VLM-Enhanced Adversarial Scene Generation from Images and Videos for Safe Autonomous Driving

Tianyi Liu
Technical University of Munich
Munich, Germany
tianyi.liu@tum.de

Yingjie Xu
Technical University of Munich
Munich, Germany
yingjie.xu@tum.de

Yinlong Liu [*]
City University of Macau
Macau, China
ylliu@cityu.edu.mo

Abstract

Safety-critical scenario generation is essential for evaluating autonomous vehicles, yet existing approaches often require extensive manual design and lack scalability. This work proposes an automated framework that combines vision–language models (VLMs) and large language models (LLMs) to generate realistic safety-critical driving scenarios from naturalistic driving videos. We propose a three-mode generation framework that transforms both accident and normal traffic videos into adversarial scenarios while preserving dataset-specific distributions and cultural driving patterns. Our pipeline first employs a VLM to convert input videos into structured scene descriptions capturing road geometry, traffic participants, and their interactions with the ego vehicle. These descriptions are then translated into executable Scenic programs, supported by an LLM-based error-correction module that ensures executable code and stable simulation in CARLA. We evaluate the framework using SafeBench across eight challenging base scenario categories and test the generated scenarios against three reinforcement-learning driving agents. The results demonstrate that our method produces diverse and realistic adversarial situations, improving scenario variety, realism, and failure coverage compared to baseline approaches. Overall, this work shows that integrating VLMs and LLMs enables scalable generation of safety-critical scenarios, offering a promising tool for more robust and comprehensive autonomous-driving evaluation.

1 Introduction

Autonomous driving systems perform well in routine traffic, but failures in rare safety-critical situations remain a major barrier to deployment. Events such as sudden pedestrian crossings, abrupt lane changes, and unexpected braking appear infrequently in natural driving logs, forming a long-tail distribution that models rarely observe [46, 41, 7]. As a result, existing evaluation pipelines can overestimate reliability, despite brittle behavior in precisely the scenarios where safety matters most.

Simulation provides a safe and scalable environment for testing hazardous events [19], yet constructing realistic and diverse safety-critical scenarios remains challenging. Handcrafted tests lack behavioral richness, while accident datasets alone do not capture real-world variability or regional driving habits. Recent LLM-based methods ease scenario authoring but still struggle with realism, multi-agent dynamics, and execution robustness.

We propose a fully automated pipeline that generates safety-critical driving scenarios directly from real driving videos. A vision–language model (VLM) extracts structured scene descriptions, which are translated into executable Scenic programs by a large language model (LLM). An iterative

[*] Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Embodied and Safe-Assured Robotic Systems.

error-correction loop verifies and repairs scenarios in CARLA, reducing manual intervention. To scale beyond scarce crash footage, we introduce a three-mode generation paradigm that converts both accident and normal traffic videos into adversarial scenarios while preserving dataset-specific distributions and cultural traffic patterns.

Contributions This work makes the following contributions:

- An automated VLM–LLM pipeline converts real driving videos into executable, distribution-preserving safety-critical scenarios that reflect regional driving rules and behavior patterns.
- Execution-robust Scenic generation through an LLM-based verification and repair loop integrated with CARLA.
- Extension of retrieval-augmented generation (as in ChatScene [49]) with open LLMs for cost-efficient scaling.
- Comprehensive evaluation in CARLA using reinforcement learning agents across diverse safety-critical categories.

Our results demonstrate improved realism, diversity, and safety-critical coverage compared to prior scenario-generation approaches, advancing scalable evaluation for autonomous driving systems.

2 Related Work

Safety-critical scenario generation for autonomous driving has been explored through four primary paradigms: data-driven, adversarial, knowledge-based, and large-model–driven approaches. Each direction addresses different challenges in realism, coverage, and controllability.

Data-driven methods. Early work replayed naturalistic trajectories from datasets such as NGSIM [42], highD [29], and inD [2]. AADS [31] augments this paradigm by combining LiDAR-scanned environments with simulated agents for photorealistic traffic. While realistic, replay-based systems rarely expose safety-critical edge cases. Generative models (e.g., VAEs, GANs, flows) extrapolate rare events [18, 34, 3] but still inherit dataset bias and cannot guarantee consistent exposure to extreme behaviors.

Adversarial methods. Adversarial scenario generation explicitly targets failure. Naturalistic adversaries [20] inject rare but human-plausible maneuvers, while RL-based adversaries [17, 43] learn policies that trigger failures. Optimization-based frameworks such as Adaptive Stress Testing [28] and ReGentS [48] search or edit trajectories to maximize risk under semantic constraints. These methods efficiently reveal weaknesses yet risk unrealistic behaviors without strict human-likeness constraints.

Knowledge-based methods. Template-driven frameworks encode traffic laws and interpretable parameters. CommonRoad [1] and CARLA Scenario Runner [9] define canonical tasks (e.g., unprotected turns, red-light running). Such approaches provide structure and reproducibility but struggle to cover the diversity of stochastic multi-agent interactions in real traffic.

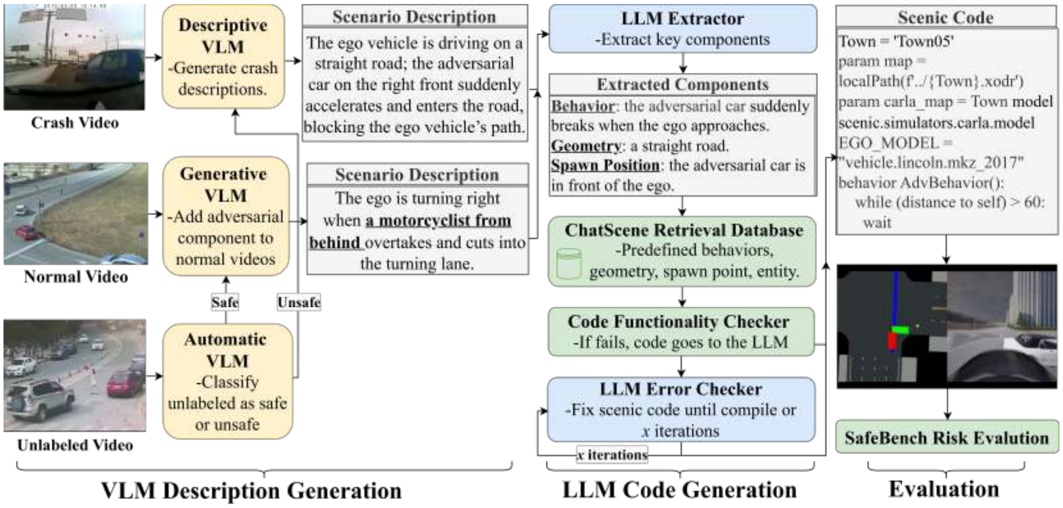
LLM-based scenario generation. Recent work uses LLMs to automate safety-critical scene generation. SafeDrive [53] and ChatScene [49] produce semantically grounded scenes; LeGEND [40] maps text to executable simulation code; ChatSUMO [30] integrates language control into SUMO; GOOSE [45] and KING [52] steer agents toward risky states; and Karacik [26] combines crash reports with RAG and Scenic correction. These systems offer controllability but face scalability and execution-robustness challenges.

Vision-language models in autonomous driving. VLMs support perception, reasoning, and planning in driving [44, 47, 39, 50]. Applications include scene understanding [37, 4, 6, 5], lightweight decision models [36], hierarchical QA [33], and planning integration [11, 25]. Pedestrian-centric models [23] and VLADBench [32] highlight multimodal reasoning. Yet reliably converting VLM descriptions into executable simulation programs remains open.

Reinforcement learning for evaluation. RL agents are widely used for safety-critical testing. SafeBench [46] standardizes evaluation in CARLA, while PPO [38], SAC [24], and TD3 [22] serve as strong baselines. D2RL [21] increases exposure to critical events. RL-based evaluation provides quantitative evidence of policy robustness under generated scenarios.

Our work differs from prior efforts by combining VLM-based scene grounding, LLM-driven Scenic synthesis, and automatic execution-repair loops to scale realistic safety-critical scenario generation from real driving videos, including both crash and non-crash data.

3 Method


Crash Video
Normal Video
Unlabeled Video
Descriptive VLM
-Generate crash descriptions.
Generative VLM
-Add adversarial component to normal videos
Automatic VLM
-Classify unlabeled as safe or unsafe
Safe
Unsafe
Scenario Description
Scenario Description
LLM Extractor
-Extract key components
Extracted Components
ChatScene Retrieval Database
-Predefined behaviors, geometry, spawn point, entity.
Code Functionality Checker
-If fails, code goes to the LLM
LLM Error Checker
-Fix scenic code until compile or x iterations
x iterations
Scenic Code
SafeBench Risk Evalution
VLM Description Generation
LLM Code Generation
Evaluation

The ego vehicle is driving on a straight road; the adversarial car on the right front suddenly accelerates and enters the road, blocking the ego vehicle's path.

The ego is turning right when **a motorcyclist from behind** overtakes and cuts into the turning lane.

Behavior: the adversarial car suddenly breaks when the ego approaches.
Geometry: a straight road.
Spawn Position: the adversarial car is in front of the ego.

```
Town = 'Town05'
param map =
localPath(f'../{Town}.xodr')
param carla_map = Town model
scenic.simulators.carla.model
EGO_MODEL =
"vehicle.lincoln.mkz_2017"
behavior AdvBehavior():
    while (distance to self) > 60:
      wait
```

Figure 1: Overview of the proposed pipeline with main components of VLM description generation, LLM code generation and evaluation. We support three VLM modes based on video context: (i) **Descriptive** for generating description from crash videos, (ii) **Generative** for adding adversarial components to normal driving video descriptions, and (iii) **Automatic** for classifying unlabeled videos and feeding into the corresponding mode.

Our goal is to generate diverse and realistic safety-critical driving scenarios directly from traffic videos, covering both crash and non-crash data. Unlike prior work that focuses only on replaying accidents or manually crafting adversarial cases, our framework converts naturalistic visual inputs into executable Scenic programs and verifies them in CARLA through an automated correction loop.

3.1 Overview

The core intuition of our approach is to leverage the abundance of real-world driving footage and automatically transform it into structured scenarios. Accident data provides valuable high-risk events but is limited in scale, whereas normal driving footage is abundant but lacks critical moments. By designing a pipeline that can operate flexibly across crash videos, normal traffic footage, and unlabeled scenes, we expand the possibility of generating diverse, realistic, and safety-critical scenarios well beyond the scope of the original ChatScene framework. Importantly, our approach does more than simply create arbitrary adversarial cases: by grounding the scenario generation process in the data distribution of the input dataset, it preserves the traffic characteristics and behavioral patterns that are present in the source. This ensures that the generated safety-critical scenarios are not only challenging but also representative of the environments from which the data was drawn.

Our system consists of three complementary modes of operation — Descriptive, Generative, and Automatic — which differ in how they handle the initial input but share the same downstream pipeline. In all cases, the visual input is processed by a vision-language model (VLM), specifically Qwen2.5-VL 3B, refined by a large language model (LLM) into structured components, converted into Scenic code through a retrieval database, validated via an error-correction loop powered by the Llama 3.3 70B API, and ultimately executed in CARLA for reinforcement learning (RL) evaluation. This modular structure ensures that our framework remains flexible and extensible, while always producing valid safety-critical scenarios.

Pipeline summary. Given an input video, the system:

1. extracts high-level scene descriptions using a VLM (Qwen2.5-VL 3B),
2. refines descriptions into structured scenario elements using an LLM (Llama 3.3 70B),
3. retrieves Scenic code templates and composes a full script,
4. verifies and repairs the script through an automated correction loop (Llama 3.3 70B),
5. executes the scenario in CARLA and outputs simulation for evaluation.

This modular design supports three modes of operation (Fig. 1): *descriptive*, *generative*, and *automatic*. All modes share the same backend, differing in prompts used to derive scenario descriptions from input videos.

3.2 Scenario Generation Modes

Descriptive mode. This mode processes crash videos or accident image sequences. A VLM first produces a high-level narrative describing causal relations and key events (e.g., a sudden brake leading to a rear-end collision), similar to the ChatScene descriptive pipeline. An LLM then extracts structured elements (ego behavior, adversarial agents, road geometry, environment) and retrieves matching Scenic templates from a curated library. The composed Scenic code is compiled and validated in CARLA, with iterative correction until successful execution, producing faithful simulations of real-world accidents.

Generative mode. To leverage abundant non-accident footage, this mode augments normal scenes with context-appropriate adversarial events. The VLM proposes realistic hazards (e.g., red-light running at intersections or abrupt pedestrian entry), preserving the underlying traffic distribution and regional driving norms. Intersection-dominant datasets yield intersection conflicts, while cultural rules (e.g., right-turn-on-red differences across regions) naturally emerge. This produces diverse and realistic safety-critical scenarios grounded in natural traffic context.

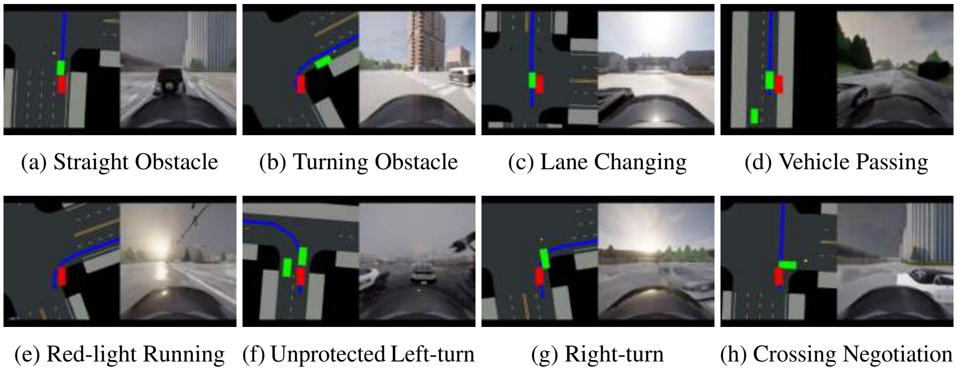
Automatic mode. The automatic mode classifies each input as crash or normal traffic and routes it accordingly: crash scenes use the descriptive pipeline, normal scenes the generative pipeline. Regardless of classification accuracy, all outputs remain safety-critical: misclassified crashes still result in valid adversarial scenarios, and misclassified normal scenes are enriched with hazards. This enables fully autonomous large-scale scenario generation from unlabeled traffic data.

3.3 Scenic Code Generation

We follow ChatScene's retrieval-based approach for Scenic generation to ensure comparability. Natural-language scenario descriptions (one per safety-critical case) are parsed by the Llama 3.3 70B API into structured elements (ego intent, adversarial behaviors, road geometry, environment). These elements query a curated retrieval database of validated Scenic fragments and templates, which are composed into complete programs that specify both static layout (lanes, intersections, signals) and dynamic interactions (cut-ins, collisions, red-light events). This retrieval-augmented design grounds outputs in syntactically consistent Scenic patterns and reproducible code. Our pipeline differs from ChatScene only in the LLM backend (Llama 3.3 70B in place of ChatGPT) for cost efficiency; the remaining process is identical, enabling direct comparison.

3.4 LLM Error Checker

To guarantee executability, each generated Scenic program is compiled and run in CARLA. On failure, the system captures the code and full error trace and enters an automatic correction loop powered by Llama 3.3 70B. A single structured prompt supplies (i) the CARLA/Scenic error message and (ii) a specification of permissible Scenic operations (actions, attributes, parameter formats, agent/scenario definitions). The model proposes a patched program, which is immediately re-validated in CARLA; the loop iterates until success or a fixed attempt budget (e.g., five). Successful scripts are marked verified and forwarded to evaluation; unsuccessful ones are saved for later inspection. Integrating simulation-grounded verification with LLM-based repair provides robust, low-overhead correction of common syntactic and semantic issues and reduces manual debugging.


(a) Straight Obstacle (b) Turning Obstacle (c) Lane Changing (d) Vehicle Passing

(e) Red-light Running (f) Unprotected Left-turn (g) Right-turn (h) Crossing Negotiation

Figure 2: Generated scenarios of eight safety-critical base scenarios generated for evaluation. Each subfigure (a–h) illustrates a different scenario configuration.

4 Experiment

4.1 Dataset

We evaluate on three complementary real-world traffic datasets covering dashcam, surveillance, and roadside perspectives. **CCD** [27] contains crash and non-crash dashcam clips with frame-level labels and diverse collision types. **TADS** [10] provides surveillance accident videos with precise temporal boundaries and human eye-gaze maps. **TUMTraf** [35, 12–16] offers multi-modal roadside data with real crashes, 3D annotations, and cooperative-perception views. Together, these datasets span complementary viewpoints and conditions for evaluating safety-critical scenario generation.

4.2 Data Preprocessing

We apply lightweight preprocessing to ensure efficient multimodal inference on a single NVIDIA 2080 Ti GPU. Images are resized with bicubic interpolation such that the longer side is at most 448 pixels while preserving aspect ratio. Videos are truncated to a fixed maximum length using `ffmpeg` without re-encoding, ensuring consistent clip duration and reduced memory usage. These steps standardize input scale across modalities and allow Qwen2.5-VL 3B to operate reliably under limited VRAM.

For dataset adaptation, we split **CCD** into crash (**CCD-C**, $1{,}500$ clips) and non-crash (**CCD-N**, $3{,}000$ clips) subsets. For **TADS**, we downsample by selecting one frame every ten and group every 10 frames into a scene, yielding $\sim 2{,}599$ scenes from 966 videos. For **TUMTraf**, we merge infrastructure subsets into **Tumtraf-I** and apply the same frame grouping, producing $\sim 7{,}000$ scenes. Video clips form **Tumtraf-V** ($\sim 5{,}000$ scenes). In total, preprocessing yields ~ 46k scenes for scenario generation and evaluation.

4.3 Base Scenarios

Following ChatScene, we evaluate across eight representative safety-critical scenarios: Straight Obstacle, Turning Obstacle, Lane Changing, Vehicle Passing, Red-light Running, Unprotected Left-turn, Right-turn, and Crossing Negotiation. These scenarios span core urban interactions and capture diverse collision-prone conditions for rigorous stress-testing of autonomous policies.

4.4 Evaluation Metrics

We adopt the SafeBench [46] protocol, which provides standardized metrics for safety, functionality, and driving etiquette. Safety measures capture violations such as collisions and off-road events; functionality measures task completion and efficiency; etiquette measures smoothness and comfort. An overall score summarizes performance, ensuring comparability with prior work. Detailed individual metrics and weights are listed in Appendix.

Metric	Method	Scenario Categories								Average
		Straight Obstacle	Turning Obstacle	Lane Changing	Vehicle Passing	Red-light Running	Unprotected Left-turn	Right-Turn	Crossing Negotiation	
↓ **RF**	ChatScene	0.895	0.924	**0.600**	**0.675**	0.899	0.863	0.868	0.939	**0.833**
	ChatScene-S	–	–	–	–	–	–	–	–	–
	NHTSA	0.906	0.929	0.849	0.887	0.907	**0.782**	0.884	**0.699**	0.855
	NHTSA-S	–	–	–	–	–	–	–	–	–
	CCD-C	**0.834**	0.929	0.734	0.919	0.943	0.919	0.814	0.908	0.875
	CCD-N	0.935	**0.665**	0.852	0.944	0.903	0.894	**0.652**	0.862	0.838
	TADS	0.894	0.898	0.784	0.850	**0.844**	0.938	0.899	0.780	0.861
	Tumtraf-I	0.851	0.776	0.832	0.930	0.929	0.851	0.753	0.887	0.851
	Tumtraf-V	0.933	0.847	0.872	0.868	0.951	0.932	0.603	0.797	0.850
	VLM-ALL	0.889	0.803	0.815	0.902	0.934	0.907	0.741	0.847	0.862
↓ **Comp**	ChatScene	0.194	0.264	0.466	0.583	0.160	1.000	0.500	**0.090**	0.407
	ChatScene-S	–	–	–	–	–	–	–	–	–
	NHTSA	0.170	0.252	0.540	1.000	0.130	0.220	0.661	0.333	0.413
	NHTSA-S	–	–	–	–	–	–	–	–	–
	CCD-C	**0.110**	**0.149**	0.551	0.500	0.129	0.328	0.619	**0.129**	0.314
	CCD-N	0.335	0.414	0.259	**0.126**	0.525	0.660	0.548	0.404	0.409
	TADS	0.616	1.000	0.422	1.000	1.000	**0.150**	0.666	0.500	0.669
	Tumtraf-I	0.145	0.181	0.301	0.252	0.168	0.653	**0.361**	0.331	**0.299**
	Tumtraf-V	0.124	0.438	**0.232**	0.218	**0.029**	0.228	0.640	0.591	0.313
	VLM-ALL	0.266	0.436	0.353	0.419	0.370	0.442	0.563	0.391	0.441

Table 1: Scenario-generation statistics for pretrained ego agent **SAC**: route following stability (RF) and route completion rate (Comp) across datasets and scenario categories.

We evaluate agents using five complementary metrics capturing safety and functionality. **Route Following Stability (RF)** measures average deviation from the reference path, reflecting control robustness under disturbances. **Route Completion (Comp)** quantifies the fraction of the planned route completed, decreasing when collisions or blockages prevent progress. **Safety Overall Score (SOS)** aggregates safety violations, including collisions, red-light and stop-sign running, and off-road events, with lower values indicating higher risk exposure. **Task Overall Score (TOS)** combines stability, completion, and time to success to assess functional performance under adversarial conditions. Finally, **Overall Score (OS)** integrates safety and task dimensions into a single robustness measure, indicating scenarios where agents exhibit both safety violations and degraded task performance.

4.5 Setup

To evaluate the adversarial strength and complexity of the generated scenarios, we follow ChatScene and curate five representative cases per dataset across eight scenario types. Each case is converted into a static Scenic program, where probabilistic sampling of positions, velocities, accelerations, and timing enables diverse instantiations from a single template. For each dataset, five scenarios per type are evaluated over 10 routes and three RL-based ego agents (SAC, PPO, TD3), with 50 executions per scenario, resulting in

$$\begin{aligned}
&8\ (\text{base scenarios}) \times 5\ (\text{scenarios per category}) \times 50\ (\text{executions per scenario}) \\
&\times 10\ (\text{routes}) \times 3\ (\text{ego agents}) \times 5\ (\text{datasets}) \\
&= 300{,}000
\end{aligned}$$

We report **VLM-ALL** (aggregated performance across CCD-D, CCD-N, TADS, Tumtraf-I, Tumtraf-V) and **VLM-Top5** (top five most challenging scenarios). Baselines include ChatScene and NHTSA [26]; since both originally report static results, we additionally include **ChatScene-S** and **NHTSA-S**. Missing metrics are denoted "–". Best values are **bold-underlined**, and VLM-Top5 is excluded from best-value marking to avoid bias. Each method group is separated with horizontal line. We note that original ChatScene scenarios involve manual modifications; we report those values for completeness.

Metric	Method	Scenario Categories								Average
		Straight Obstacle	Turning Obstacle	Lane Changing	Vehicle Passing	Red-light Running	Unprotected Left-turn	Right-Turn	Crossing Negotiation	
↓ **SOS**	ChatScene	0.686	0.375	0.289	0.289	0.375	1.000	1.000	1.000	0.627
	ChatScene-S	–	–	–	–	–	–	–	–	–
	NHTSA	0.313	0.375	0.625	0.938	0.688	0.375	0.551	0.325	0.524
	NHTSA-S	–	–	–	–	–	–	–	–	–
	CCD-C	0.671	0.375	**0.287**	0.938	0.375	**0.250**	0.563	0.375	0.479
	CCD-N	**0.250**	0.298	0.368	0.375	0.500	0.875	0.556	0.998	0.528
	TADS	0.938	0.938	0.375	1.000	1.000	0.375	0.875	0.936	0.805
	Tumtraf-I	0.375	**0.245**	0.625	0.375	0.688	0.688	0.375	**0.250**	**0.452**
	Tumtraf-V	0.375	0.375	0.375	**0.248**	0.688	0.688	**0.257**	0.625	0.454
	VLM-ALL	0.522	0.446	0.436	0.579	0.650	0.565	0.591	0.637	0.549
↓ **TOS**	ChatScene	0.696	0.729	**0.689**	0.753	0.686	0.873	0.703	**0.676**	0.726
	ChatScene-S	–	–	–	–	–	–	–	–	–
	NHTSA	0.692	0.727	0.710	0.881	0.679	**0.667**	0.754	0.677	0.724
	NHTSA-S	–	–	–	–	–	–	–	–	–
	CCD-C	**0.648**	0.693	0.762	0.723	0.691	0.749	0.725	0.679	0.709
	CCD-N	0.757	0.693	0.704	**0.690**	0.727	0.768	**0.624**	0.755	0.715
	TADS	0.753	0.884	0.735	0.868	0.866	0.696	0.771	0.672	0.781
	Tumtraf-I	0.665	**0.652**	0.711	0.727	0.699	0.752	0.705	0.740	**0.706**
	Tumtraf-V	0.686	0.762	0.701	0.695	**0.660**	0.720	0.748	0.707	0.710
	VLM-ALL	0.702	0.731	0.723	0.747	0.729	0.737	0.715	0.711	0.724
↓ **OS**	ChatScene	0.692	0.454	**0.379**	0.377	0.449	0.969	0.952	0.942	0.652
	ChatScene-S	0.450	0.505	0.451	0.489	0.582	0.492	0.426	0.461	**0.482**
	NHTSA	0.392	0.445	0.645	0.922	0.645	0.433	0.593	0.404	0.560
	NHTSA-S	0.460	0.500	0.533	0.445	0.438	0.572	0.518	0.520	0.498
	CCD-C	0.679	0.413	0.387	0.905	**0.404**	**0.328**	0.600	0.423	0.517
	CCD-N	**0.342**	0.361	0.442	0.434	0.544	0.854	0.579	0.954	0.564
	TADS	0.902	0.922	0.453	0.969	0.970	0.445	0.854	0.893	0.801
	Tumtraf-I	0.427	**0.336**	0.647	0.455	0.697	0.702	0.450	**0.347**	0.508
	Tumtraf-V	0.435	0.455	0.434	**0.331**	0.693	0.670	**0.352**	0.642	0.502
	VLM-ALL	0.557	0.481	0.469	0.684	0.662	0.600	0.597	0.673	0.590

Table 2: Scenario-generation statistics for pretrained ego agent **SAC**: safety overall score (SOS), task overall score (TOS), and final overall score (OS) across datasets and scenario categories.

4.6 Evaluation Statistics with Ego Agent SAC

Table 1 reports the impact of generated scenarios on the pretrained SAC agent in terms of route following stability (RF) and route completion (Comp), where lower values indicate stronger disruption. At the dataset level, ChatScene yields the lowest average RF (0.833), showing consistent deviation from the reference path, while Tumtraf-I (0.299), Tumtraf-V (0.313), and CCD-C (0.314) result in the lowest completion rates, indicating frequent route failure. At the scenario level, Lane Changing and Vehicle Passing in ChatScene lead to the largest RF drops, while Turning Obstacle and Right-Turn in CCD-N expose instability in turning maneuvers. For completion, the most adversarial cases include Red-light Running in Tumtraf-V (0.029) and Crossing Negotiation in ChatScene (0.090), where success is rare. CCD-C also generates challenging settings, with Straight Obstacle (RF 0.834, Comp 0.110) and Turning Obstacle (RF 0.929, Comp 0.149) jointly degrading stability and completion. These results show that scenario generation surfaces distinct and complementary weaknesses, rather than uniformly degrading agent performance.

Table 2 reports the safety overall score (SOS), task overall score (TOS), and final overall score (OS) for the pretrained SAC agent, where lower values indicate stronger disruption. At the dataset level, the largest safety degradation occurs in Tumtraf-I (0.452) and Tumtraf-V (0.454), followed by CCD-C (0.479), indicating frequent collisions, violations, and off-road failures. Task performance is similarly affected, with the lowest TOS in CCD-C (0.709), Tumtraf-I (0.706), and Tumtraf-V (0.710), reflecting reduced route adherence and completion. Combined OS values again identify Tumtraf-V (0.502), Tumtraf-I (0.508), and CCD-C (0.517) as the most adversarial datasets.

At the scenario level, minima highlight different weaknesses: Straight Obstacle (0.342, CCD-N) and Turning Obstacle (0.336, Tumtraf-I) degrade performance in maneuvering, Vehicle Passing (0.331, Tumtraf-V) is most disruptive for functional planning, and Unprotected Left-turn (0.328, CCD-C) stresses multi-agent intersections. Overall, CCD-C, Tumtraf-I, and Tumtraf-V consistently produce the lowest scores, confirming their strength in exposing SAC vulnerabilities across safety, compliance, and trajectory execution.

4.7 Full SafeBench Metrics of Eight Base Scenarios

Method	**Safety Level**				**Functionality Level**			**Etiquette Level**			**OS ↓**
	CR ↑	RR ↑	SS ↑	OR ↑	RF ↓	Comp ↓	TS ↑	ACC ↑	YV ↑	LI ↑	
ChatScene	0.458	0.000	0.125	0.159	0.833	0.463	0.064	0.343	0.292	0.135	0.692
ChatScene-S	**0.831**	0.179	0.143	0.035	0.833	0.544	0.223	0.705	0.532	0.243	**0.482**
NHTSA	0.529	0.382	0.265	0.016	0.838	0.483	0.120	0.393	0.194	0.094	0.619
NHTSA-S	0.732	0.293	0.124	0.017	0.885	0.516	**0.173**	0.326	0.219	0.064	0.544
CCD-C	0.313	0.219	**0.469**	0.064	0.651	0.536	0.140	0.259	0.237	0.138	0.711
CCD-N	0.656	0.375	0.156	**0.140**	0.791	**0.359**	0.079	**0.448**	0.291	0.109	0.546
TADS	0.433	0.200	0.200	0.052	0.790	0.465	0.118	0.359	0.264	0.102	0.684
Tumtraf-I	0.407	0.389	0.463	0.008	0.643	0.497	0.117	0.378	0.161	0.106	0.652
Tumtraf-V	0.362	**0.431**	0.345	0.063	**0.600**	0.562	0.156	0.357	**0.370**	**0.140**	0.671
VLM-ALL	0.434	0.323	0.327	0.065	0.695	0.484	0.122	0.360	0.265	0.119	0.657
VLM-Top5	0.900	1.000	0.100	0.003	0.746	0.310	0.049	0.605	0.507	0.035	0.373

Table 3: Statistics of full SafeBench evaluation metrics across eight base scenarios. Bold underlined entries indicate best (for ↑) or worst (for ↓) in each column.

In 3, at the safety level, each dataset exposes different vulnerabilities. ChatScene-S yields the highest collision rate (CR = 0.831), while Tumtraf-V induces the most red-light violations (RR = 0.431). CCD-C causes the most stop-sign failures (SS = 0.469), and ChatScene results in the largest out-of-road deviation (OR = 0.159), indicating greater loss of lane control.

At the functionality level, adversarial difficulty appears through reduced stability and progress. Tumtraf-V has the lowest route-following stability (RF = 0.600), CCD-N shows the lowest route completion (Comp = 0.359), and NHTSA-S leads to the longest task time (TS = 0.173).

At the etiquette level, disturbances produce uneven and abrupt driving. CCD-N has the highest acceleration variance (ACC = 0.448), while Tumtraf-V yields the highest yaw rate (YV = 0.370) and most lane invasions (LI = 0.140).

Finally, ChatScene-S records the lowest overall score (OS = 0.482), followed by CCD-N (0.546) and NHTSA (0.619), indicating the strongest adversarial effect. These results show that different datasets stress distinct aspects of autonomous driving, from safety failures to control instability and degraded etiquette.

4.8 Full SafeBench Metrics of Three Ego Agents

In 4, at the safety level, SAC shows the highest collision rate (CR $= 0.518$), red-light violations (RR $= 0.357$), and out-of-road distance (OR $= 0.218$), while PPO exhibits the most stop-sign violations (SS $= 0.288$). TD3 maintains comparatively lower safety violations.

For functionality, TD3 yields the lowest route completion (Comp $= 0.453$), SAC has the weakest route-following stability (RF $= 0.687$), and PPO requires the longest time to succeed (TS $= 0.155$).

In etiquette metrics, SAC shows the highest acceleration (ACC $= 0.402$), PPO the largest yaw velocity (YV $= 0.315$), and TD3 the most lane invasions (LI $= 0.172$).

Agent	Safety Level				Functionality Level			Etiquette Level			OS ↓
	CR ↑	RR ↑	SS ↑	OR ↑	RF ↓	Comp ↓	TS ↑	ACC ↑	YV ↑	LI ↑	
SAC	**0.518**	**0.357**	0.196	**0.218**	**0.687**	0.513	0.104	**0.402**	0.293	0.102	**0.607**
TD3	0.446	0.203	0.162	0.058	0.809	**0.453**	0.097	0.346	0.108	**0.172**	0.684
PPO	0.500	0.308	**0.288**	0.016	0.775	0.500	**0.155**	0.394	**0.315**	0.092	0.632

Table 4: Statistics of full SafeBench evaluation metrics on all three pretrained ego agents with composition of all VLM datasets .

Aggregating all metrics, SAC achieves the lowest overall score (OS $= 0.607$), followed by PPO (0.632), with TD3 performing best overall (0.684). Overall, SAC is most vulnerable to adversarial scenarios, PPO struggles with rule-following and efficiency, and TD3 demonstrates relatively stronger robustness while still experiencing degradation in completion and etiquette.

Importantly, our approach remains effective across all three agents, revealing consistent failure patterns and demonstrating robustness in adversarial scenario generation under diverse policy behaviors.

4.9 Key Findings

Our generated safety-critical scenarios consistently destabilize pretrained RL agents and expose weaknesses beyond standard benchmarks. Performance varies by agent and dataset, revealing vulnerabilities at multiple levels.

Agent behavior. SAC is most affected, especially in rule- and interaction-heavy settings (e.g., red-light running, crossing negotiation), with sharp drops in route following and completion. TD3 incurs fewer safety violations but struggles with turning and lane-changing, often failing to complete routes. PPO shows broad vulnerabilities, particularly in lane-changing, right-turn, and red-light scenarios, where safety and task performance degrade jointly.

Dataset effects. CCD-N produces the most disruptive scenarios overall, strongly reducing completion and etiquette performance. TumTraf-I/V also cause substantial degradation, particularly in intersection and multi-agent scenes. CCD-C and ChatScene are less adversarial on average but still surface targeted weaknesses, while TADS shows moderate but occasionally strong impacts. Notably, our approach outperforms ChatScene by generating more diverse, semantically aligned, and distribution-preserving adversarial scenarios, revealing richer agent failure patterns and more consistently stressing safety-critical behaviors.

Together, these findings show that our generated scenarios successfully expose diverse weaknesses across agents and datasets. SAC was most destabilized, PPO showed major rule-following issues, and TD3—though steadier—struggled with completion and etiquette. Dataset variation further reveals that CCD-N and Tumtraf create especially challenging, realistic conditions, while others surface more specific failure modes. Combining them yields a comprehensive evaluation across safety, functionality, and etiquette dimensions.

5 Conclusion

We introduced a pipeline that uses vision–language models (VLMs) and large language models (LLMs) to automatically convert real driving videos into safety-critical simulation scenarios. Our system performs scenario decomposition, Scenic code synthesis, and automated error correction to transform naturalistic traffic scenes into executable CARLA simulations with minimal human input. The generated scenarios are realistic, diverse, and adversarial, capturing nuanced interactions and exposing complementary weaknesses across SAC, TD3, and PPO in SafeBench evaluations. Overall, our results show that multimodal foundation models can serve as scalable tools for constructing rich autonomous-driving test suites, advancing scenario-based evaluation and supporting safer deployment.

References

[1] Matthias Althoff, Markus Koschi, and Stephan Manzinger. Commonroad: Driving scenario database and simulation framework. *IV*, pages 2190–2197, 2017.

[2] Julian Bock, Robert Krajewski, Thomas Moers, Sebastian Runde, Lukas Vater, and Lutz Eckstein. The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections. In *IV*, pages 1090–1095. IEEE, 2019.

[3] Hu Cao, Guang Chen, Jiahao Xia, Genghang Zhuang, and Alois Knoll. Fusion-based feature attention gate component for vehicle detection based on event camera. *IEEE Sensors Journal*, 21(21):24540–24548, 2021. doi: 10.1109/JSEN.2021.3115016.

[4] Hu Cao, Guang Chen, Zhijun Li, Yingbai Hu, and Alois Knoll. Neurograsp: Multimodal neural network with euler region regression for neuromorphic vision-based grasp pose estimation. *IEEE Transactions on Instrumentation and Measurement*, 71:1–11, 2022. doi: 10.1109/TIM.2022.3179469.

[5] Hu Cao, Guang Chen, Hengshuang Zhao, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Alois Knoll. Sdpt: Semantic-aware dimension-pooling transformer for image segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 25(11):15934–15946, 2024. doi: 10.1109/TITS.2024.3417813.

[6] Hu Cao, Zhongnan Qu, Guang Chen, Xinyi Li, Lothar Thiele, and Alois Knoll. Ghostvit: Expediting vision transformers via cheap operations. *IEEE Transactions on Artificial Intelligence*, 5 (6):2517–2525, 2024. doi: 10.1109/TAI.2023.3326795.

[7] Hu Cao, Zehua Zhang, Yan Xia, Xinyi Li, Jiahao Xia, Guang Chen, and Alois Knoll. Embracing events and frames with hierarchical feature refinement network for object detection. In *European Conference on Computer Vision*, pages 161–177. Springer, 2024.

[8] CARLA Autonomous Driving Leaderboard. Traffic scenarios – carla autonomous driving leaderboard. `https://leaderboard.carla.org/scenarios/`, 2025. Accessed: 2025-11-04.

[9] CARLA Simulator. Carla scenario runner. `https://github.com/carla-simulator/scenario_runner`, 2019.

[10] Yachuang Chai, Jianwu Fang, Haoquan Liang, and Wushouer Silamu. Tads: a novel dataset for road traffic accident detection from a surveillance perspective. *The Journal of Supercomputing*, 80:26226–26249, 2024. doi: 10.1007/s11227-024-06429-7.

[11] Xuesong Chen, Linjiang Huang, Tao Ma, Rongyao Fang, Shaoshuai Shi, and Hongsheng Li. Solve: Synergy of language-vision and end-to-end networks for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. doi: 10.48550/arXiv.2505.16805. URL `https://arxiv.org/abs/2505.16805`.

[12] TUMTraf Consortium. Tumtraf-a: A highway accident dataset for roadside infrastructure perception. In *IEEE ITSC*, 2023.

[13] TUMTraf Consortium. Tumtraf-i: An intersection dataset for roadside infrastructure-based 3d perception. In *IEEE ITSC*, 2023.

[14] TUMTraf Consortium. Tumtraf-e: Calibration and fusion dataset for roadside event-based and rgb cameras. *Sensors*, 2024.

[15] TUMTraf Consortium. Tumtraf-v2x: A cooperative roadside and onboard perception dataset. *arXiv preprint arXiv:2403.01316*, 2024.

[16] TUMTraf Consortium. Tumtraf-videoqa: Video question answering and grounding dataset for roadside perception. *arXiv preprint*, 2024.

[17] Wenhao Ding, Baiming Chen, Minjun Xu, and Ding Zhao. Learning to collide: An adaptive safety-critical scenarios generating method. In *IROS*, pages 2243–2250. IEEE, 2020.

[18] Wenhao Ding, Chejian Xu, Mansur Arief, Haohong Lin, Bo Li, and Ding Zhao. A survey on safety-critical driving scenario generation—a methodological perspective. *IEEE Transactions on Intelligent Transportation Systems*, 24(12):6971–6988, 2022. doi: 10.1109/TITS.2023.3259322. URL `https://doi.org/10.1109/TITS.2023.3259322`.

[19] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *CoRL*, pages 1–16, 2017. URL `https://arxiv.org/abs/1711.03938`.

[20] Shuo Feng, Xintao Yan, Haowei Sun, Yiheng Feng, and Henry X. Liu. Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment. *Nature Communications*, 12:748, 2021. doi: 10.1038/s41467-021-21007-8.

[21] 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. doi: 10.1038/s41586-023-05732-2.

[22] 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 (ICML)*, 2018. doi: 10.48550/arXiv.1802.09477. URL `https://arxiv.org/abs/1802.09477`.

[23] Haoxiang Gao and Yu Zhao. Application of vision-language model to pedestrians behavior and scene understanding in autonomous driving. *arXiv*, 2025. doi: 10.48550/arXiv.2501.06680. URL `https://arxiv.org/abs/2501.06680`.

[24] 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 (ICML)*, 2018. doi: 10.48550/arXiv.1801.01290. URL `https://arxiv.org/abs/1801.01290`.

[25] Zilin Huang, Zihao Sheng, Yansong Qu, Junwei You, and Sikai Chen. Vlm-rl: A unified vision language models and reinforcement learning framework for safe autonomous driving. *arXiv*, 2024. doi: 10.48550/arXiv.2412.15544. URL `https://arxiv.org/abs/2412.15544`.

[26] Nigar Doga Karacik. Enhancing safety models for autonomous driving in critical scenarios. Unpublished manuscript, 2025.

[27] Jinkyu Kim et al. Car crash dataset (ccd). `https://github.com/Cogito2012/CarCrashDataset`, 2022.

[28] Mark Koren and Matthias Althoff. Adaptive stress testing for autonomous vehicles. In *IV*, pages 1–7. IEEE, 2018.

[29] Robert Krajewski, Julian Bock, Lennart Kloeker, and Lutz Eckstein. The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways. In *ITSC*, pages 2118–2125. IEEE, 2018.

[30] Jian Li, Ming Xu, Lei Zhao, and Kai Chen. Chatsumo: Large language model for automating traffic scenario generation in simulation of urban mobility. *arXiv preprint arXiv:2406.17890*, 2024.

[31] Wei Li, Chengwei Pan, Rong Zhang, Jiaping Ren, Yuexin Ma, Jin Fang, Feilong Yan, Qichuan Geng, Xinyu Huang, Huajun Gong, Weiwei Xu, Guoping Wang, Dinesh Manocha, and Ruigang Yang. Aads: Augmented autonomous driving simulation using data-driven algorithms. *Science Robotics*, 4(28):eaaw0863, 2019. doi: 10.1126/scirobotics.aaw0863.

[32] Yue Li, Meng Tian, Zhenyu Lin, Jiangtong Zhu, Dechang Zhu, Haiqiang Liu, Zining Wang, Yueyi Zhang, Zhiwei Xiong, and Xinhai Zhao. Fine-grained evaluation of large vision-language models in autonomous driving (vladbench). *arXiv*, 2025. doi: 10.48550/arXiv.2503.21505. URL `https://arxiv.org/abs/2503.21505`.

[33] Safaa Abdullahi Moallim Mohamud, Minjin Baek, and Dong Seog Han. Hierarchical question-answering for driving scene understanding using vision-language models. *arXiv*, 2025. doi: 10.48550/arXiv.2506.02615. URL `https://arxiv.org/abs/2506.02615`.

[34] Mateusz Niedoba, Frederik Naujoks, and Adrien Gaidon. High-fidelity simulation of rare events in autonomous driving. In *IV*, pages 1509–1516. IEEE, 2019.

[35] TUM Providentia++ Project. Tumtraf devkit. `https://github.com/tum-traffic-dataset/tum-traffic-dataset-dev-kit`, 2023.

[36] Zhijie Qiao, Haowei Li, Zhong Cao, and Henry X. Liu. Lightemma: Lightweight end-to-end multimodal model for autonomous driving. *arXiv*, 2025. doi: 10.48550/arXiv.2505.00284. URL `https://arxiv.org/abs/2505.00284`.

[37] Lübberstedt Jannik Rivera Esteban, Nico Uhlemann, and Markus Lienkamp. Scenario understanding of traffic scenes through large vision-language models. *arXiv*, 2025. doi: 10.48550/arXiv.2501.17131. URL `https://arxiv.org/abs/2501.17131`.

[38] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017. doi: 10.48550/arXiv.1707.06347. URL `https://arxiv.org/abs/1707.06347`.

[39] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. *arXiv*, 2023. doi: 10.48550/arXiv.2312.14150. URL `https://arxiv.org/abs/2312.14150`.

[40] Shuncheng Tang, Zhenya Zhang, Jixiang Zhou, Lei Lei, Yuan Zhou, and Yinxing Xue. Legend: A top-down approach to scenario generation of autonomous driving systems assisted by large language models. In *ASE*, 2024.

[41] Yu Tian et al. Tokenize the world into object-level knowledge to address long-tail events in autonomous driving. *arXiv preprint arXiv:2407.00959*, 2024. URL `https://arxiv.org/abs/2407.00959`.

[42] US Department of Transportation. Next generation simulation (ngsim) vehicle trajectories and supporting data. `https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm`, 2007.

[43] Jingkang Wang, Ava Pun, James Tu, Sivabalan Manivasagam, Abbas Sadat, Sergio Casas, Mengye Ren, and Raquel Urtasun. Advsim: Generating adversarial scenarios for self-driving vehicles. In *CVPR*, pages 9909–9918. IEEE, 2021.

[44] Licheng Wen, Xuemeng Yang, Daocheng Fu, Xiaofeng Wang, Pinlong Cai, Xin Li, Tao Ma, Yingxuan Li, Linran Xu, Dengke Shang, Zheng Zhu, Shaoyan Sun, Yeqi Bai, Xinyu Cai, Min Dou, Shuanglu Hu, Botian Shi, and Yu Qiao. On the road with gpt-4v(ision): Early explorations of visual-language models on autonomous driving. *arXiv*, 2023. doi: 10.48550/arXiv.2311.05332. URL `https://arxiv.org/abs/2311.05332`.

[45] Bowen Xie, Zekai Li, Yu Jiang, Mingli Song, and Yue Wang. Goose: Goal-conditioned reinforcement learning for safety-critical scenario generation. *arXiv preprint arXiv:2407.02467*, 2024.

[46] Junjie Xu et al. Safebench: A benchmarking platform for safety evaluation of autonomous vehicles. In *NeurIPS Datasets and Benchmarks*, 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/a48ad12d588c597f4725a8b84af647b5-Paper-Datasets_and_Benchmarks.pdf`.

[47] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K. Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *arXiv*, 2023. doi: 10.48550/arXiv.2310.01412. URL `https://arxiv.org/abs/2310.01412`.

[48] Yuan Yin, Pegah Khayatan, Éloi Zablocki, Alexandre Boulch, and Matthieu Cord. Regents: Real-world safety-critical driving scenario generation made stable. arXiv preprint arXiv:2409.07830, 2024. Available at `https://arxiv.org/abs/2409.07830`.

[49] Shanhe You, Xuewen Luo, Xinhe Liang, Jiashu Yu, Chen Zheng, and Jiangtao Gong. A comprehensive llm-powered framework for driving intelligence evaluation. *arXiv preprint arXiv:2503.05164*, 2025. doi: 10.48550/arXiv.2503.05164.

[50] Jie Zhang, Xin Li, Bo Wang, and Huazhe Yu. Think-driver: From driving-scene understanding to decision-making with vision language models. *arXiv*, 2024. doi: 10.48550/arXiv.2405.09876. URL `https://arxiv.org/abs/2405.09876`.

[51] 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.

[52] Wei Zhang, Ling Chen, Hao Ma, Rui Zhao, and Bo Wang. King: Generating safety-critical driving scenarios for robust imitation via kinematics gradients. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[53] Zhiyuan Zhou, Heye Huang, Boqi Li, Shiyue Zhao, Yao Mu, and Jianqiang Wang. Safedrive: Knowledge- and data-driven risk-sensitive decision-making for autonomous vehicles with large language models. *arXiv preprint arXiv:2412.13238*, 2024.

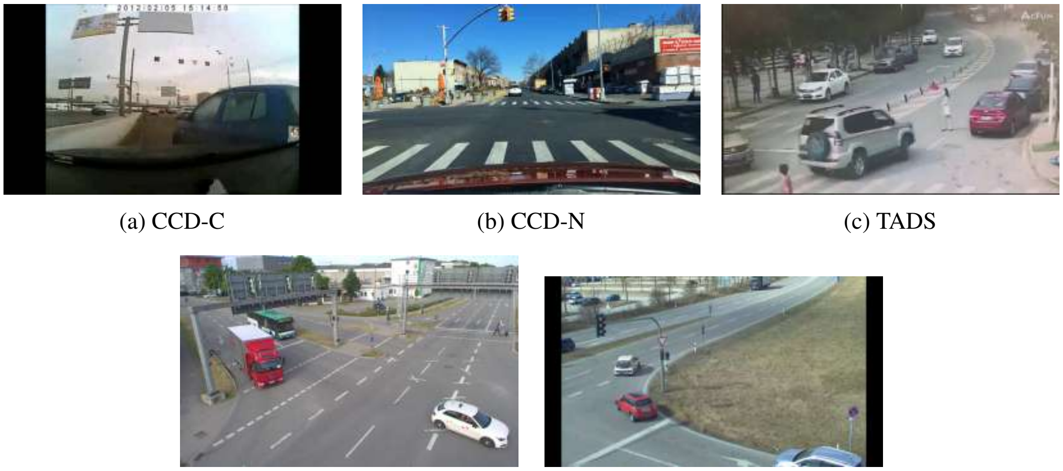
A Appendix

A.1 Hardware

We perform all VLM inference using the Qwen2.5-VL 3B model on a single NVIDIA RTX 2080 Ti GPU, without distributed computation or large-scale servers.

A.2 Dataset Example

We include an example frame from each dataset 3.



(a) CCD-C

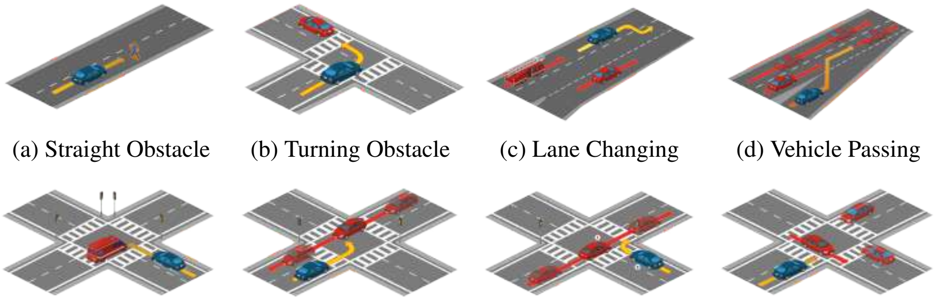
(b) CCD-N

(c) TADS

(d) Tumtraf-I

(e) Tumtraf-V

Figure 3: Example frames of five datasets. Frames of CCD-C, CCD-N and Tumtraf-V are extracted from original videos whereas frames of TADS and Tumtraf-I are example frames of image sequences.



(a) Straight Obstacle (b) Turning Obstacle (c) Lane Changing (d) Vehicle Passing

(e) Red-light Running (f) Unprotected Left-turn (g) Right-turn (h) Crossing Negotiation

Figure 4: Examples of eight safety-critical base scenarios. Each subfigure (a–h) illustrates a different scenario configuration [8].

A.3 Detailed Metrics

- **Safety level**

  Collision rate (CR). Average number of collisions per scenario.

  $$\mathrm{CR} \;=\; \mathbb{E}_{\tau\sim\mathcal{P}}\big[c(\tau)\big]$$

  where τ is a scenario trajectory sampled from a distribution $\mathcal{P}$, and $c(\tau)$ is the count of collisions observed in τ.

  Red-light running (RR). Average number of red-light violations per scenario.

  $$\mathrm{RR} \;=\; \mathbb{E}_{\tau\sim\mathcal{P}}\big[r(\tau)\big]$$

  where $r(\tau)$ is the number of times a red light is run in scenario τ.

  Stop-sign running (SS). Average number of missed stop signs per scenario.

  $$\mathrm{SS} \;=\; \mathbb{E}_{\tau\sim\mathcal{P}}\big[s(\tau)\big]$$

  where $s(\tau)$ is the number of stop signs not properly obeyed in scenario τ.

  Out-of-road distance (OR). Average distance driven outside the drivable area per scenario.

  $$\mathrm{OR} \;=\; \mathbb{E}_{\tau\sim\mathcal{P}}\big[d(\tau)\big]$$

  where $d(\tau)$ is the total distance traveled off-road in scenario τ.

- **Functionality level**

  Route following stability (RF). Average distance between the ego vehicle and the reference route during testing.

  $$\mathrm{RF} \;=\; 1 \;-\; \mathbb{E}_{\tau\sim\mathcal{P}}\left[\min\left(\frac{x(\tau)}{x_{\max}},\,1\right)\right]$$

  where $x(\tau)$ is the mean lateral/Euclidean deviation from the reference route in scenario τ, and $x_{\max} > 0$ is a chosen cap used for normalization.

  Route completion (Comp). Mean percentage of the route completed.

  $$\mathrm{Comp} \;=\; \mathbb{E}_{\tau\sim\mathcal{P}}\big[p(\tau)\big]$$

  where $p(\tau) \in [0, 100]$ is the route completion percentage in scenario τ.

  Time to succeed (TS). Mean time to finish scenarios that reach full completion.

  $$\mathrm{TS} \;=\; \mathbb{E}_{\tau\sim\mathcal{P}}\big[t(\tau)\,\big|\,p(\tau) = 100\%\big]$$

  where $t(\tau)$ is the elapsed time for scenario τ; the expectation is taken over scenarios that finish with $p(\tau) = 100\%$.

- **Etiquette level**

Symbol	Safety Level				Functionality Level			Etiquette Level		
	CR ↑	RR ↑	SS ↑	OR ↑	RF ↓	Comp ↓	TS ↑	ACC ↑	YV ↑	LI ↑
m^i_{max}	1	1	1	50	1	1	60	8	3	20
w^i	0.495	0.099	0.099	0.099	0.050	0.050	0.050	0.020	0.020	0.020

Table 5: Constants and weights used in SafeBench evaluation metrics. Metrics are grouped into **Safety Level**, **Functionality Level**, and **Etiquette Level**.

Average acceleration (ACC). Mean acceleration magnitude aggregated per scenario.

$$\text{ACC} \;=\; \mathbb{E}_{\tau\sim\mathcal{P}}\big[acc(\tau)\big]$$

where $acc(\tau)$ is an aggregate (e.g., mean or mean absolute) acceleration measure for scenario τ.

Yaw velocity (YV). Mean yaw-rate magnitude aggregated per scenario.

$$\text{YV} \;=\; \mathbb{E}_{\tau\sim\mathcal{P}}\big[y(\tau)\big]$$

where $y(\tau)$ is an aggregate yaw-rate measure for scenario τ.

Lane invasions (LI). Mean count of lane-boundary violations per scenario.

$$\text{LI} \;=\; \mathbb{E}_{\tau\sim\mathcal{P}}\big[l(\tau)\big]$$

where $l(\tau)$ is the number of lane invasions recorded in scenario τ.

- **Overall score (OS)**

  Weighted aggregation of normalized metrics.

$$\text{OS} \;=\; \sum_{i=1}^{10} w^i\, g\big(m^i\big)$$

where m^i is the i-th metric from $\{\text{CR}, \text{RR}, \text{SS}, \text{OR}, \text{RF}, \text{Comp}, \text{TS}, \text{ACC}, \text{YV}, \text{LI}\}$, $w^i \geq 0$ is its weight (often with $\sum_i w^i = 1$), and $g(\cdot)$ is the normalization:

$$g(m^i) \;=\; \begin{cases} \dfrac{m^i}{m^i_{\max}}, & \text{if the metric is higher-is-better,} \\ 1 - \dfrac{m^i}{m^i_{\max}}, & \text{if the metric is lower-is-better,} \end{cases}$$

with $m^i_{\max}$ a chosen upper bound for m^i that ensures $g(m^i) \in [0, 1]$.

A.4 Evaluation with PPO, TD3

We include the evaluation results of PPO and TD3 with the same metrics as SAC. We can 6 and 7 for ego agnet PPO and 8 and 9 for ego agent TD3.

Metric	**Method**	**Scenario Categories**								**Average**
		Straight Obstacle	**Turning Obstacle**	**Lane Changing**	**Vehicle Passing**	**Red-light Running**	**Unprotected Left-turn**	**Right-Turn**	**Crossing Negotiation**	
↓ **RF**	ChatScene	0.909	0.834	0.842	**0.786**	**0.808**	0.807	0.920	0.831	0.842
	ChatScene-S	–	–	–	–	–	–	–	–	–
	NHTSA	0.822	0.918	0.913	0.859	0.912	0.907	0.967	0.853	0.894
	NHTSA-S	–	–	–	–	–	–	–	–	–
	CCD-C	**0.805**	0.903	0.891	0.898	0.898	0.904	0.821	0.861	0.873
	CCD-N	0.918	**0.802**	**0.796**	0.919	0.917	0.795	**0.683**	0.931	0.845
	TADS	0.880	0.831	0.954	0.870	0.872	**0.544**	**0.441**	0.905	**0.787**
	Tumtraf-I	0.899	0.910	0.888	0.913	0.859	0.738	0.767	**0.631**	0.826
	Tumtraf-V	0.829	0.914	0.920	0.900	0.867	0.919	0.769	0.850	0.871
	VLM-ALL	0.890	0.872	0.890	0.900	0.902	0.860	0.776	0.847	0.867
↓ **Comp**	ChatScene	0.362	0.223	0.644	**0.154**	0.229	0.128	0.030	**0.064**	**0.229**
	ChatScene-S	–	–	–	–	–	–	–	–	–
	NHTSA	0.189	0.188	0.111	0.275	**0.125**	0.612	**0.010**	0.555	0.258
	NHTSA-S	–	–	–	–	–	–	–	–	–
	CCD-C	0.559	**0.129**	0.525	0.855	0.195	0.155	0.180	0.610	0.401
	CCD-N	0.918	0.802	0.796	0.919	0.917	0.795	0.683	0.931	0.845
	TADS	**0.067**	0.214	**0.060**	0.165	1.000	1.000	0.291	0.307	0.388
	Tumtraf-I	0.611	0.540	0.265	0.309	0.663	**0.114**	0.309	0.187	0.375
	Tumtraf-V	0.553	0.323	0.145	0.655	0.686	0.216	0.687	0.585	0.481
	VLM-ALL	0.542	0.402	0.358	0.581	0.692	0.456	0.430	0.524	0.498

Table 6: Scenario-generation statistics for pretrained ego agent **PPO**: route following stability (RF) and route completion rate (Comp) across datasets and scenario categories.

Metric	Method	Scenario Categories								Average
		Straight Obstacle	Turning Obstacle	Lane Changing	Vehicle Passing	Red-light Running	Unprotected Left-turn	Right-Turn	Crossing Negotiation	
↓ **SOS**	ChatScene	0.625	0.688	0.688	0.375	0.375	0.375	0.688	0.375	0.523
	ChatScene-S	–	–	–	–	–	–	–	–	–
	NHTSA	**0.375**	**0.313**	**0.250**	**0.313**	0.375	0.625	1.000	0.688	0.492
	NHTSA-S	–	–	–	–	–	–	–	–	–
	CCD-C	0.688	0.375	0.688	0.563	0.625	0.375	0.375	0.688	0.547
	CCD-N	**0.375**	0.375	0.375	0.375	**0.250**	0.375	0.313	1.000	**0.430**
	TADS	**0.375**	0.362	1.000	0.375	1.000	1.000	**0.250**	0.688	0.631
	Tumtraf-I	0.938	1.000	0.313	0.563	0.688	0.375	0.375	**0.250**	0.563
	Tumtraf-V	0.688	0.625	0.375	0.875	0.938	**0.313**	0.590	0.688	0.636
	VLM-ALL	0.613	0.547	0.550	0.550	0.700	0.488	0.381	0.663	0.561
↓ **TOS**	ChatScene	0.757	0.686	0.718	0.647	0.679	0.645	0.650	0.632	0.677
	ChatScene-S	–	–	–	–	–	–	–	–	–
	NHTSA	0.670	0.702	0.675	0.711	0.679	0.734	0.659	0.706	0.692
	NHTSA-S	–	–	–	–	–	–	–	–	–
	CCD-C	0.688	0.677	0.703	0.815	0.697	0.686	0.667	0.715	0.706
	CCD-N	0.722	**0.640**	0.673	0.679	**0.652**	0.671	0.612	0.678	**0.666**
	TADS	**0.649**	0.682	**0.671**	0.678	0.860	0.745	**0.577**	0.738	0.700
	Tumtraf-I	0.734	0.713	0.718	0.741	0.743	**0.617**	0.692	0.606	0.696
	Tumtraf-V	0.875	0.945	0.727	**0.406**	0.650	0.850	0.725	**0.581**	0.720
	VLM-ALL	0.734	0.731	0.698	0.664	0.720	0.714	0.655	0.664	0.698
↓ **OS**	ChatScene	0.634	0.679	0.700	0.440	0.435	0.443	0.686	0.438	0.557
	ChatScene-S	0.497	0.578	0.444	0.421	0.503	0.705	0.504	0.417	0.509
	NHTSA	**0.407**	**0.388**	**0.343**	**0.388**	0.429	0.650	0.944	0.696	0.531
	NHTSA-S	0.560	0.493	0.609	0.483	0.613	0.617	0.511	0.620	0.563
	CCD-C	0.690	0.434	0.682	0.617	0.653	0.436	0.440	0.693	0.581
	CCD-N	0.452	0.432	0.445	0.442	**0.338**	0.422	0.383	0.929	**0.480**
	TADS	0.426	0.433	0.948	0.444	0.961	0.959	**0.320**	0.692	0.648
	Tumtraf-I	0.903	0.953	0.402	0.608	0.693	0.434	0.446	**0.325**	0.596
	Tumtraf-V	0.699	0.662	0.442	0.828	0.892	**0.404**	0.616	0.684	0.653
	VLM-ALL	0.634	0.583	0.584	0.588	0.707	0.531	0.441	0.665	0.592

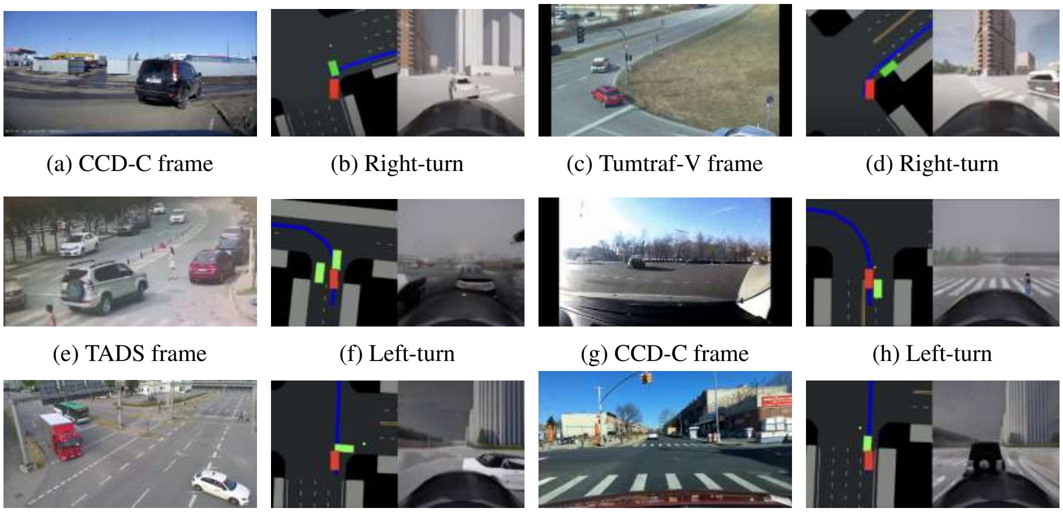
Table 7: Scenario-generation statistics for pretrained ego agent **PPO**: safety overall score (SOS), task overall score (TOS), and final overall score (OS) across datasets and scenario categories.

Metric	Method	Scenario Categories								Average
		Straight Obstacle	Turning Obstacle	Lane Changing	Vehicle Passing	Red-light Running	Unprotected Left-turn	Right-Turn	Crossing Negotiation	
↓ RF	ChatScene	0.890	0.948	0.928	0.871	0.920	0.866	0.930	0.821	0.897
	ChatScene-S	–	–	–	–	–	–	–	–	–
	NHTSA	0.884	0.911	0.931	0.826	0.919	0.930	0.860	0.918	0.897
	NHTSA-S	–	–	–	–	–	–	–	–	–
	CCD-C	**0.834**	0.934	0.900	**0.430**	**0.726**	0.927	0.870	0.886	**0.813**
	CCD-N	0.934	0.914	0.877	0.794	0.877	0.896	0.883	0.887	0.883
	TADS	0.931	**0.791**	**0.654**	0.944	0.928	0.893	0.873	0.895	0.864
	Tumtraf-I	0.924	0.908	0.797	0.932	0.936	**0.824**	**0.855**	0.921	0.887
	Tumtraf-V	0.925	0.930	0.848	0.915	0.874	0.925	0.864	**0.860**	0.893
	VLM-ALL	0.910	0.895	0.815	0.803	0.868	0.893	0.869	0.890	0.868
↓ Comp	ChatScene	1.000	**0.020**	0.244	0.294	0.307	**0.060**	**0.015**	1.000	0.368
	ChatScene-S	–	–	–	–	–	–	–	–	–
	NHTSA	1.000	0.194	0.111	0.266	0.100	0.537	1.000	**0.295**	0.438
	NHTSA-S	–	–	–	–	–	–	–	–	–
	CCD-C	0.594	0.515	0.858	0.425	0.373	0.338	0.540	0.545	0.524
	CCD-N	0.328	0.384	0.329	0.793	0.069	0.069	0.438	0.505	**0.364**
	TADS	0.470	0.179	0.974	**0.124**	**0.080**	0.597	0.546	0.460	0.429
	Tumtraf-I	**0.319**	0.575	0.594	0.305	0.164	1.000	0.515	0.589	0.508
	Tumtraf-V	0.994	0.358	**0.110**	0.575	0.571	0.503	0.532	0.708	0.544
	VLM-ALL	0.541	0.402	0.573	0.444	0.251	0.501	0.514	0.561	0.474

Table 8: Scenario-generation statistics for pretrained ego agent **TD3**: route following stability (RF) and route completion rate (Comp) across datasets and scenario categories.

Metric	Method	Scenario Categories								Average
		Straight Obstacle	Turning Obstacle	Lane Changing	Vehicle Passing	Red-light Running	Unprotected Left-turn	Right-Turn	Crossing Negotiation	
↓ SOS	ChatScene	0.938	1.000	0.375	0.662	0.625	0.375	0.375	1.000	0.669
	ChatScene-S	–	–	–	–	–	–	–	–	–
	NHTSA	0.875	**0.375**	**0.250**	**0.302**	0.683	0.938	0.868	0.625	0.615
	NHTSA-S	–	–	–	–	–	–	–	–	–
	CCD-C	0.688	0.688	0.938	1.000	1.000	0.375	0.625	0.688	0.750
	CCD-N	**0.313**	0.687	0.665	0.898	**0.307**	**0.313**	1.000	0.688	**0.609**
	TADS	1.000	**0.375**	1.000	0.375	1.000	1.000	0.688	**0.563**	0.750
	Tumtraf-I	0.938	0.938	0.292	0.563	1.000	1.000	0.688	0.625	0.755
	Tumtraf-V	0.813	0.938	0.364	0.563	0.688	0.625	**0.610**	0.966	0.696
	VLM-ALL	0.790	0.769	0.652	0.733	0.799	0.663	0.720	0.726	0.733
↓ TOS	ChatScene	0.877	**0.656**	0.724	0.722	0.742	**0.642**	**0.648**	0.868	0.735
	ChatScene-S	–	–	–	–	–	–	–	–	–
	NHTSA	0.831	0.702	0.681	0.697	0.673	0.823	0.875	0.738	0.752
	NHTSA-S	–	–	–	–	–	–	–	–	–
	CCD-C	**0.721**	0.712	0.834	**0.619**	0.700	0.755	0.683	0.733	0.720
	CCD-N	0.754	0.766	0.735	0.718	**0.649**	0.655	0.774	0.704	**0.719**
	TADS	0.801	0.657	0.791	0.689	0.669	0.830	0.734	0.785	0.744
	Tumtraf-I	0.748	0.827	0.797	0.746	0.700	0.870	0.707	**0.677**	0.759
	Tumtraf-V	0.873	0.762	**0.653**	0.739	0.742	0.810	0.710	0.856	0.768
	VLM-ALL	0.783	0.737	0.762	0.702	0.692	0.784	0.722	0.755	0.742
↓ OS	ChatScene	0.919	0.944	0.459	0.680	0.655	0.438	**0.440**	0.966	0.687
	ChatScene-S	0.462	0.483	0.407	0.410	0.527	0.483	0.493	**0.385**	**0.456**
	NHTSA	0.864	0.448	**0.348**	**0.388**	0.689	0.921	0.868	0.652	0.647
	NHTSA-S	0.592	0.503	0.568	0.496	0.573	0.643	0.539	0.654	0.570
	CCD-C	0.699	0.694	0.914	0.934	0.948	0.460	0.641	0.697	0.748
	CCD-N	**0.412**	0.707	0.686	0.855	**0.390**	**0.392**	0.954	0.698	0.637
	TADS	0.962	**0.442**	0.963	0.450	0.947	0.968	0.699	0.603	0.754
	Tumtraf-I	0.898	0.918	0.388	0.611	0.946	0.966	0.688	0.632	0.756
	Tumtraf-V	0.817	0.899	0.429	0.606	0.700	0.668	0.635	0.939	0.712
	VLM-ALL	0.757	0.772	0.680	0.691	0.780	0.689	0.723	0.733	0.729

Table 9: Scenario-generation statistics for pretrained ego agent **TD3**: safety overall score (SOS), task overall score (TOS), and final overall score (OS) across datasets and scenario categories.



(a) CCD-C frame (b) Right-turn (c) Tumtraf-V frame (d) Right-turn

(e) TADS frame (f) Left-turn (g) CCD-C frame (h) Left-turn

(i) TADS frame (j) Straight Obstacle (k) CCD-N frame (l) Straight Obstacle

Figure 5: Generated right-turn, left-turn and straight obstacle scenarios from dashcam imagery (CCD-C and CCD-N) and CCTV traffic recordings (TADS, Tumtraf-I and Tumtraf-V), demonstrating consistent scene translation across viewpoints and environments.

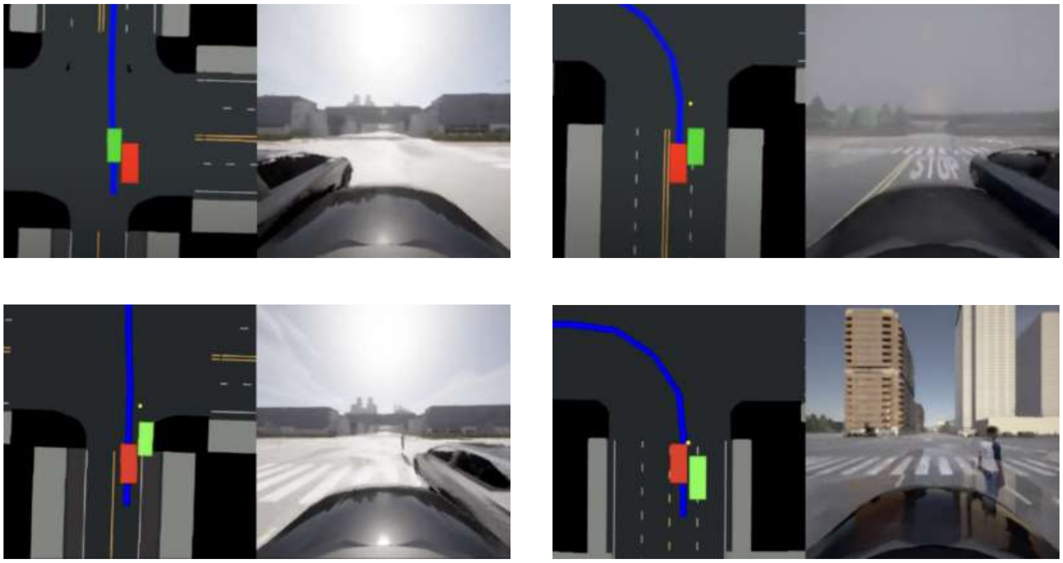


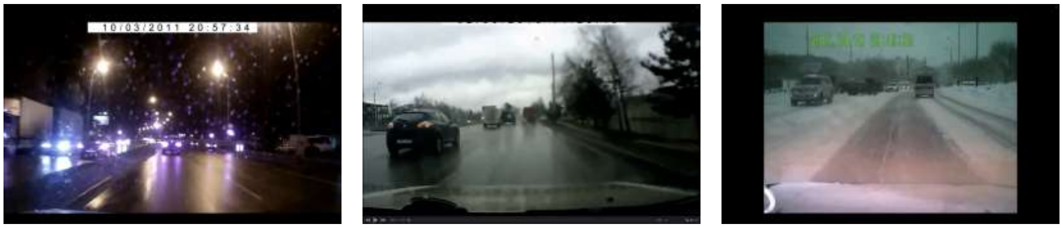
Figure 6: Illustration of near collision but no crash scenarios

A.5 Generation Eamples

We include generated example scenarios of three base scenario categories from all 5 datasets in 5.

A.6 Limitations

Lack of static baseline comparison. Prior adversarial driving baselines—Learning-to-Collide [17], Adversarial Simulation [43], ScenarioRunner [9], and adversarial trajectory optimization [51]—operate in *static* conditions, where spawn points, agent behaviors, and traffic light states are fixed. Such settings are stable but lack the uncertainty and variability of real traffic. Our framework generates *dynamic* scenarios where spawn timing, interactions, and geometry vary at runtime. This improves realism but complicates direct comparison, since SafeBench is designed for static inputs.



(a) Dark condition (b) Rainy condition (c) Snowy condition

Figure 7: Example frames of safety critical scenario in CCD-C caused by various weather condition.

For fairness and reproducibility, we evaluate all systems in dynamic settings and avoid manually editing generated code, but this makes static baseline alignment challenging.

Generation and repair failures. The ChatScene retrieval database achieves ∼90% Scenic code generation success. Most failures arise from syntax issues (indentation, parentheses, undefined symbols), and our automated error-recovery pipeline resolves ∼50% of them. Remaining failures stem from inconsistencies in the retrieval database. Even valid Scenic scripts may fail at simulation time due to infeasible geometry (e.g., overlapping agents, invalid lane positions) or weak adversarial interactions. To preserve full automation, we do not manually modify outputs; invalid or low-quality scenarios are logged but excluded from evaluation.

Benchmark limitations. SafeBench does not distinguish difficulty levels, measure crash severity, or capture near-miss events. Minor scrapes and multi-vehicle collisions are scored similarly, and near-misses—critical for understanding perception, braking margin, and emergency maneuvers—are ignored 6. This may favor trivial collision-forcing scenarios over realistic late-braking or evasive-maneuver settings. A richer benchmark is needed to evaluate risk-aware policies that respond proactively rather than only avoiding crashes.

Limited environmental factors. SafeBench currently does not incorporate weather, visibility, or illumination into evaluation. While various weather conditions are presented in the datasets such as rain, fog, or glare, these are excluded by benchmark constraints 7. As adverse weather and lighting are key contributors to real-world risk, future evaluation pipelines should integrate environmental exposure and performance degradation metrics.

Pretrained ego agent behavior. The pretrained RL agents occasionally behave unnaturally (e.g., slow highway driving, abrupt accelerations, partial-sidewalk overtakes). Such behaviors reduce interaction frequency and adversarial pressure, especially in timing-critical intersection cases. Training ego agents directly on dynamic scenarios may yield more human-like behavior and more faithful robustness evaluation under diverse safety-critical conditions.

A.7 Generated Scenic Code

```
'''The ego vehicle is driving straight on a snowy road; the adversarial vehicle, a
    black sedan, is stopped at a traffic light. The traffic light changes from red
    to green, allowing the ego vehicle to proceed. The adversarial vehicle remains
    stationary, waiting for the light to change again.'''
Town = 'Town05'
param map = localPath(f'../maps/{Town}.xodr')
param carla_map = Town
model scenic.simulators.carla.model
EGO_MODEL = "vehicle.lincoln.mkz_2017"

behavior AdvBehavior():
    while (distance to self) > 60:
        wait
    do FollowLaneBehavior(globalParameters.OPT_ADV_SPEED)

param OPT_ADV_SPEED = Range(0, 20)
```

```
## MONITORS
monitor TrafficLights:
    freezeTrafficLights()
    while True:
        if withinDistanceToTrafficLight(ego, 100):
            setClosestTrafficLightStatus(ego, "green")
        if withinDistanceToTrafficLight(AdvAgent, 100):
            setClosestTrafficLightStatus(AdvAgent, "red")
        wait

intersection = Uniform(*filter(lambda i: i.is4Way and i.isSignalized,
    network.intersections))
egoInitLane = Uniform(*intersection.incomingLanes)
egoManeuver = Uniform(*filter(lambda m: m.type is ManeuverType.STRAIGHT,
    egoInitLane.maneuvers))
egoTrajectory = [egoInitLane, egoManeuver.connectingLane, egoManeuver.endLane]
egoSpawnPt = OrientedPoint in egoManeuver.startLane.centerline

ego = Car at egoSpawnPt,
    with regionContainedIn None,
    with blueprint EGO_MODEL

require 10 <= (distance to intersection) <= 40

param OPT_GEO_BLOCKER_Y_DISTANCE = Range(0, 40)
param OPT_GEO_X_DISTANCE = Range(-8, 0)
param OPT_GEO_Y_DISTANCE = Range(10, 30)

laneSec = network.laneSectionAt(ego)
IntSpawnPt = OrientedPoint following roadDirection from egoSpawnPt for
    globalParameters.OPT_GEO_BLOCKER_Y_DISTANCE
Blocker = Car at IntSpawnPt,
    with heading IntSpawnPt.heading,
    with regionContainedIn None

SHIFT = globalParameters.OPT_GEO_X_DISTANCE @ globalParameters.OPT_GEO_Y_DISTANCE
AdvAgent = Car at Blocker offset along IntSpawnPt.heading by SHIFT,
    with heading IntSpawnPt.heading + 180 deg,
    with regionContainedIn laneSec._laneToLeft,
    with behavior AdvBehavior()
```

A.8 Prompt

A.8.1 Prompt for CCTV View Normal Traffic Scenario

```
You are an expert compiler-fixer for Scenic programs targeting CARLA.

Goal: Produce a single corrected Scenic source file which compiles without errors.
    Prefer minimal, semantics-preserving edits that keep the original intent. If a
    construct is unsupported or undefined, replace it with the closest valid
    Scenic/Carla alternative rather than deleting essential logic.

Inputs (verbatim):
Error:
{err_text}

Original Code:
{code}

STRICT OUTPUT FORMAT (must follow exactly):
- Return ONLY the corrected Scenic program text.
- No explanations, no reasoning, no extra lines, no markdown fences, no code
    fences, no comments, no quotes.
- One complete Scenic file.

Hard requirements (compilation-first):
1) The program must parse and type-check under Scenic for CARLA.
2) Include a valid model line, e.g.:
   model scenic.simulators.carla.model
3) Define exactly one ego object; do not reference ego before it is defined.
4) Ensure every identifier is defined before use (params, variables, objects,
    behaviors, monitors).
5) Use valid Scenic syntax for objects and fields, e.g.:
   obj = Car at <region/oriented point>, with blueprint "vehicle...", with behavior
    SomeBehavior(), ...
6) Use valid units/constructs:
   - Angles with "deg" where appropriate.
   - Ranges and distributions: Range(a, b), Uniform(...), Discrete(...).
   - Booleans are True/False; use "and/or/not"; avoid Python f-strings or print.
7) Keep the program in a single file; no external imports beyond the model line; no
    raw Python that Scenic won't accept.

Semantic preservation & scope of edits (apply only as needed to compile):
8) Preserve the scenario's intent (agents, roles, rough geometry, behaviors,
    monitors). Prefer targeted fixes over wholesale rewrites.
9) If an attribute or API is undefined (e.g., globalParameters.X,
    network.laneSectionAt, laneSec._laneToLeft, custom helpers), do ONE of:
   - Replace with an equivalent defined symbol already present (e.g., X if "param
    X" exists).
   - Substitute with a valid Scenic/Carla construct (e.g., use known lanes/regions,
    or remove a fragile attribute while keeping object placement).
   - Introduce a minimal, valid definition (param/constant) if it is clearly
    intended and harmless.
10) If both map forms are present, keep a consistent one:
    - Either param carla_map = "TownXX"
    - Or param map = localPath("...xodr")
    but not conflicting duplicates.
11) Behaviors/monitors must be valid Scenic blocks:
    behavior Name():
        while <cond>:
            wait
        do SomeBehavior(arg)

    monitor Name:
        while True:
            ...
            wait
12) Replace unsupported calls with safe equivalents or remove them if non-essential:
    - If setClosestTrafficLightStatus/withinDistanceToTrafficLight or
    freezeTrafficLights are unavailable, either provide a minimal valid monitor or
    omit the call while keeping the rest compiling.
13) Keep randomness where intended; fix invalid distributions, names, or parameter
    uses (e.g., change globalParameters.OPT_ADV_SPEED to OPT_ADV_SPEED if param
    exists).

Common fixes (apply as needed):
14) Add missing commas after object fields; fix colons/indentation in blocks.
15) Ensure lists/tuples/brackets are balanced; remove trailing stray characters.
16) Ensure "distance to <object>" uses valid Scenic grammar; avoid "self" outside
    behaviors.
17) Ensure OrientedPoint/offset syntax is valid; if a helper is unknown, replace
    with a simpler, valid placement (e.g., at ego.position offset by (dx, dy) or
    at roadRegion).
18) Ensure "require" statements are valid boolean constraints and reference defined
    names.

Validation checklist before returning:
- Does it include the model line?
- Is exactly one ego defined?
- Are all names defined before use?
- Do behaviors/monitors parse?
- Are map parameters consistent?
- No undefined helpers remain.
- No comments or extra text are output.

Return ONLY the corrected Scenic program.
"""
```

A.8.2 Prompt for Error Checker

```
You are viewing the scene from a fixed road surveillance CCTV camera.

The current driving situation is calm and uneventful.

Your task is to transform the uneventful situation into a single high-risk driving
    interaction by following these instructions carefully:
1) From the list of eight event categories below, select EXACTLY ONE category that
    best fits the context.
2) Rewrite the uneventful scene into a hazardous interaction belonging to that
    category.
3) Be realistic and creative, ensuring the rewritten event could plausibly happen
    in real-world urban or highway traffic.

STRICT OUTPUT RULES:
- Output only the rewritten scene text.
- Do NOT include headings, labels, numbers, or categories.
- Do NOT reference the safe situation. Only write the rewritten version.
- Write one short paragraph, 1-3 sentences, no more than 70 words.
- Always begin with "The ego vehicle...".
- Forbidden words: Dangerous, Scenario, Description, Safe, Safety, Task, Type,
    Category, Label, Prompt.

Event Categories with Examples:
Straight Obstacle: The ego vehicle goes straight and is sideswiped by a car merging
    suddenly from a parking spot, forcing the driver to swerve into the adjacent
    lane.
Turning Obstacle: The ego vehicle is turning left in a 4-way intersection when a
    pedestrian suddenly emerges from the right of a stopped vehicle and walks into
    the lane.
```

```
Lane Changing: The ego vehicle is performing a lane change when the car in the
    target lane suddenly brakes, forcing an abrupt maneuver to avoid collision.
Vehicle Passing: The ego vehicle is passing a bus when it swerves toward the center
    without signaling, squeezing the ego into oncoming traffic.
Red-light Running: The ego vehicle proceeds straight through an intersection as a
    truck from the left runs the red light and stalls in the middle.
Unprotected Left-turn: The ego vehicle begins an unprotected left turn while
    yielding when a cyclist from the crosswalk suddenly accelerates into the
    turning path.
Right-turn: The ego vehicle is turning right when a pedestrian suddenly steps from
    behind a stopped car into the crosswalk.
Crossing Negotiation: The ego vehicle enters an intersection as a car from the
    opposite direction makes a sudden left turn and halts in front, nearly causing
    a collision.
```

A.9 Scenario Descriptions

A.9.1 Scenario Descriptions for CCD-C

Table 10: Examples of CCD-C scenarios categorized into the 8 SafeBench base scenario types.

Base Scenario	**Example Descriptions**
Straight Obstacle	1. The ego vehicle is driving straight on a city street when a black SUV suddenly accelerates and drives into the intersection from the left front, blocking the ego vehicle's path. The ego vehicle reacts quickly and swerves to the right to avoid a collision. 2. The ego vehicle is driving on a straight road; the adversarial truck is driving on the same road and suddenly stops in front of the ego vehicle, blocking its path. 3. The ego vehicle is driving straight on a road; the adversarial vehicle approaches from the left front and suddenly swerves to the right, causing the ego vehicle to react quickly to avoid a collision. 4. The ego vehicle is driving straight on a rural road; the adversarial vehicle, a white sedan, is following closely behind. The white sedan suddenly stops abruptly in front of the ego vehicle, blocking its path. The ego vehicle continues to drive forward, unable to pass due to the stopped white sedan. 5. The ego vehicle is driving on a straight road; the adversarial car on the right front suddenly accelerates and enters the road, blocking the ego vehicle's path.

Continued on next page

Base Scenario	Example Descriptions
Turning Obstacle	1. The ego vehicle is driving on a straight road; the adversarial vehicle is driving on the same road and is following closely behind the ego vehicle. The adversarial vehicle suddenly accelerates and takes a sharp turn to the right, causing the ego vehicle to react quickly to avoid a collision. 2. The ego vehicle is driving straight on a city street; the adversarial vehicle, a black SUV, is driving in the same direction and is positioned slightly ahead of the ego vehicle. The adversarial vehicle suddenly accelerates and takes a sharp right turn, causing the ego vehicle to react quickly to avoid a collision. 3. The ego vehicle is driving on a straight road; the adversarial car is driving in the same lane and suddenly accelerates, causing the ego vehicle to react quickly to avoid a collision. 4. The ego vehicle is driving straight on a snowy road at night; the adversarial vehicle, a white sedan, is following closely behind. As they approach an intersection, the adversarial vehicle suddenly swerves to the right, causing the ego vehicle to react quickly to avoid a collision. 5. The ego vehicle is driving straight on a snowy street; the adversarial vehicle approaches from the left front and suddenly swerves to the right, causing the ego vehicle to react quickly to avoid a collision.
Lane Changing	1. The ego vehicle is driving on a straight road; the adversarial vehicle is following closely behind, blocking the ego vehicle's path. The ego vehicle attempts to overtake, but the adversarial vehicle suddenly swerves to the right, causing the ego vehicle to react quickly to avoid a collision. 2. The ego vehicle is driving straight on a snowy road; the adversarial vehicle is following closely behind, blocking the ego vehicle's path. The ego vehicle attempts to maintain its lane but is unable to do so due to the adversarial vehicle's obstruction. 3. The ego vehicle is driving on a straight road; the adversarial vehicle is driving on the same road in front of the ego vehicle. The adversarial vehicle suddenly stops in front of the ego vehicle, blocking its path. 4. The ego vehicle is driving on a straight road; the adversarial vehicle is driving in the same lane and suddenly stops in front of the ego vehicle, blocking its path. 5. The ego vehicle is driving straight on a city street; the adversarial vehicle is following closely behind, blocking the ego vehicle's path. The ego vehicle attempts to overtake, but the adversarial vehicle remains stationary, creating a traffic jam.
Vehicle Passing	1. The ego vehicle is driving on a straight road; the adversarial vehicle is driving on the same road and is overtaking the ego vehicle. 2. The ego vehicle is turning right when a motorcyclist from behind overtakes and cuts into the turning lane. 3. The ego vehicle is driving straight on a road when a white sedan suddenly appears from the right side of the road and drives past the ego vehicle. 4. The ego vehicle is driving straight on a snowy urban road when a black sedan suddenly accelerates and cuts in front of the ego vehicle, causing the ego vehicle to slightly drift to the right. The ego vehicle then continues straight, following the black sedan. 5. The ego vehicle is driving straight on a snowy street when a red car suddenly appears from the left and drives past the ego vehicle. The ego vehicle continues straight, following the red car.

Continued on next page

Base Scenario	Example Descriptions
Red-light Running	1. The ego vehicle is driving straight on a snowy road; the adversarial vehicle, a black sedan, is stopped at a traffic light. The traffic light changes from red to green, allowing the ego vehicle to proceed. The adversarial vehicle remains stationary, waiting for the light to change again. 2. The ego vehicle is driving straight on a snowy urban street at night. The street is illuminated by streetlights and the headlights of various vehicles. Snow is falling heavily, creating a blizzard-like atmosphere. The road is covered in a thick layer of snow, and there are several cars and trucks driving slowly, their headlights casting long shadows on the snow. Some vehicles are stationary, possibly due to the poor visibility caused by the snowstorm. 3. The ego vehicle is driving straight on a city street at night. The street is illuminated by streetlights and the headlights of other vehicles. There are several cars on the road, including a red car in front of the ego vehicle. The ego vehicle is following the red car and maintaining a safe distance. 4. The ego vehicle is driving on a straight road; the adversarial car is driving on the same road and is following closely behind the ego vehicle. The ego vehicle maintains a steady speed and the adversarial car also maintains a steady speed. The two vehicles remain parallel to each other throughout the scene. 5. The ego vehicle is driving straight on a city street; the adversarial vehicle, a silver minivan, is following closely behind. The minivan suddenly stops abruptly in front of the ego vehicle, blocking its path. The ego vehicle continues to drive forward, trying to navigate around the stopped minivan.
Unprotected Left-turn	1. The ego vehicle is driving straight on a snowy road at night. A white truck approaches from the left front, its headlights illuminating the snow-covered road. The truck continues straight ahead, passing the ego vehicle. The ego vehicle maintains its course, following the truck's path. 2. The ego vehicle is driving on a straight road; the adversarial vehicle, a white sedan, is following closely behind. The ego vehicle maintains a steady speed, while the adversarial vehicle occasionally adjusts its speed slightly. The road is relatively clear, with no other vehicles in sight. 3. The ego vehicle is driving on a straight road; the adversarial vehicle, a white van, is driving in the same lane and slightly ahead of the ego vehicle. The van suddenly stops abruptly, blocking the path of the ego vehicle. 4. The ego vehicle is driving straight on a city street when a red car suddenly appears from the left and drives past the ego vehicle. The red car continues straight ahead and then turns right, passing the ego vehicle on the right side. 5. The ego vehicle is driving straight on a city street; the adversarial pedestrian appears from behind a bus stop on the right front, then suddenly sprints out onto the road in front of the ego vehicle and stops.

Continued on next page

Base Scenario	Example Descriptions
Right-turn	1. The ego vehicle is driving on a straight road; the adversarial pedestrian appears from behind a parked car on the right front and suddenly stop and walk diagonally. 2. The ego vehicle is driving on a straight road; the adversarial vehicle appears from behind a parked car on the right front and suddenly stops in the middle of the intersection, blocking the ego vehicle's path. 3. The ego vehicle is driving on a straight road; the adversarial vehicle appears from behind a bus stop on the right front, then suddenly sprints out onto the road in front of the ego vehicle and stops. 4. The ego vehicle is driving on a straight road; the adversarial pedestrian stands behind a bus stop on the right front, then suddenly sprints out onto the road in front of the ego vehicle and stops. 5. The ego vehicle is driving on a straight road; the adversarial pedestrian appears from behind a parked car on the right front and suddenly stops and walks diagonally.
Crossing Negotiation	1. The ego vehicle is driving straight on a city street when a pedestrian suddenly crosses from the right front and suddenly stops as the ego vehicle approaches. 2. The ego vehicle is driving on a straight road; the adversarial vehicle appears from behind a snowbank on the right front, then suddenly sprints out onto the road in front of the ego vehicle and stops. 3. The ego vehicle is driving straight on a road; the adversarial pedestrian is standing on the sidewalk on the right front, then suddenly sprints out onto the road in front of the ego vehicle and stops. 4. The ego vehicle is driving straight on a snowy road; the adversarial pedestrian appears from behind a parked car on the right front and suddenly stop. 5. The ego vehicle is driving on a straight road; the adversarial pedestrian appears from behind a parked car on the right front and suddenly stop and walk diagonally.

A.9.2 Scenario Descriptions for CCN-N

Table 11: Examples of CCD-N scenarios categorized into the 8 SafeBench base scenario types.

Base Scenario	Description of Scenarios
Straight Obstacle	1. The ego vehicle is driving straight on a rural road when a child suddenly runs out from the left front, forcing the ego to brake sharply to avoid a collision. 2. The ego vehicle is driving straight on a city street when a cyclist swerves out of a driveway directly into the ego's path. 3. The ego vehicle is driving straight when a delivery van parked on the right suddenly opens its door into the lane, blocking the ego's path. 4. The ego vehicle is driving straight on a highway when a box falls from a truck ahead, forcing the ego to react abruptly. 5. The ego vehicle is driving straight when a dog suddenly runs across the road from the left, causing the ego to swerve.

Continued on next page

Base Scenario	Description of Scenarios
Turning Obstacle	1. The ego vehicle is turning left at a 4-way intersection when a vehicle oncoming from the opposite lane accelerates aggressively, forcing the ego to stop mid-turn. 2. The ego vehicle is turning left at an intersection when a pedestrian suddenly runs across from the left front, blocking the ego vehicle's path. 3. The ego vehicle is turning right at an intersection when a cyclist riding along the crosswalk veers into the lane unexpectedly. 4. The ego vehicle is turning left at a busy intersection when a motorcyclist behind overtakes and cuts across the ego's front wheel. 5. The ego vehicle is turning right at an intersection when a bus in the adjacent lane swings wide and enters the ego's turning path.
Lane Changing	1. The ego vehicle is changing lanes on a highway when an adversarial truck behind accelerates and closes the gap, leaving the ego stuck between lanes. 2. The ego vehicle is attempting to change to the left lane, but the vehicle in that lane suddenly brakes, creating a near collision. 3. The ego vehicle is performing a right-lane change when a motorcycle weaves between lanes and cuts directly into its path. 4. The ego vehicle is changing lanes to overtake a slow vehicle when another car simultaneously moves into the same lane from the opposite side. 5. The ego vehicle initiates a lane change, but a hidden car emerges quickly from the blind spot and forces the ego to abort.
Vehicle Passing	1. The ego vehicle attempts to pass a slow tractor, but as it pulls alongside, the tractor unexpectedly swerves right into the passing lane. 2. The ego vehicle begins passing a bus when the bus suddenly accelerates and drifts toward the center, narrowing the ego's maneuvering space. 3. The ego vehicle is overtaking a parked car when the parked car suddenly starts moving into the lane without signaling. 4. The ego vehicle is attempting to pass a delivery truck when a pedestrian emerges from behind it and walks across the lane. 5. The ego vehicle is passing a stopped vehicle when another vehicle in the oncoming lane accelerates rapidly toward the ego, forcing it to brake.
Red-light Running	1. The ego vehicle moves straight through a green light, but a vehicle from the cross street runs the red light at high speed and cuts across the ego's path. 2. The ego is proceeding through an intersection when two motorcycles simultaneously run a red light from opposite directions. 3. The ego vehicle drives straight at a green light when a truck from the right runs its red light and stalls mid-intersection. 4. The ego continues through a green signal when a car from the left accelerates through a red light, forcing the ego to brake suddenly. 5. The ego is driving straight at an intersection when a bus enters illegally on red and blocks the entire junction.
Unprotected Left-turn	1. The ego vehicle attempts an unprotected left turn when an oncoming car suddenly accelerates instead of yielding, forcing the ego to brake mid-turn. 2. The ego starts an unprotected left turn when a motorcycle in the opposite lane makes an unexpected right turn across the ego's path. 3. The ego initiates a left turn, but an oncoming car tailgates another vehicle and squeezes through, cutting into the intersection. 4. The ego is making an unprotected left when a cyclist enters the intersection diagonally, blocking the turning path. 5. The ego attempts a left turn when a vehicle from the opposite direction runs through and swerves erratically into the intersection.

Continued on next page

Base Scenario	Description of Scenarios
Right-turn	1. The ego vehicle is turning right when a pedestrian suddenly appears from behind a utility pole and crosses the crosswalk. 2. The ego is turning right on a green signal when a bicycle rider on the sidewalk suddenly swerves into the lane. 3. The ego vehicle is turning right when a motorcyclist from behind overtakes and cuts into the turning lane. 4. The ego vehicle is turning right at an intersection when a delivery van blocks the turn by stopping mid-crosswalk. 5. The ego is making a right turn when an oncoming car from the left runs a late yellow and nearly collides.
Crossing Negotiation	1. The ego is approaching a 4-way intersection when a car from the left enters aggressively, forcing the ego to yield despite having priority. 2. The ego vehicle approaches the intersection when two vehicles, one from the right and one from the opposite side, both enter at once, blocking the ego. 3. The ego enters a crossing when a truck from the opposite direction begins a wide left turn and stops in the ego's path. 4. The ego proceeds straight when a car from the right side misjudges the gap and cuts across the intersection, nearly colliding. 5. The ego approaches the intersection while a bus from the left enters slowly and blocks multiple lanes, forcing the ego to stop.

A.9.3 Scenario Descriptions for TADS

Table 12: Examples of TADS scenarios categorized into the 8 SafeBench base scenario types.

Base Scenario	Description of Scenarios
Straight Obstacle	1. The ego vehicle is driving on a straight road; the adversarial pedestrian suddenly runs out from behind a parked van on the right front and suddenly stops in the lane. 2. The ego vehicle is driving on a straight road; the adversarial pedestrian steps out from between two parked cars on the left front and suddenly stops. 3. The ego vehicle is driving on a straight road; the adversarial cyclist emerges from a side alley on the right front and stops across the ego's path. 4. The ego vehicle is driving on a straight road; the adversarial car pulls out of a driveway on the left front without yielding and stalls perpendicular to the lane. 5. The ego vehicle is driving on a straight road; the adversarial pedestrian appears from behind a bus stop on the right front and suddenly stops in front of the ego.
Turning Obstacle	1. The ego vehicle is turning left at a 4-way intersection; the adversarial pedestrian from the right side steps into the crosswalk and stops directly in the ego's path. 2. The ego vehicle is turning left at an intersection; the adversarial vehicle ahead brakes abruptly mid-turn and blocks the lane. 3. The ego vehicle is turning left at a 4-way intersection; the adversarial cyclist on the right front swerves into the road and stops at the apex. 4. The ego vehicle is turning left at an intersection; the adversarial pedestrian emerges from behind a stopped vehicle on the left front and stops in the crosswalk. 5. The ego vehicle is turning right at an intersection; the adversarial pedestrian from the right front enters the crosswalk and suddenly stops in the middle.

Continued on next page

Base Scenario	**Description of Scenarios**
Lane Changing	1. The ego vehicle is changing lanes to the left; the adversarial car in the target lane accelerates to close the gap and then brakes, blocking the merge.
	2. The ego vehicle is changing lanes to the right; the adversarial car in the target lane matches the ego's speed and holds position next to the ego, preventing the lane change.
	3. The ego vehicle is changing lanes to avoid a slow vehicle; the adversarial car in the target lane suddenly slows down, forcing the ego to abort the merge.
	4. The ego vehicle is changing lanes on a multi-lane road; the adversarial motorcycle lane-splits across the ego's path and stops briefly ahead.
	5. The ego vehicle is changing lanes to pass; the adversarial vehicle from the opposite side also changes into the same lane and then brakes abruptly.
Vehicle Passing	1. The ego vehicle is passing a stopped bus; the adversarial bus pulls out without signaling and then stops for a late passenger, narrowing the space.
	2. The ego vehicle is passing a parked car that blocks the lane; an oncoming vehicle appears and the adversarial car ahead hesitates, forcing the ego to slow sharply.
	3. The ego vehicle is passing road debris; the adversarial car ahead swerves around it without signaling and then stops in the ego's lane.
	4. The ego vehicle is passing a delivery van; a person steps out from the far side of the van into the lane and suddenly stops.
	5. The ego vehicle is passing a stalled vehicle; the adversarial taxi begins a U-turn into the same gap and stops sideways, blocking the lane.
Red-light Running	1. The ego vehicle goes straight through a 4-way intersection on green; the adversarial vehicle from the right runs a red light and then stops mid-intersection.
	2. The ego vehicle goes straight through an intersection; two adversarial motorcycles from the opposite left-turn pocket launch on red and cut in front of the ego.
	3. The ego vehicle goes straight at a 4-way intersection; the adversarial truck from the left runs the red and stalls diagonally across the crosswalk.
	4. The ego vehicle proceeds straight at an intersection; the adversarial car from the left runs the red and turns right across the ego's path, then stops.
	5. The ego vehicle goes straight at an intersection; the adversarial vehicle from the left front runs the red and makes an abrupt left turn, forcing a hard brake.
Unprotected Left-turn	1. The ego vehicle attempts an unprotected left turn; the oncoming lead car slows while the second oncoming car accelerates, closing the accepted gap.
	2. The ego vehicle begins an unprotected left turn; the adversarial car behind overtakes into the intersection and cuts across the ego's front.
	3. The ego vehicle starts an unprotected left turn; the adversarial pedestrian enters from the right crosswalk area and stops in the turning path.
	4. The ego vehicle commits to an unprotected left; the adversarial vehicle approaches at normal speed and then makes a sudden right turn across the ego.
	5. The ego vehicle performs an unprotected left; the oncoming car veers erratically and then stops in the intersection, obstructing the turn.

Continued on next page

Base Scenario	Description of Scenarios
Right-turn	1. The ego vehicle is turning right at an intersection; the adversarial pedestrian steps from the sidewalk into the crosswalk and stops directly in front of the ego. 2. The ego vehicle is turning right at a 4-way intersection; the adversarial vehicle from the left enters the intersection at speed and then brakes sharply ahead of the ego. 3. The ego vehicle is turning right at an intersection; the adversarial car ahead on the right reverses abruptly and stops in the lane. 4. The ego vehicle is turning right at an intersection; the adversarial cyclist riding along the sidewalk abruptly enters the road and stops at the exit of the turn. 5. The ego vehicle is turning left at an intersection; the adversarial pedestrian from the near right corner steps into the crosswalk and stops on the lane line.
Crossing Negotiation	1. The ego vehicle approaches a 4-way intersection requiring crossing negotiation; the adversarial car from the right accelerates to enter first and then stops across the ego's path. 2. The ego vehicle enters a 4-way intersection requiring crossing negotiation; the adversarial vehicle from the opposite lane turns left and stops in front of the ego, creating a near collision. 3. The ego vehicle goes straight through an intersection requiring crossing negotiation; the adversarial vehicle from the left side misjudges the gap and stops in the ego's path. 4. The ego vehicle enters a crossing where two adversarial cars, from the right and left, both attempt to cross and stop midway, blocking the ego. 5. The ego vehicle approaches a crossing where a bus from the opposite side begins a wide left turn and stops in the middle of the intersection.

A.9.4 Scenario Descriptions for Tumtraf-I

Table 13: Examples of Tumtraf-I scenarios categorized into the 8 SafeBench base scenario types.

Base Scenario	Description of Scenarios
Straight Obstacle	1. The ego vehicle is driving on a straight road; the adversarial pedestrian emerges from behind a parked bus on the right front and suddenly stops in the lane. 2. The ego vehicle is driving on a straight road; the adversarial pedestrian appears from a shop entrance on the left front and stops directly ahead. 3. The ego vehicle is driving on a straight road; the adversarial cyclist comes out from between parked cars on the right front and stops across the lane. 4. The ego vehicle is driving on a straight road; the adversarial car reverses from a parking space on the left front and stops, blocking the lane. 5. The ego vehicle is driving on a straight road; the adversarial scooter rider exits a side alley on the right front and halts in front of the ego.

Continued on next page

Base Scenario	Description of Scenarios
Turning Obstacle	1. The ego vehicle is turning left at a 4-way intersection; the adversarial pedestrian from the near right corner enters the crosswalk and stops mid-zebra. 2. The ego vehicle is turning left at an intersection; the adversarial vehicle ahead brakes abruptly mid-turn and blocks the lane. 3. The ego vehicle is turning right at an intersection; the adversarial cyclist riding along the sidewalk cuts into the road and stops at the exit of the turn. 4. The ego vehicle is turning left at an intersection; the adversarial pedestrian emerges from behind a stopped vehicle on the right front and stops in the crosswalk. 5. The ego vehicle is turning right at an intersection; the adversarial vehicle from the left enters the intersection and brakes sharply ahead of the ego.
Lane Changing	1. The ego vehicle is changing lanes to the left; the adversarial car in the target lane accelerates to close the gap and then brakes, blocking the merge. 2. The ego vehicle is changing lanes to the right; the adversarial car in the target lane matches the ego's speed and holds position beside the ego, preventing the merge. 3. The ego vehicle is changing lanes; the adversarial motorcycle lane-splits across the ego's path and stops briefly ahead. 4. The ego vehicle is changing lanes to the left; the adversarial vehicle from the opposite side also moves into the same lane and then brakes abruptly. 5. The ego vehicle is changing lanes; the adversarial car in the target lane drifts over the lane line without signaling and stops straddling lanes.
Vehicle Passing	1. The ego vehicle is passing a stopped bus; the adversarial bus pulls out without signaling and then re-brakes for a late passenger, compressing the gap. 2. The ego vehicle is moving around a stalled car; the adversarial taxi begins a U-turn into the same gap and stops sideways, blocking the lane. 3. The ego vehicle is passing a delivery van; the adversarial pedestrian steps from the far side of the van into the lane and suddenly stops. 4. The ego vehicle is passing road debris; the adversarial car ahead swerves around it without signaling and then stops in the ego's lane. 5. The ego vehicle is overtaking a street sweeper; the adversarial sweeper veers toward the centerline, narrowing the passage and forcing the ego to slow.
Red-light Running	1. The ego vehicle goes straight at a 4-way intersection on green; the adversarial vehicle from the right runs a red light and stops mid-intersection. 2. The ego vehicle goes straight through an intersection; the adversarial vehicle from the opposite approach runs the red to make a fast left and then stalls in the ego's lane. 3. The ego vehicle proceeds straight through an intersection; the adversarial delivery truck from the left runs the red and stops diagonally across the crosswalk. 4. The ego vehicle goes straight through an intersection; the adversarial van from the left runs the red and turns right across the ego's path, then stops. 5. The ego vehicle goes straight at an intersection; the adversarial car from the left front runs the red and makes an abrupt left turn, forcing a hard brake.

Continued on next page

Base Scenario	**Description of Scenarios**
Unprotected Left-turn	1. The ego vehicle attempts an unprotected left turn; the oncoming lead car slows while the second oncoming car accelerates, closing the accepted gap. 2. The ego vehicle begins an unprotected left turn; the adversarial car behind overtakes into the intersection and cuts across the ego's front. 3. The ego vehicle starts an unprotected left turn; the adversarial pedestrian enters from the right crosswalk area and stops in the turning path. 4. The ego vehicle commits to an unprotected left; the adversarial vehicle approaches at normal speed and then makes a sudden right turn across the ego. 5. The ego vehicle performs an unprotected left; the oncoming car veers unpredictably and then stops in the intersection, obstructing the turn.
Right-turn	1. The ego vehicle is turning right at an intersection; the adversarial pedestrian steps from the sidewalk into the crosswalk and stops directly in front of the ego. 2. The ego vehicle is turning right at a 4-way intersection; the adversarial vehicle from the left enters the intersection at speed and then brakes sharply ahead of the ego. 3. The ego vehicle is turning right at an intersection; the adversarial car ahead on the right reverses abruptly and stops in the lane. 4. The ego vehicle is turning right at an intersection; the adversarial cyclist riding along the sidewalk abruptly enters the road and stops at the exit of the turn. 5. The ego vehicle is turning left at an intersection; the adversarial pedestrian from the near right corner steps into the crosswalk and stops on the lane line.
Crossing Negotiation	1. The ego vehicle approaches a 4-way intersection requiring crossing negotiation; the adversarial car from the left accelerates to enter first and then stops across the ego's path. 2. The ego vehicle enters a 4-way intersection requiring crossing negotiation; the adversarial vehicle from the opposite lane turns left and stops in front of the ego, creating a near collision. 3. The ego vehicle approaches a 4-way intersection requiring crossing negotiation; the adversarial car from the right accelerates into the intersection first and then stops. 4. The ego vehicle enters an intersection requiring crossing negotiation; the adversarial vehicle from the opposite approach turns left and stops on the centerline, blocking the path. 5. The ego vehicle approaches a 4-way intersection requiring crossing negotiation; a bus from the left enters aggressively and then halts mid-intersection, obstructing traffic.

A.9.5 Scenario Descriptions for Tumtraf-V

Table 14: Examples of Tumtraf-V scenarios categorized into the 8 SafeBench base scenario types.

Base Scenario	Description of Scenarios
Straight Obstacle	1. The ego vehicle is driving on a straight road; the adversarial pedestrian steps out from behind a bus shelter on the right front and suddenly stops. 2. The ego vehicle is driving on a straight road; the adversarial cyclist rides out of a driveway on the left front and stops across the lane. 3. The ego vehicle is driving on a straight road; the adversarial pedestrian opens a car door from the right curb and steps into the lane, then stops. 4. The ego vehicle is driving on a straight road; the adversarial scooter rider exits a side alley on the right front and halts in front of the ego. 5. The ego vehicle is driving on a straight road; the adversarial pedestrian steps from a median refuge and stops on the centerline.
Turning Obstacle	1. The ego vehicle is turning left at a 4-way intersection; the adversarial pedestrian from the near right corner enters the crosswalk and stops in the middle. 2. The ego vehicle is turning left at an intersection; the adversarial vehicle ahead brakes abruptly mid-turn and blocks the lane. 3. The ego vehicle is turning right at an intersection; the adversarial cyclist riding along the sidewalk cuts into the road and stops at the turn exit. 4. The ego vehicle is turning left at an intersection; the adversarial pedestrian emerges from behind a stopped car on the left front and stops in the crosswalk. 5. The ego vehicle is turning left at a 4-way intersection; the adversarial motorcyclist from the opposite approach cuts across and then brakes to a stop.
Lane Changing	1. The ego vehicle is changing lanes to the left; the adversarial car in the target lane accelerates to close the gap and then brakes. 2. The ego vehicle is changing lanes to the right; the adversarial car in the target lane matches speed and holds beside the ego, preventing the merge. 3. The ego vehicle is changing lanes on a multi-lane road; the adversarial motorcycle lane-splits across the ego's path and stops briefly ahead. 4. The ego vehicle is changing lanes to pass; the adversarial vehicle from the opposite side also moves into the same lane and then brakes abruptly. 5. The ego vehicle is changing lanes; the adversarial car in the target lane drifts over the lane line without signaling and stops straddling lanes.
Vehicle Passing	1. The ego vehicle is passing a stopped bus; the adversarial bus pulls out without signaling and then re-brakes for a late passenger, compressing the gap. 2. The ego vehicle is moving around a stalled car; the adversarial taxi begins a U-turn into the same gap and stops sideways, blocking the lane. 3. The ego vehicle is passing a delivery van; the adversarial pedestrian steps from the far side of the van into the lane and suddenly stops. 4. The ego vehicle is passing road debris; the adversarial car ahead swerves around it without signaling and then stops in the ego's lane. 5. The ego vehicle is overtaking a street sweeper; the adversarial sweeper veers toward the centerline, narrowing the passage and forcing the ego to slow.

Continued on next page

Base Scenario	Description of Scenarios
Red-light Running	1. The ego vehicle goes straight at a 4-way intersection on green; the adversarial vehicle from the right runs a red light and stops mid-intersection.
	2. The ego vehicle goes straight through an intersection; the adversarial vehicle from the opposite approach runs the red to make a fast left and then stalls in the ego's lane.
	3. The ego vehicle proceeds straight through an intersection; the adversarial delivery truck from the left runs the red and stops diagonally across the crosswalk.
	4. The ego vehicle goes straight through an intersection; the adversarial van from the left runs the red and turns right across the ego's path, then stops.
	5. The ego vehicle goes straight at an intersection; the adversarial car from the left front runs the red and makes an abrupt left turn, forcing a hard brake.
Unprotected Left-turn	1. The ego vehicle attempts an unprotected left turn; the oncoming lead car slows while the second oncoming car accelerates, closing the accepted gap.
	2. The ego vehicle begins an unprotected left turn; the adversarial car behind overtakes into the intersection and cuts across the ego's front.
	3. The ego vehicle starts an unprotected left turn; the adversarial pedestrian enters from the right crosswalk area and stops in the turning path.
	4. The ego vehicle commits to an unprotected left; the adversarial vehicle approaches at normal speed and then makes a sudden right turn across the ego.
	5. The ego vehicle performs an unprotected left; the oncoming car veers unpredictably and then stops in the intersection, obstructing the turn.
Right-turn	1. The ego vehicle is turning right at an intersection; the adversarial pedestrian steps from the sidewalk into the crosswalk and stops directly in front of the ego.
	2. The ego vehicle is turning right at a 4-way intersection; the adversarial vehicle from the left enters the intersection at speed and then brakes sharply ahead of the ego.
	3. The ego vehicle is turning right at a T-junction; the adversarial pedestrian from the near-left curb enters the crosswalk and stops.
	4. The ego vehicle is turning right at an intersection; the adversarial car ahead on the right reverses abruptly and stops in the lane.
	5. The ego vehicle is turning right on green; the adversarial delivery van ahead on the right reverses a short distance and stops in the lane.
Crossing Negotiation	1. The ego vehicle approaches a 4-way intersection requiring crossing negotiation; the adversarial car from the left accelerates to enter first and then stops across the ego's path.
	2. The ego vehicle enters a 4-way intersection requiring crossing negotiation; the adversarial vehicle from the opposite lane turns left and stops in front of the ego, creating a near collision.
	3. The ego vehicle approaches a 4-way intersection requiring crossing negotiation; the adversarial car from the right accelerates into the intersection first and then stops.
	4. The ego vehicle enters an intersection requiring crossing negotiation; the adversarial vehicle from the opposite approach turns left and stops on the centerline, blocking the path.
	5. The ego vehicle is approaching a signal while maintaining speed; the adversarial pedestrian steps off the curb on the right front against the signal and stops between lanes.

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: All claims made accurately reflect the paper's contributions and scope

   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: Limitation section is in Appendix

   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: The paper does not include theoretical results.

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: Prompts and scenario description are in Appendix.

   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: [NA]

Justification: The paper uses Chatscene as backbone and provides prompts and scenario descriptions which are enough for reproducing results.

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: All experiment information is in experiment and appendix sections.

   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: [No]

   Justification: This study does not report error bars.

   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: It is reported in Appendix.

   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: The study conforms with the NeurIPS Code of Ethics.

   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: [No]

    Justification: They are not part of this study and thus not discussed.

    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: All models are either api or pretrained.

    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: They are all cited or mentioned properly.

    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: [NA]

    Justification: No new assets are released.

    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: It does not involve human subjects.

    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: It does not involve human subjects.

    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: Usage is stated and explained in method section.

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.