} + \lambda_3 \underbrace{\|\Delta^2 \tilde{\tau}_{i^\star}\|_2^2}_{\mathcal{L}_{\text{smooth}}}. \tag{3.4}$$

The objective function $\mathcal{L}$ comprises three components, as shown in Eq. (3.4). ❶ $\mathcal{L}_{\text{ego}}$ minimizes the Euclidean distance between the perturbed background trajectory $\tilde{\tau}_{i^\star}$ and the ego trajectory $\tilde{\tau}_{\text{ego}}$ within a temporal window. ❷ $\mathcal{L}_{\text{occ}}$ minimizes the normalized perpendicular distance via a 2D cross product, promoting alignment along the ego–adversary line-of-sight. ❸ $\mathcal{L}_{\text{smooth}}$ penalizes second-order differences $\Delta^2 \tilde{\tau}_{i^\star}$ to reduce abrupt motion changes. From a behavioral modeling perspective, $\mathcal{L}_{\text{ego}}$ encourages spatial proximity to induce planning hesitation, $\mathcal{L}_{\text{occ}}$ amplifies perceptual ambiguity through occlusion, and $\mathcal{L}_{\text{smooth}}$ ensures kinematic feasibility via smoothness constraints, collectively balancing adversarial strength with physical plausibility. Finally, the optimized perturbations $\{\tilde{\tau}^\star_{i^\star}\}_{i^\star \in K}$ replace the corresponding segments of the original trajectories, yielding the final adversarial scenario $\mathcal{S}_{adv}$, which poses a significant threat to the ego vehicle's safe driving

Table 1: **Evaluation of adversarial scenario generation methods across CR, and OS metrics.** Performance is assessed on eight base scenarios in CARLA, averaged across PPO, SAC, and TD3 models. Best results are highlighted in bold. Higher CR and lower OS values, indicate better adversarial effectiveness.

Metric	Algo.	Straight Obstacle	Turning Obstacle	Lane Changing	Vehicle Passing	Red-light Running	Unprotected Left-turn	Right-turn	Crossing Negotiation	Avg.
		Base Traffic Scenarios								
CR ↑	LC	0.241	0.159	0.736	0.792	0.317	0.325	0.321	0.313	0.401
	AS	0.451	0.399	0.726	0.832	0.177	0.335	0.115	0.303	0.417
	CS	0.391	0.679	0.756	0.812	0.237	0.325	0.411	0.333	0.493
	AT	0.441	0.379	0.646	0.782	0.317	0.315	0.321	0.353	0.440
	ChatScene	0.750	0.647	0.660	**0.907**	**0.833**	0.620	0.743	0.850	0.751
	Ours	**0.860**	**0.773**	**0.837**	0.897	0.823	**0.747**	**0.763**	**0.863**	**0.820**
OS ↓	LC	0.789	0.816	0.566	0.530	0.799	0.790	0.692	0.717	0.712
	AS	0.694	0.687	0.561	0.506	0.866	0.775	0.841	0.721	0.706
	CS	0.726	0.552	0.549	0.513	0.839	0.787	0.649	0.708	0.665
	AT	0.696	0.706	0.599	0.528	0.805	0.795	0.689	0.698	0.690
	ChatScene	0.559	0.572	0.607	0.472	0.544	0.656	0.511	**0.459**	0.548
	Ours	**0.503**	**0.526**	**0.504**	**0.457**	**0.507**	**0.519**	**0.498**	0.477	**0.499**

4 Experiment and Evaluation

4.1 Experimental Setup

Simulation environment and benchmark. We utilise the CARLA simulator [17], an open-source and highly customizable urban driving simulator, to create a closed-loop simulation environment. We adopt SafeBench [37] as the benchmarking framework, which supports diverse RL-based AV agents and standardized evaluation. Following [11], we use 8 base traffic scenarios (*e.g.*, `Straight Obstacle, Lane Changing`) curated from the NHTSA Pre-Crash Typology Report [35], each containing 10 diverse driving routes. For each route, 10 adversarial scenarios are generated, resulting in 800 challenging scenarios for evaluation and comparison per method. Experiments are conducted on a server with an Intel(R) Core(TM) i9-14900K CPU, 128GB system memory, and two NVIDIA GeForce RTX 4090 GPUs with 24GB memory.

AV algorithms. Following [11], we mainly employ 3 representative RL-based AV algorithms as testing agents, including Proximal Policy Optimization (PPO) [38], Soft Actor-Critic (SAC) [39], and Twin Delayed Deep Deterministic Policy Gradient (TD3) [40].

Compared baselines. We compare our framework SCENGE, with several existing scenario generation methods, including Learning-to-Collide (LC) [7], AdvSim (AS) [8], Carla Scenario Generator (CS) [9], Adversarial Trajectory Optimization (AT) [10], and ChatScene [11]. For fair comparisons, each method is applied independently on the same 8 base scenarios and routes to generate 800 challenging scenarios under consistent generation logic and evaluation settings.

Metrics. Following SafeBench [37], we adopt a set of key metrics to evaluate AV performance in generated scenarios. Two core indicators are used: the **collision rate** (CR ↑) measures the frequency of collisions and reflects safety risk, and the **overall score** (OS ↓) aggregates system-level performance.

In addition, we evaluate three additional dimensions: the **safety level** (*frequency of running red lights* (RR ↑), *frequency of running stop signs* (SS ↑), and *average distance driven out of road* (OR ↑), the **functionality level** (*route following stability* (RF ↓), *average percentage of route completion* (Comp ↓), and *average time spent to complete the route* (TS ↑)), and the **etiquette level** (*average acceleration* (ACC ↑), *average yaw velocity* (YV ↑), and *frequency of lane invasion* (LI ↑). Higher (↑) values indicate worse performance, while ↓ indicates the contrary.

Implementation details. In this paper, the LLM used for Meta-Scenario Generationis qwq-32b [41], the reasoning model from the Qwen series. In the Complex Scenario Evolution module, we construct 10 background vehicles and perturb the trajectories of 4 selected ones, each over 60% of their trajectory. The 4 perturbed vehicles are selected based on the highest attention relevance to ego and adversarial agents, as defined in the collaborator graph. The 60% perturbation window is centered around each vehicle's most relevant keyframe. We set γ in the decay matrix to 0.8. In the loss calculation, we set $\lambda_1 = 0.3$, $\lambda_2 = 0.5$, and $\lambda_3 = 0.2$.

4.2 Main Results

Tab. 1 and Tab. 2 summarize the performance of SCENGE compared to several baseline methods across eight standard base traffic scenarios. We evaluate three primary aspects: *safety and risk exposure*, *functionality under stress*, and *driving etiquette*.

Safety and Risk Exposure. As shown in Tab. 1, SCENGE achieves the highest average CR, with a relative improvement of **31.96%** over the strongest baseline. Unlike prior approaches relying on excessive rule violations to induce failures, SCENGE maintains moderate RR, SS, and OR values while causing significantly more frequent and persistent collisions. These results suggest that SCENGE induces collisions by targeting control and planning weaknesses, without relying on conspicuous or excessive rule violations.

Functionality Challenges. As shown in Tab. 1, SCENGE obtains the lowest OS, representing a **16.52%** reduction over the best baseline. A **4.96%** drop in RF and a **29.16%** reduction in Comp are also observed, according to the functionality metrics in Tab. 2. TS remains moderate, reflecting shorter trajectories caused by early collisions. This demonstrates that SCENGE induces rapid and decisive failures through persistent planning disruptions.

Table 2: **Aggregated evaluation results across safety, functionality, and etiquette dimensions.** Each value represents the average over three RL-based AV agents and eight base scenarios.

Algo.	Safety Level			Functionality Level			Etiquette Level		
	RR ↑	SS ↑	OR ↑	RF ↓	Comp ↓	TS ↑	ACC ↑	YV ↑	LI ↑
LC	**0.325**	0.165	0.039	0.884	0.807	0.224	0.225	0.231	0.087
AS	0.299	0.167	0.032	0.901	0.821	0.269	0.217	0.233	0.102
CS	0.312	**0.168**	**0.043**	0.880	0.817	0.252	0.229	0.235	0.106
AT	0.311	0.167	0.035	0.883	0.802	**0.287**	0.233	0.236	0.112
ChatScene	0.228	0.145	0.018	0.890	0.571	0.074	0.281	0.225	0.064
Ours	0.231	0.125	0.009	**0.838**	**0.472**	0.124	**0.402**	**0.359**	**0.179**

Driving Etiquette. As shown in Tab. 2, SCENGE increases ACC, YV, and LI by **16.5%**, **12.7%**, and **8.48%** respectively. These results suggest that SCENGE causes AV to behave less smoothly and more erratically in ways that remain socially plausible. Introducing temporally coordinated perturbations across multiple agents disrupts fine-grained control and social driving compliance, revealing limitations that simpler, single-agent or rule-based methods fail to expose.

4.3 Evaluation on VLM AV Models

Beyond RL-based AV models, we further evaluate our generated scenarios on VLM-based AV models, focusing on LMDrive [2], a large vision-language model for AV deployed on the CARLA Leaderboard [42]. LMDrive navigates by following natural language instructions sequentially, using the multi-view camera and Lidar perception for scene understanding and planning. To accommodate its instruction-driven execution mode, we redesign the test routes into multi-instruction sequences that mimic real-world navigation tasks. Evaluation follows LMDrive's original metrics: Route Completion (RC), Infraction Score (IS), and Driving Score (DS). Complete metric definitions and computation details are provided in the Supplementary Material. We evaluate LMDrive under three increasingly challenging settings: ❶ ego-only benign routes as a baseline, ❷ meta-scenarios with a single adversarial agent, and ❸ full adversarial scenarios generated by our framework, including the perturbed background vehicle. As shown in Tab. 3, LMDrive's performance drops from **87.7** DS in the benign case to **83.7** in meta-scenarios and further to **80.4** under full adversarial conditions. These results demonstrate that our generated scenarios significantly stress LMDrive's planning capability, especially under rare or occluded interactions.

Table 3: Performance of LMDrive under generated adversarial scenarios across eight base traffic tasks.

Algo.	Benign Scenario	Meta-Scenario	Adversarial Scenario
RC	92.2 $_{\pm 2.9}$	92.9 $_{\pm 2.7}$	89.9 $_{\pm 5.1}$
IS	0.97 $_{\pm 0.01}$	0.9 $_{\pm 0.05}$	0.89 $_{\pm 0.04}$
DS	87.7 $_{\pm 2.4}$	83.7 $_{\pm 4.7}$	80.4 $_{\pm 5.5}$

4.4 Ablation Studies

We perform ablation experiments by selectively disabling key modules and observing the effect on overall performance. Otherwise specified, this part keeps the same setting as the main experiment. Fig. 2 reports the CR and OS under different settings. ❶ Knowledge Prior. Removing $\mathbf{D}_r$ yields CR 79.4% and OS 52.3%, reflecting its role in guiding rule-focused violations. Removing $\mathbf{D}_l$ gives

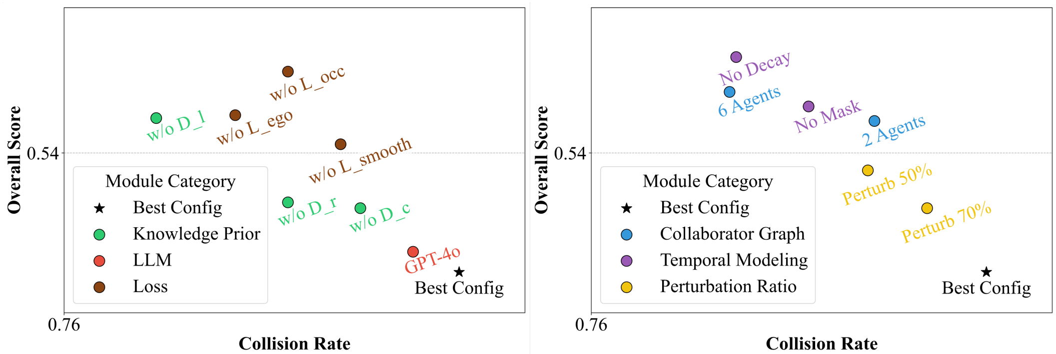


Figure 2: Scatter plot of **CR** ↑ vs. **OS** ↓ across ablation settings. Each color denotes an ablation type, and each point represents a specific variant. Points closer to the bottom-right indicate stronger adversarial effects.

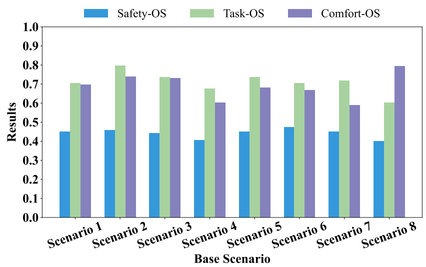
CR 77.4% and OS 55.2%, showing its effect on enhancing logical consistency in agent behavior. Removing $\mathbf{D}_c$ results in CR 80.5% and OS 52.1%, confirming its importance in producing realistic and high-risk scenarios. ❷ LLM. Among the compared models, GPT-4o yields CR 81.3% and OS 50.6%, benefiting from strong general capabilities but often producing conservative scenarios with limited adversarial diversity. In contrast, qwq-32b [41] achieves the highest CR 82% and lowest OS 49.9%, generating coherent, rule-violating, high-impact cases that better leverage driving priors. Based on these results, qwq-32b is adopted as the default LLM in our framework. ❸ Collaborator Graph. We ablate the number of perturbed agents with settings of 2, 4, and 6 during agent selection. Perturbing 4 agents performs best with CR 82% and OS 49.9%, balancing adversarial strength and scenario plausibility. Using 2 agents lowers interaction complexity, resulting in CR 80.3% and OS 55.1%, while 6 agents introduce excessive interference and unrealistic behavior, yielding CR 78.1% and OS 56.1%. These results suggest that moderate perturbation is most effective. ❹ Temporal Modeling. We ablate three temporal modeling configurations: with both mask and decay, without mask, and without decay. The full setting yields the best result with CR 82% and OS 49.9%. Removing the temporal mask reduces temporal causality in collaborator selection, leads to CR 79.3% and OS 55.6%, while removing the temporal decay results in CR 78.2% and OS 57.3%. These results highlight the complementary role of both components in capturing temporally coherent influence. ❺ Perturbation Ratio. We compare three perturbation ratios centered around the selected keyframe: 50%, 60%, and 70%. Perturbing 60% of the segment achieves the best result with CR 82% and OS 49.9%. A 50% ratio leads to CR 80.2% and OS 53.4%, indicating insufficient behavioral deviation, while 70% causes CR 81.1% and OS 52.1% due to over-modification and reduced plausibility. These results suggest that moderate perturbation best balances realism and adversarial effect. ❻ Loss. We ablate each component in $\mathcal{L}$ to assess its contribution. Removing $\mathcal{L}_{\text{ego}}$ leads to CR 78.6% and OS 55.3%, reflecting reduced collision targeting. Removing $\mathcal{L}_{\text{occ}}$ results in CR 79.4% and OS 56.8%, indicating weaker alignment between adversary and ego. Excluding $\mathcal{L}_{\text{smooth}}$ yields CR 80.2% and OS 54.3%, with trajectories becoming visibly unstable. The full loss yields the best trade-off, and ablating any term consistently reduces CR and increases OS.

4.5 Adversarial Training on the Generated Scenarios

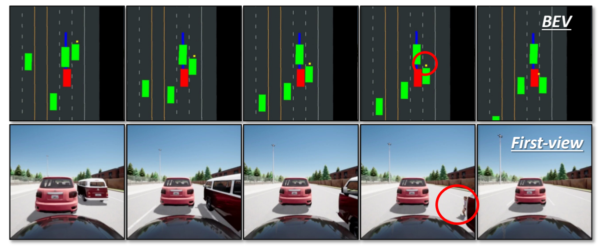
To further evaluate the utility of our SCENGE, we conduct adversarial training (AT) experiments using the generated scenarios. Specifically, we adversarially train the SAC-based ego vehicle across eight base traffic scenarios using scenes from the first eight routes per scenario for each method, and evaluate on unseen scenes from the remaining two routes. The training process uses 500 epochs with a learning rate of 0.0001. As shown in Tab. 4, adversarial training with SCENGE-generated scenarios yields the best overall results among all methods, reducing the CR by **3.1%** while increasing the OS by **94.7%**. This highlights the method's superior ability to create

Table 4: **Evaluation of ego agent performance after adversarial training.** SAC-based ego models are adversarially trained on 80% scenarios generated by each method, and tested to measure performance improvements in CR and OS.

Metric	LC	AS	CS	AT	ChatScene	**Ours**
CR ↑	0.210	0.216	0.176	0.135	0.043	**0.031**
OS ↓	0.813	0.806	0.825	0.864	0.905	**0.947**



(a) Composite metric scores (Safety-OS, Task-OS, Comfort-OS) of the SAC-based ego vehicle across 8 adversarial scenarios.



(b) Representative frames from a case scenario generated by SCENGE. The keyframes highlight how the agents are jointly leading to a collision.

Figure 3: Model performance over different base scenes and a case visualization of SCENGE.

training curricula that enhance the robustness of AV. These findings confirm that exposing the ego vehicle to complex, multi-agent, and socially disruptive adversarial scenarios generated by SCENGE directly contributes to advancing the reliability and safety of autonomous driving systems.

4.6 Case Study

As shown in Fig. 3a, we evaluate the SAC-based ego vehicle across three composite metrics: safety, task completion, and comfort. These metrics summarize AV behavior by aggregating relevant low-level indicators, and their formal definitions are provided in the Supplementary Material. We select Scenario 1 for detailed analysis because its scores lie near the median across all three dimensions, making it a representative and balanced failure case. It is neither a trivial success nor an extreme failure, but rather a typical scenario where multiple contributing factors combine realistically. This makes it well-suited for illustrating how adversarial interactions manifest under plausible traffic conditions. In Fig. 3b, we show some frames from the final adversarial scenario (both BEV and first-view). The adversarial agent (pedestrian) suddenly crosses the road from behind a parked truck, directly into the ego vehicle's path. Meanwhile, background vehicles induce restrict maneuvering space and strong occlusion: one occupies the left adjacent lane, blocking a potential lane change and exerting close-range pressure on the ego vehicle, a behavior promoted by $\mathcal{L}_{\text{ego}}$; another vehicle ahead reduces visibility by obstructing the line-of-sight to the adversary, aligning with the occlusion modeling objective of $\mathcal{L}_{\text{occ}}$. These compounded interactions prevent the ego vehicle from executing a safe evasive action. This illustrates how SCENGE generates adversarial scenarios where the combination of subtle behavior of background vehicles exposes critical AV vulnerabilities.

5 Conclusion and Future Work

In this paper, we introduce SCENGE, a two-stage framework for generating safety-critical scenarios to expose vulnerabilities in AV. From a benign scene description, *Meta-Scenario Generation* uses an LLM grounded in structured driving knowledge to generate an executable meta-scenario. *Complex Scenario Evolution* then introduces background vehicles and perturbs key trajectories to increase interaction complexity and induce failures. Experiments on multiple RL-based AV models show that SCENGE reveals more severe collision cases.

Limitations. While SCENGE shows strong performance in generating adversarial scenarios, it has several limitations. It depends on high-fidelity simulations that may not reflect real-world complexity. Evaluation is limited to specific AV models, and performance may vary with traffic complexity, LLM choice, and knowledge base.

Ethical Statement and Broader Impact. This work does not involve human subjects, private data, or real-world deployment. All experiments are conducted in simulation with procedurally generated scenarios. SCENGE supports academic research by identifying failure modes in AV under adversarial conditions. While it introduces challenging scenarios, its purpose is solely evaluation and stress-testing, not malicious use. We do not foresee any direct ethical risks or negative societal impacts; the work aims to promote safer AV technologies.

References

[1] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[2] Hao Shao, Yuxuan Hu, Letian Wang, Guanglu Song, Steven L Waslander, Yu Liu, and Hongsheng Li. Lmdrive: Closed-loop end-to-end driving with large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15120–15130, 2024.

[3] Yingzi Ma, Yulong Cao, Jiachen Sun, Marco Pavone, and Chaowei Xiao. Dolphins: Multimodal language model for driving. In *European Conference on Computer Vision*, pages 403–420. Springer, 2024.

[4] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European Conference on Computer Vision*, pages 36–52. Springer, 2022.

[5] Xuan Cai, Xuesong Bai, Zhiyong Cui, Danmu Xie, Daocheng Fu, Haiyang Yu, and Yilong Ren. Text2scenario: Text-driven scenario generation for autonomous driving test. *arXiv preprint arXiv:2503.02911*, 2025.

[6] Qiujing Lu, Xuanhan Wang, Yiwei Jiang, Guangming Zhao, Mingyue Ma, and Shuo Feng. Multimodal large language model driven scenario testing for autonomous vehicles. *arXiv preprint arXiv:2409.06450*, 2024.

[7] Wenhao Ding, Baiming Chen, Minjun Xu, and Ding Zhao. Learning to collide: An adaptive safety-critical scenarios generating method. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2243–2250. IEEE, 2020.

[8] Jingkang Wang, Ava Pun, James Tu, Sivabalan Manivasagam, Abbas Sadat, Sergio Casas, Mengye Ren, and Raquel Urtasun. Advsim: Generating safety-critical scenarios for self-driving vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9909–9918, 2021.

[9] Scenario Runner Contributors. Carla Scenario Runner. `https://github.com/carla-simulator/scenario_runner`, 2019.

[10] Qingzhao Zhang, Shengtuo Hu, Jiachen Sun, Qi Alfred Chen, and Z Morley Mao. On adversarial robustness of trajectory prediction for autonomous vehicles. *arXiv preprint arXiv:2201.05057*, 2022.

[11] Jiawei Zhang, Chejian Xu, and Bo Li. Chatscene: Knowledge-enabled safety-critical scenario generation for autonomous vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15459–15469, 2024.

[12] Chejian Xu, Aleksandr Petiushko, Ding Zhao, and Bo Li. Diffscene: Diffusion-based safety-critical scenario generation for autonomous vehicles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 8797–8805, 2025.

[13] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-drive world models for autonomous driving. In *European Conference on Computer Vision*, pages 55–72. Springer, 2024.

[14] Shangyu Wu, Ying Xiong, Yufei Cui, Haolun Wu, Can Chen, Ye Yuan, Lianming Huang, Xue Liu, Tei-Wei Kuo, Nan Guan, et al. Retrieval-augmented generation for natural language processing: A survey. *arXiv preprint arXiv:2407.13193*, 2024.

[15] Daniel J Fremont, Tommaso Dreossi, Shromona Ghosh, Xiangyu Yue, Alberto L Sangiovanni-Vincentelli, and Sanjit A Seshia. Scenic: a language for scenario specification and scene generation. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 63–78, 2019.

[16] Daniel J Fremont, Edward Kim, Tommaso Dreossi, Shromona Ghosh, Xiangyu Yue, Alberto L Sangiovanni-Vincentelli, and Sanjit A Seshia. Scenic: A language for scenario specification and data generation. *Machine Learning*, pages 1–45, 2022.

[17] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[18] Quanyi Li, Zhenghao Peng, Zhenghai Xue, Qihang Zhang, and Bolei Zhou. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *arXiv preprint arXiv:2109.12674*, 2021.

[19] Licheng Wen, Daocheng Fu, Song Mao, Pinlong Cai, Min Dou, Yikang Li, and Yu Qiao. Limsim: A long-term interactive multi-scenario traffic simulator. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1255–1262. IEEE, 2023.

[20] Shuhan Tan, Kelvin Wong, Shenlong Wang, Sivabalan Manivasagam, Mengye Ren, and Raquel Urtasun. Scenegen: Learning to generate realistic traffic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 892–901, 2021.

[21] Simon Suo, Sebastian Regalado, Sergio Casas, and Raquel Urtasun. Trafficsim: Learning to simulate realistic multi-agent behaviors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10400–10409, 2021.

[22] Davis Rempe, Jonah Philion, Leonidas J Guibas, Sanja Fidler, and Or Litany. Generating useful accident-prone driving scenarios via a learned traffic prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17305–17315, 2022.

[23] Lan Feng, Quanyi Li, Zhenghao Peng, Shuhan Tan, and Bolei Zhou. Trafficgen: Learning to generate diverse and realistic traffic scenarios. In *2023 IEEE international conference on robotics and automation (ICRA)*, pages 3567–3575. IEEE, 2023.

[24] Baiming Chen, Xiang Chen, Qiong Wu, and Liang Li. Adversarial evaluation of autonomous vehicles in lane-change scenarios. *IEEE transactions on intelligent transportation systems*, 23(8):10333–10342, 2021.

[25] Yulong Cao, Chaowei Xiao, Anima Anandkumar, Danfei Xu, and Marco Pavone. Advdo: Realistic adversarial attacks for trajectory prediction. In *European Conference on Computer Vision*, pages 36–52. Springer, 2022.

[26] Hao Xiang, Runsheng Xu, Xin Xia, Zhaoliang Zheng, Bolei Zhou, and Jiaqi Ma. V2xp-asg: Generating adversarial scenes for vehicle-to-everything perception. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3584–3591. IEEE, 2023.

[27] Ruiyuan Gao, Kai Chen, Enze Xie, Lanqing Hong, Zhenguo Li, Dit-Yan Yeung, and Qiang Xu. MagicDrive: Street view generation with diverse 3d geometry control. In *International Conference on Learning Representations*, 2024.

[28] Ziyuan Zhong, Davis Rempe, Yuxiao Chen, Boris Ivanovic, Yulong Cao, Danfei Xu, Marco Pavone, and Baishakhi Ray. Language-guided traffic simulation via scene-level diffusion. In *Conference on Robot Learning*, pages 144–177. PMLR, 2023.

[29] Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14749–14759, 2024.

[30] Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. Occworld: Learning a 3d occupancy world model for autonomous driving. In *European Conference on Computer Vision*, pages 55–72. Springer, 2025.

[31] Shuyang Li, Talha Azfar, and Ruimin Ke. Chatsumo: Large language model for automating traffic scenario generation in simulation of urban mobility. *IEEE Transactions on Intelligent Vehicles*, 2024.

[32] Yongqi Zhao, Wenbo Xiao, Tomislav Mihalj, Jia Hu, and Arno Eichberger. Chat2scenario: Scenario extraction from dataset through utilization of large language model. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pages 559–566. IEEE, 2024.

[33] Haoxiang Tian, Xingshuo Han, Guoquan Wu, Yuan Zhou, Shuo Li, Jun Wei, Dan Ye, Wei Wang, and Tianwei Zhang. Lmm-enhanced safety-critical scenario generation for autonomous driving system testing from non-accident traffic videos. *arXiv preprint arXiv:2406.10857*, 2024.

[34] Shuo Feng, Haowei Sun, Xintao Yan, Haojie Zhu, Zhengxia Zou, Shengyin Shen, and Henry X Liu. Dense reinforcement learning for safety validation of autonomous vehicles. *Nature*, 615(7953):620–627, 2023.

[35] Wassim G Najm, John D Smith, Mikio Yanagisawa, et al. Pre-crash scenario typology for crash avoidance research. Technical report, United States. National Highway Traffic Safety Administration, 2007.

[36] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, Miami, Florida, USA, 2024. Association for Computational Linguistics.

[37] Chejian Xu, Wenhao Ding, Weijie Lyu, Zuxin Liu, Shuai Wang, Yihan He, Hanjiang Hu, Ding Zhao, and Bo Li. Safebench: A benchmarking platform for safety evaluation of autonomous vehicles. *Advances in Neural Information Processing Systems*, 35:25667–25682, 2022.

[38] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[39] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 1861–1870, 2018.

[40] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 1587–1596, 2018.

[41] Qwen Team. Qwq-32b: Embracing the power of reinforcement learning, March 2025.

[42] CARLA team. Carla autonomous driving leaderboard. https://leaderboard.carla.org/, 2020. Accessed: 2021-02-11.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state that SCENGE generates diverse and realistic adversarial scenarios by leveraging LLMs with safety knowledge priors and scenario evolution, and achieves significantly higher CR than baselines. These claims are supported by detailed methodology in Sec. 3 and validated by experiments in Sec. 4.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a dedicated **Limitations** paragraph in Sec. 5, where we clearly state several limitations of SCENGE. These include reliance on high-fidelity simulations, limited evaluation on specific AV models, and sensitivity to factors such as traffic complexity, LLM selection, and knowledge base composition.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: Our paper does not include formal theoretical results, theorems, or proofs. The methodology and contributions are algorithmic, focusing on simulation-based scenario generation and evaluation (Sec. 3, Sec. 4).

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We fully disclose the simulation environment setup, baselines, model details, evaluation metrics, and implementation configurations in Sec. 4. Furthermore, our code is publicly available at `https://scenge.github.io/`, allowing others to reproduce the main results. Specific parameter settings (*e.g.*, agent count, perturbation ratios, loss weights) are reported in detail to support reproducibility.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide open access to our code and necessary resources at `https://scenge.github.io/`, which includes implementation details, simulation setups, and evaluation scripts. The webpage also contains instructions for reproducing the experimental results, including environment setup and usage of the CARLA simulator. Details are further described in the supplemental material.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide detailed descriptions of the simulation setup, benchmark scenarios, model baselines, and metrics in Sec. 4. We also report hyperparameters used for training (*e.g.*, learning rate, perturbation ratios, loss weights), the number of episodes, and details about the models evaluated (RL-based and VLM-based).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report standard deviations in Tab. 3, and aggregate results over multiple base scenarios in Tab. 1 and Tab. 2. Fig. 2 and Fig. 3a further illustrate variability across ablation settings and scenarios. These collectively support the statistical significance of our results (Sec. 4.2, Sec. 4.3, Sec. 4.4, Sec. 4.6).

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provided the types of the CPU, system memory, and GPU in the computer used in our experiment, at the end of the **Simulation environment and benchmark** paragraph.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: We have reviewed and fully complied with the NeurIPS Code of Ethics. Our research does not involve human subjects, personal data, or real-world deployment. All experiments are conducted in simulation.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss potential societal impacts, including benefits for autonomous vehicle safety testing and risks of misuse, in a dedicated **Ethical Statement and Broader Impact** paragraph in Sec. 5.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work does not release any pretrained models or datasets with a high risk of misuse. All components, including scenario generation and evaluation, are conducted in controlled simulation environments for safety research purposes only. The released code operates within these constraints and poses no foreseeable dual-use risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We use publicly available assets such as the CARLA simulator and the SafeBench benchmark, both of which are properly cited in Sec. 4 and the references section. These tools are released under permissive open-source licenses, and we adhere to their terms of use. Other referenced models and datasets are also appropriately credited.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We release a new adversarial scenario generation framework, SCENGE, along with code and configuration files. Our project page provides documentation, including usage instructions, scenario formats, and integration details with the CARLA simulator. All assets are anonymized for submission and will be properly licensed and documented upon publication.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: Our work does not involve any crowdsourcing or research with human subjects. All experiments are conducted in simulation environments.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our research does not involve human subjects or crowdsourced participants; therefore, IRB approval is not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Our method explicitly incorporates a large language model (LLM) as a core component in the Meta-Scenario Generation stage. We describe its role, architecture (qwq-32b), and integration with the driving knowledge prior in detail in Sec. 3.2, as well as analyze its impact in Sec. 4.4.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.