Safety Reference Benchmarks with Avoidability Criteria for Evaluating Autonomous Driving Systems

Duong Dinh Tran, Peter Riviere, Takashi Tomita, and Toshiaki Aoki
Japan Advanced Institute of Science and Technology, Ishikawa 923-1292, Japan
{duongtd, priviere, tomita, toshiaki}@jaist.ac.jp

***Abstract*—In scenario-based testing and validation of autonomous driving systems (ADSs), existing studies often focus on generating scenarios that expose system weaknesses, such as collisions, but lack a principled way to determine whether those collisions are avoidable or unavoidable. This paper presents safety reference benchmarks that formally distinguish between collision-avoidable and collision-unavoidable conditions for the safety evaluation, testing, and verification of ADSs. The benchmarks are constructed for two critical, yet often neglected, oncoming traffic scenario classes: (1) an oncoming vehicle executing a U-turn, and (2) an oncoming vehicle temporarily swerving out of its lane to avoid an obstacle. These benchmarks serve as a universally applicable reference baseline for evaluating and comparing the safety performance of different ADS implementations and provide a direct, benchmark-driven means to identify safety-critical scenarios. To validate the utility of the reference benchmarks, we conduct empirical evaluations using a production-grade ADS (Autoware), its shielded variant, and six end-to-end learning-based agents. The results reveal safety weaknesses in Autoware as well as the six agents in collision-avoidable scenarios and demonstrate that a controller safety shield can substantially improve safety.**

***Index Terms*—Autonomous Driving, Safety Benchmark, Scenario, Autoware, Testing Oracle, Avoidable Collision, TransFuser, InterFuser**

I. Introduction

Scenario-based approach has become a widely adopted testing and validation methodology for autonomous driving systems (ADSs), as it enables assessment of ADSs under specific traffic conditions in a controlled manner. Given the vast space of all possible driving scenarios, identifying challenging or safety-critical scenarios that can effectively stress the system has become a common idea to reduce the effort [1]. Much research focused on the automated generation of such scenarios using fuzzing, search-based, or sampling-based methods [2]–[10]. These studies typically concentrate on high-density intersections or a common set of interactions, e.g., vehicles decelerating and changing lanes, to identify safety-critical situations. Oncoming traffic scenarios, e.g., when an opposing vehicle executes a U-turn, are notably absent from the critical cases generated in these prior studies, despite their frequency in real-world driving.

In studies on safety-critical scenario generation, when a collision is observed, it is often unclear whether the collision was *avoidable* for the ADS or inherently *unavoidable* because, for example, the surrounding traffic behaved unrealistically or too abruptly. The importance of distinguishing between *avoidable* and *unavoidable* collisions has been highlighted in prior work [11], [12]. In these studies, a collision is defined as *avoidable* if it can be prevented in the same scenario by modifying the configuration of the path planning module in the ADS. They implicitly assumed failures arise from improper configurations; however, collisions may also result from flawed logic in perception, planning, or control modules. Moreover, the definition is tightly coupled to the specific design of that path planning module, restricting its generalizability across different ADSs.

Addressing these critical limitations, this work presents **safety reference benchmarks** for the safety evaluation, testing, and verification of ADSs, explicitly defining the conditions under which a collision should be avoidable by a reasonably behaving ADS. The benchmarks are constructed for two critical classes of oncoming traffic scenarios: (1) an oncoming vehicle executing a U-turn, and (2) an oncoming vehicle temporarily swerving out of its lane to avoid an obstacle. To the best of our knowledge, no prior studies have explicitly examined ADS safety and reliability under these oncoming traffic scenarios. Unlike same-direction scenarios, where the motion patterns are relatively predictable, these maneuvers challenge the ADS to distinguish between a routine approach and a sudden path intrusion with high relative velocities. Our findings reveal that these scenarios expose a fundamental limitation in state-of-the-art ADSs: the failure to accurately predict traveling intent until the moment of impact.

The resulting benchmarks are designed to serve as a universally applicable reference baseline for evaluating the safety performance of different ADSs. Within the scenario set defined by the benchmarks, we can systematically identify safety-critical scenarios, namely those that are collision-avoidable yet expose collision-prone behaviors in the ADS under test/verification. This enables focused stress testing of ADSs without relying on search-based techniques or extensive random exploration, as commonly adopted in prior work. Furthermore, because outcome interpretation is standardized, the benchmarks enable a fair and objective performance comparison between different ADS implementations or system variants.

The proposed benchmarks are grounded in the industrial

safety standard proposed by the Japan Automobile Manufacturers Association (JAMA) [13]. This safety standard systematically decomposes driving conditions into multiple classes of scenarios formed through combinations of relevant factors (e.g., road geometry, relative positions and motions of surrounding vehicles) that affect the ADS's operation. This systematic decomposition offers a principled way to reason about how sufficiently the tackled scenarios cover the entire space of possible driving conditions, which is difficult to assess in prior scenario-generation approaches.

Using the benchmarks as an objective safety oracle, we evaluate the safety of Autoware [14], an open-source autonomous driving (AD) platform. While different variants of this platform have been deployed in public commercial services, e.g., autonomous taxis and buses[1], systematic research and evaluation efforts focusing on it remain limited. Our empirical experiments reveal that in some scenarios, Autoware fails to prevent collisions, even though those collisions are avoidable according to the benchmarks. The experiments are conducted using the AWSIM-Labs simulator [15] and an extended version of AWSIM-Script [16] for scenario specification.

To further validate the utility and effectiveness of the reference benchmarks, we further evaluate six end-to-end learning-based AD agents [17]–[20] in the CARLA simulator [21], enabling direct comparison between a modular production-grade ADS (Autoware) and learning-based approaches under identical scenarios. In addition, we evaluate Autoware augmented with a controller *safety shield* [22], which enforces the safety of control commands at runtime, validating the safety improvements in this shielded system.

Contributions. In summary, this work presents the following key contributions:

- **Safety reference benchmarks**: Formalized benchmarks for two critical oncoming traffic scenario classes that serve as a safety oracle for ADS testing and verification. The benchmarks explicitly characterize collision-avoidable boundaries, allowing for straightforward identification of safety-critical scenarios.
- **Benchmark validation**: Empirical evaluation across Autoware, a shielded Autoware variant, and multiple end-to-end AD agents, revealing safety weaknesses and demonstrating measurable safety improvements.

For the evaluation on Autoware, we make additional engineering contributions by extending AWSIM-Labs to support swerve and U-turn maneuvers and extending AWSIM-Script to facilitate scenario specification and execution. These extended tools, the safety benchmarks, experiment trace data, and guidelines to reproduce them are publicly available in [23].

The remainder of this paper is organized as follows. Section II summarizes the necessary background, and Section III reviews closely related work. Section IV describes the construction of the safety reference benchmarks. Section V reports the empirical evaluations on Autoware and the six end-to-end AD agents. Finally, Section VI concludes the paper.

[1]https://autoware.org/case-studies/

II. Background

A. Scenario Decomposition in the JAMA Standard

JAMA [13] proposed a structured approach for decomposing driving conditions into multiple classes of scenarios, enabling safety evaluation and analysis of ADSs on a per-scenario-class basis. This approach has subsequently been incorporated into UN Regulation No. 157 [24] and ISO 34502 [25]. In this approach, factors affecting ADS operation are categorized into three types of disturbances:

- *Perception disturbances*: referring to conditions under which the sensor system may misperceive hazards;
- *Traffic disturbances*: arising from interactions among road geometry, positions and actions of surrounding traffic ;
- *Vehicle disturbances*: referring to other factors that may make the vehicle fail to control its own dynamics, e.g., road surface conditions and weight distribution.

This work focuses on *traffic disturbances*. In real-world driving environments, an autonomous vehicle must operate alongside other traffic participants such as cars and cyclists, whose behaviors may change dynamically and unpredictably. Consequently, traffic disturbances are inevitable and represent a fundamental aspect of ADS operation. Evaluating ADS behavior under these disturbances is therefore essential for assessing the system's ability to safely handle real-world traffic situations.

Under these traffic disturbances, JAMA systematically combines variations in road geometry (e.g., intersection, non-intersection, and merge zones), relative positions of surrounding vehicles (e.g., front, lateral, and oncoming), and their motion patterns (e.g., accelerating, decelerating, turning, or swerving) to construct a structured scenario matrix. Each element in this matrix represents a distinct, abstract traffic scenario class.

Fig. 1 depicts seven scenario classes that must be considered on non-intersection roads where the ego vehicle (in blue) keeps going straight. The second, fourth, and fifth cells represent scenario classes when: a leading vehicle suddenly decelerates, a vehicle in an adjacent lane cuts into the ego lane, and a leading vehicle cuts out while there exists a stopped vehicle ahead, respectively. The present work focuses on the swerve and U-turn classes, depicted in the last two cells, as these oncoming maneuvers involve sudden intrusions into the ego lane and make motion intent difficult for the ADS to infer until the last moment.

Gaps. While JAMA defines high-level conceptual scenario classes, it does not provide sufficient detail to construct concrete scenarios. As shown in Fig. 1, the information is presented only as a matrix of scenario categories. It does not specify traffic participant behaviors or identify the relevant parameters (e.g., relative position, velocity, and timing) for each scenario class. To construct the safety reference benchmarks, we therefore need to formalize the precise maneuver, parameters, and constraints of each scenario class.

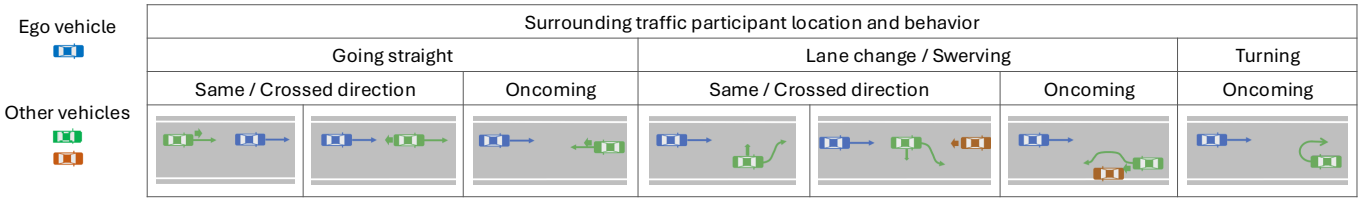


Fig. 1: Traffic disturbance scenario classes when the ego vehicle (in blue) travels straight on a non-intersection road [13].

B. JAMA's Careful Driver Model

The careful driver model simulates how a careful human driver with excellent driving skills detects hazards, makes decisions, and takes actions such as braking to avoid dangers. This model is built around the core elements of perception, judgment, and operation:

- *Risk perception*: The time it takes for the driver to identify a potential hazard in their environment. In the cut-in scenarios, this is defined as the duration for the cut-in vehicle to complete a lateral movement of 0.375 m.
- *Risk judgment*: The time needed for the driver to assess the risk and determine the necessity of an emergency brake. In the cut-in scenarios, JAMA defines this as 0.4 seconds, based on some real experiments with humans.
- *Braking operation delay*: The time elapsed between the decision to brake and actual brake application, covering the decision-making delay, releasing the accelerator pedal, transferring the foot to the brake pedal, and activating the brake. JAMA sets this parameter at 0.75 seconds, aligning with the standards used by police and domestic courts in Japan.

III. Related Work

A. Scenario-based Testing and Verification

Techniques for generating safety-critical scenarios are a central focus in research on ADS testing. A large body of work has employed *search-based approaches*, where optimization or evolutionary algorithms are used to explore the scenario space guided by well-defined fitness functions to uncover safety-critical cases [2], [3], [5]. Another line of work has explored *fuzzing-based approaches* for safety-critical scenario generation [6], [8]–[10], [26]. These methods rely on randomness: test cases are generated by stochastically perturbing inputs, often enhanced with objective functions to increase the likelihood of reaching critical cases. A third category includes *sampling-based approaches*, which probabilistically sample the parameter space of driving conditions to create a broad distribution of scenarios [4], [7].

Complementing these approaches, our work identifies safety-critical scenarios directly from the reference benchmarks. The benchmarks also offer a principled manner for distinguishing between collision-avoidable and unavoidable scenarios, a distinction that was often unclear in the previously generated scenarios. Moreover, these prior studies have focused on Apollo [27] or other AI-based perception systems, while limited attention has been given to Autoware—even though public AD services have already been deployed based on this platform.

Recently, with the rapid emergence of large language models (LLMs), researchers have begun exploring LLM-based approaches for driving scenario generation [28]–[30], though the reliability and integration into testing frameworks remain limited. In parallel, another research line leverages real-world data, e.g., crash reports and dashcam videos, to reconstruct dangerous scenarios [31]–[36]. While realistic, this approach may struggle with the scarcity and non-diversity of data.

Using the JAMA standard [13], Tran et al. [16] evaluated the safety of Autoware (July 2024 release) in deceleration, cut-in, and cut-out scenarios (depicted in the 2nd, 4th, and 5th cells in Fig. 1). While JAMA provides safety benchmarks for these scenarios, the underlying models or simulation code used to derive them is not disclosed. In this work, we shift the focus to oncoming traffic scenarios, which present more intent-prediction challenges than the same-direction interactions studied previously. Unlike those scenarios where the general travel path is relatively predictable, oncoming maneuvers require the ADS to distinguish between routine approaches and sudden path intrusions under high relative velocities. Because JAMA provides only a high-level matrix of scenario classes, constructing reference benchmarks requires us to identify the relevant parameters and formalize the participant behaviors. In addition to Autoware (February 2025 release), the present work also evaluates the safety of six end-to-end learning-based AD agents, examining the safety performance difference between them.

B. Avoidable/Unavoidable Collisions

Calo et al. [11], [12] have highlighted the problem of distinguishing between avoidable and unavoidable collisions, relating it to the oracle problem in software testing [37]. In these studies, collision avoidability is defined with respect to *system configuration* changes: a collision in a given scenario is considered avoidable if it can be prevented by reconfiguring the ADS while keeping the scenario unchanged. Specifically, the configuration corresponds to the internal cost-function weights of the ADS's path planning module, which assign penalties to different factors, such as excessive acceleration and the occurrence of a dangerous situation in the generated path. A collision is therefore deemed avoidable if there exists an alternative assignment of these weights under which the ADS can avoid the collision in the same scenario. While this definition provides a clear notion of avoidability, it has two key

limitations. First, it implicitly assumes that unsafe behavior arises from suboptimal configuration of an otherwise correct system; while in fact, collisions may stem from flaws in the path planning logic itself, as well as perception or control modules that cannot be resolved through configuration changes alone. Second, the definition is tightly coupled to the specific design of the path planning module under test and its weight-based cost function, limiting its generality across different ADSs.

IV. Safety Reference Benchmarks

This section first provides some definitions related to the safety reference benchmarks, and then details the construction of the benchmarks for the two oncoming traffic scenario classes.

A. Definitions

Definition 1 (Actor state)**.** *For an actor a (ego or NPC), its state at time t is denoted as $x_a(t) = \langle p_a(t), \theta_a(t), v_a(t), a_a(t) \rangle$, where $p_a(t), \theta_a(t), v_a(t), a_a(t) \in \mathbb{R}^3$, denoting the position, orientation (in Euler angles), velocity, and acceleration, respectively.*

Definition 2 (Trajectory)**.** *A trajectory $\pi_a = \{x_a(t) \mid t \in [0, T]\}$ of actor a is a time-indexed sequence of states defined over a finite time horizon T.*

Definition 3 (Scenario)**.** *A scenario s is a tuple $s = \langle x_e(0), G_e, \mathcal{A}, \Pi, \Theta \rangle$, where:*

- *$x_e(0)$ is the initial state of the ego vehicle;*
- *G_e is the target destination of the ego vehicle;*
- *$\mathcal{A}$ is a finite set of NPCs (non-ego actors);*
- *$\Pi = \{\pi_a \mid a \in \mathcal{A}\}$ is the set of NPC trajectories;*
- *Θ represents relevant configuration parameters of the ego vehicle, e.g., speed limit.*

When executing a scenario s with an ADS, the ego vehicle's trajectory is produced by the ADS given the initial pose and target destination. When the ego vehicle is governed by the JAMA's careful driver model, a reference ego trajectory π_{ref} is generated.

Definition 4 (Avoidability oracle)**.** *Given a scenario s, the avoidability oracle $O(s)$ outputs:*

- `collision` *if a collision occurs between the ego vehicle following the reference trajectory π_{ref} and any NPC trajectory π_a in Π;*
- `no_collision` *otherwise.*

A scenario s is classified as *collision-avoidable* if $O(s) =$ `no_collision`, and as *collision-unavoidable* otherwise. It should be noted that the reference ego trajectory does not represent the set of all possible safe behaviors. Instead, it provides a conservative baseline reference for assessing avoidability. A scenario classified as *collision-avoidable* under this model indicates that the careful driver's reaction (i.e., braking alone) is insufficient to prevent a collision, although alternative evasive maneuvers (e.g., steering and braking) may exist.

Definition 5 (Safety reference benchmark)**.** *A safety reference benchmark is a finite set $\mathcal{B} = \{\langle s_i, o_i \rangle \mid i = 1, \ldots, N\}$, where each s_i is a scenario and $o_i = O(s_i)$ is the corresponding collision-avoidability outcome.*

The benchmark serves as a reference baseline for evaluating an ADS by comparing the system execution trace of scenario s_i against the expected outcome o_i. An ADS violates the benchmark in a collision-avoidable scenario if it produces a collision where the reference outcome is `no_collision`.

B. Methodology Overview

This work follows a structured methodology to construct and apply safety reference benchmarks for evaluating ADSs. The overall process consists of five main steps.

NPC Trajectory Definition: For each considered scenario class, we define the trajectories of non-ego traffic participants. These trajectories encode the intended maneuver of the surrounding vehicles, such as a U-turn or a temporary swerve, and are parameterized by scenario variables, e.g., longitudinal and lateral velocities.

Reference Ego Response Formalization: Given the NPC trajectories and scenario parameters, we formalize the ego vehicle's response using the careful driver model. This model deterministically generates a reference ego trajectory by explicitly modeling risk perception, risk judgment, and braking action.

Safety Reference Benchmark Construction: From the NPC trajectories and the reference ego trajectory, the safety reference benchmark is constructed. Each element of the benchmark corresponds to a concrete scenario paired with a collision-avoidability outcome, indicating whether a collision occurs under the reference behavior.

Safety-Critical Scenario Identification: We systematically identify safety-critical scenarios from the constructed benchmark. These scenarios lie near the boundary between collision-avoidable and collision-unavoidable conditions and are therefore collision-prone. This process is driven directly by the benchmark structure, rather than relying on search-based methods or extensive random exploration.

ADS Evaluation Against Benchmark: Finally, we evaluate an ADS by executing the identified safety-critical scenarios and comparing the execution traces against the benchmark outcomes. Collisions occurring in scenarios classified as collision-avoidable indicate safety violations, while adherence to the benchmark demonstrates correct or improved safety behavior.

The following subsections instantiate this methodology for two oncoming traffic scenario classes, U-turn and swerve.

C. U-turn Scenarios

Fig. 2 illustrates a representative U-turn scenario, assuming left-hand traffic conditions. We consider a straight, two-way road, potentially with multiple lanes in each direction. The ego vehicle (shown in blue) travels straight in its own lane at a constant speed of v_e m/s. An oncoming passenger car (shown in orange) approaches from the opposite direction in

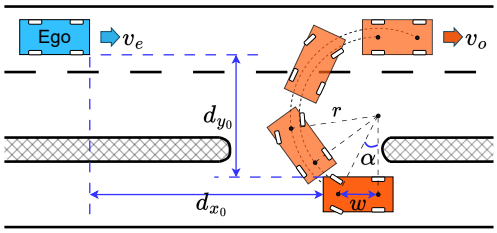


Fig. 2: U-turn scenario illustration.

TABLE I: Parameters and their configuration in U-turn scenarios to derive the result visualized in Fig. 3. The values in the left table are from JAMA [13].

Parameter	Value
τ_{judge}	0.4 s
τ_{delay}	0.75 s
τ_{jerk}	0.6 s
$a_{\min}$	-7.6 m/s^2

Parameter	Value
v_e	$\{14, 20, 25, 30, 35, 40, 45, 50\}$ km/h
v_o	$\{10, 15\}$ km/h
d_{x_0}	9–50 m
w	2.5 m
α	$\pi/6$ rad

the innermost lane at a constant speed of v_o m/s. We also call the oncoming vehicle a non-player character (NPC).

When the longitudinal distance between the two vehicles is d_{x_0}, the oncoming vehicle initiates a U-turn maneuver. During the maneuver, the vehicle is assumed to keep the steering wheel at its maximum angle. For simplicity, we restrict vehicle motion to the horizontal plane and approximate both vehicles as two-dimensional rectangles, abstracting away vertical dynamics.

1) NPC Trajectory: The turning radius r of the U-turn is:

$$r = \frac{w}{\sin(\alpha)} \tag{1}$$

where w is the wheelbase of the vehicle and α is the average of the maximum front wheel steering angles (inner and outer) measured relative to the vehicle longitudinal axis.

Turning Center: The turning center C lies on the line passing through the vehicle's rear wheels, at a distance η from the midpoint between them, where:

$$\eta = \frac{w}{\tan(\alpha)} \tag{2}$$

Pose Update: Let $\psi_a(t)$ denote the vehicle heading angle at time t. As the yaw rate during the U-turn is $\frac{v_o}{r}$, the updated heading angle after a time increment Δt during the U-turn is:

$$\psi_a(t + \Delta t) = \psi_a(t) - \frac{v_o \Delta t}{r} \tag{3}$$

Let the midpoint between the two front wheels represent the vehicle position $p_a(t)$. The updated position is:

$$p_a(t + \Delta t) = \text{rotate}(p_a(t), C, -\frac{v_o \Delta t}{r}) \tag{4}$$

where $\text{rotate}(X, Y, \phi)$ is a function rotating point X around point Y counter-clockwise by an angle ϕ.

Given $p_a(t)$ and $\psi_a(t)$, the trajectory of the oncoming vehicle can be computed.

2) Reference Ego Response: When observing the U-turn, the ego vehicle responds following the careful driver model, considering braking as the sole evasive maneuver.

Risk Perception: The ego driver is assumed to observe the risk when the oncoming vehicle first intrudes into the roadway traveling in the same direction as the ego vehicle, i.e., any lane of the ego's direction of travel, not only the lane occupied by the ego. Formally, the perception moment t_0 (Section II-A) is defined as the earliest time when one vertex of the oncoming vehicle crosses into any lane in the ego vehicle's direction of travel. This definition reflects a common-sense interpretation, in which a potential hazard should be observed as soon as an oncoming vehicle enters the ego vehicle's side of the roadway.

Reaction and Braking: After t_0, the driver requires τ_{judge} seconds to assess the risk and decide to brake to mitigate the risk. It takes a further τ_{delay} seconds of delay from the decision moment until the brake is actually applied. Thus, braking begins at:

$$t_1 = t_0 + \tau_{\text{judge}} + \tau_{\text{delay}} \tag{5}$$

From t_1, the ego vehicle's deceleration increases linearly over a jerk period τ_{jerk} until reaching the minimum acceleration $a_{\min}$ ($a_{\min} < 0$). For a sufficiently small Δt, the ego's acceleration at $t + \Delta t$ is:

$$a_e(t+\Delta t) = \begin{cases} 0 & \text{if } t < t_1 \\ a(t) + a_{\min} * \frac{\Delta t}{\tau_{\text{jerk}}} & \text{if } t_1 \le t < t_1 + \tau_{\text{jerk}} \\ a_{\min} & \text{if } t \ge t_1 + \tau_{\text{jerk}} \end{cases} \tag{6}$$

Given $a_e(t)$, along with the initial position and speed, the ego vehicle's trajectory over time can be computed. Based on this reference trajectory and the U-turn trajectory, it is possible to determine whether a collision occurs.

3) Safety Benchmark with Collision-avoidable Boundary: We simulate the trajectories of the ego and oncoming vehicles for each scenario derived from the parameter configuration shown in Table I. The values of τ_{judge}, τ_{delay}, τ_{jerk}, and $a_{\min}$ are set according to the JAMA standard. The oncoming vehicle's speed was set to 10 km/h and 15 km/h, reflecting the fact that U-turn maneuvers are typically performed at low speeds. The oncoming vehicle is assumed to be a passenger car, with a wheelbase of 2.5 m and an average steering angle of 30 degrees, following common practice. We consider two cases, when the ego vehicle travels in the innermost lane and in the lane adjacent to the innermost lane. Two corresponding values of d_{y_0} are derived for these two cases.

Fig. 3 visualizes a subset of the benchmark for the case where the ego vehicle travels in the lane adjacent to the innermost lane, and the oncoming vehicle travels at 10 km/h. The x-axis represents the initial longitudinal distance d_{x_0} (in meters), while the y-axis represents the ego vehicle's speed (in km/h). Each point represents a single scenario.

Collision Avoidable vs. Unavoidable Distinction: In the chart in Fig. 3, green points correspond to scenarios classified as collision-avoidable, while red points correspond to

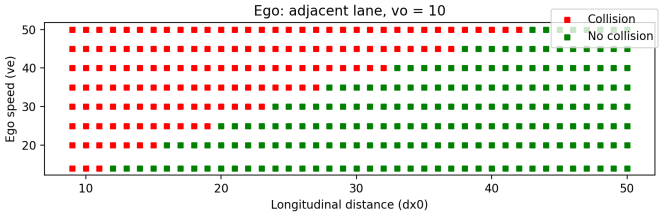


Fig. 3: Visualization of a subset of the safety reference benchmark for U-turn scenarios, when the ego vehicle travels in the lane adjacent to the innermost lane, and the NPC speed is 10 km/h.

collision-unavoidable scenarios. For instance, when the ego and oncoming vehicle speeds v_e and v_o are 20 km/h and 10 km/h, respectively, a collision occurs if the initial longitudinal distance d_{x_0} is 15 m, whereas no collision occurs if d_{x_0} is 16 m. The *collision-avoidable boundary* can thus be intuitively interpreted as the zigzag line separating the green and red regions. To demonstrate safety, an ADS is expected to behave safely (at least) for all scenarios within the green region.

Safety-Critical Scenario Identification: The safety reference benchmark further enables systematic identification of safety-critical scenarios. In Fig. 3, these safety-critical scenarios correspond to the points located close to the collision-avoidable boundary, as small parameter variations (e.g., 1 m difference in d_{x_0}) can determine whether a collision is avoidable or not. Such scenarios are particularly valuable for evaluation because they place the ADS under tight safety margins and limited reaction time, requiring precise and timely responses. The systematic identification of these scenarios from the reference benchmark complements the search-based generation methods commonly adopted in prior studies, enabling efficient discovery of system weaknesses while ensuring that evaluation is grounded in a formal and interpretable reference model. Section V presents an experimental evaluation on Autoware and different AI-based AD agents when exposed to these safety-critical scenarios.

We assume that the ego vehicle uses braking as the sole evasive maneuver. This choice intentionally reflects a conservative baseline safety requirement: braking represents the most fundamental and widely implemented risk-mitigation behavior in ADSs. While more advanced evasive actions, such as steering maneuvers, may also avoid a collision in some situations, their availability and reliability depend on additional perception, planning, and control capabilities that may not be consistently implemented across systems. By restricting the reference behavior to braking, the proposed benchmark identifies scenarios in which a collision should be avoidable even under a minimal safety response. In other words, if a reasonably behaving vehicle that only performs controlled braking can avoid the collision, then any ADS implementing basic collision-mitigation behavior is expected to maintain safety in that scenario.

The complete benchmark, together with the simulation scripts used to derive it and animations visualizing the interactions between the two vehicles, is publicly available in [23]. We emphasize that the parameter configuration in Table I represents one concrete instantiation that follows the JAMA standard [13]. The purpose of these parameters is not to model the full variability of human driving behavior, but rather to provide a consistent baseline for determining collision avoidability. The scripts in [23] allow the benchmark to be regenerated under alternative parameter configurations.

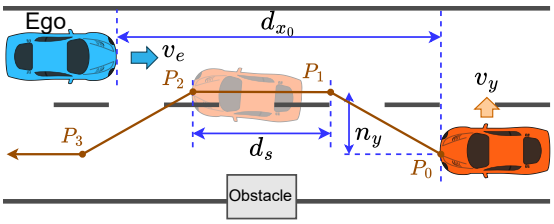


Fig. 4: Swerve scenario illustration.

D. Swerve Scenarios

Fig. 4 illustrates a representative swerve scenario. The orange car, traveling at a constant speed of v_o in the opposite direction to the ego vehicle, needs to temporarily swerve out of its lane to avoid an obstacle before returning to its original lane.

The trajectory of the oncoming vehicle is represented by four waypoints P_0, P_1, P_2, and P_3, corresponding to the front-center point of the vehicle at key moments of the maneuver. Specifically, P_0 marks the start of the swerve, P_1 the end of the outward lateral displacement, P_2 the continuation of forward travel while maintaining the lateral offset, and P_3 the point where the vehicle merges back into its original lane. The relative positions of the waypoints with respect to P_0 are defined as:

$$P_1(-n_x, n_y), P_2(-n_x - d_s, n_y), P_3(-2n_x - d_s, 0)$$

Here, n_y denotes the lateral offset of the swerve. These four waypoints provide a simple and reproducible description of the swerving maneuver while capturing its key characteristic: a temporary lateral intrusion into the ego vehicle's lane. This simplified trajectory does not aim to reproduce the full diversity of real-world swerving behaviors. Instead, it provides a canonical maneuver that captures the key safety-critical element of the scenario, while enabling systematic analysis of the resulting avoidability boundary. Alternative trajectory shapes could be incorporated within the same benchmark generation framework by modifying the waypoint configuration.

Let v_y be the average lateral velocity of the movement between P_0 and P_1 (and similarly between P_2 and P_3), n_x can be derived from v_o, v_y, and n_y as follows:

$$n_x = n_y * \frac{\sqrt{v_o^2 - v_y^2}}{v_y} \tag{7}$$

Once the vehicle reaches waypoint P_i, steering is applied to gradually adjust the heading toward the next waypoint P_{i+1}.

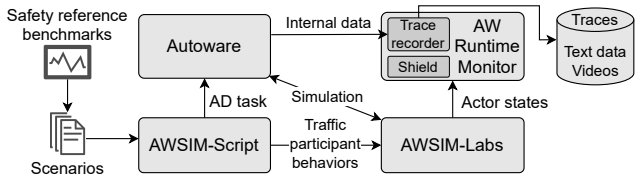


Fig. 5: Tools and overall experimental workflow for the experiments on Autoware.

Following the pure pursuit algorithm [38], the instantaneous yaw rate is expressed as:

$$\omega(t) = \frac{2v_o \sin(\alpha(t))}{l(t)} \tag{8}$$

where $\alpha(t)$ is the angle and $l(t)$ the lookahead distance to the target waypoint. Given $\omega(t)$, the vehicle's heading angle $\psi_a(t)$ and position $p_a(t)$ can be integrated over time, thus yielding the full trajectory of the swerve maneuver.

The ego driver is assumed to perceive the risk once the oncoming vehicle first enters the ego's lane and to respond in the same manner as in the U-turn case. The safety reference benchmark for swerve scenarios is constructed in the same way as for the U-turn scenarios.

V. Safety Evaluation Experiments

A. Research Questions

Experiments were conducted on Autoware—a modular, real-world ADS—and six end-to-end AD agents, to answer the following research questions (RQs):

- **RQ1**: Are the safety reference benchmarks valid and sound for ADS evaluation?
  This RQ examines whether the benchmarks meaningfully characterize collision avoidability and provide a reliable basis for ADS safety assessment.
  - **RQ1.1**: Can the safety reference benchmarks effectively expose unsafe behaviors in existing ADS implementations?
  - **RQ1.2**: Do scenarios classified as collision-avoidable by the reference benchmarks indeed admit collision-free behavior by at least one ADS or agent?
- **RQ2**: How does safety performance differ across ADS implementations when evaluated using the same reference benchmarks?
  This RQ directly compares how different ADS implementations behave under the same benchmark scenarios, including production-grade systems, learning-based agents, and a shielded variant of Autoware.
  - **RQ2.1**: How does safety performance differ between a production-grade ADS (Autoware) and end-to-end learning-based ADSs when evaluated under identical benchmark scenarios?
  - **RQ2.2**: How does adding a controller safety shield change Autoware's safety performance under the same benchmark scenarios?
- **RQ3**: Under which scenario conditions and characteristics do ADSs often fail to maintain safety?

B. Autoware

Autoware is an open-source, production-grade ADS [14], with Level 4 of autonomy according to the SAE J3016 classification [39]. Unlike end-to-end learning-based AD solutions, Autoware follows a classical modular architecture, explicitly separating perception, planning, and control [1]. The experiments were conducted on Autoware version 0.41.2 (released February 20, 2025). The ego vehicle is equipped with one camera and three LiDAR sensors.

Experimental Environment and Supporting Tools: The experiments with Autoware were performed in AWSIM-Labs [15], a simulator developed by the Autoware Foundation for launching Autoware in a virtual environment. To specify and execute traffic scenarios, we build upon AWSIM-Script [16], a scenario specification language. In this work, we extend both AWSIM-Labs and AWSIM-Script to support the simulation of U-turn and swerve scenarios.

a) Extended AWSIM-Script and AWSIM-Labs: AWSIM-Labs was extended to support U-turn and swerve maneuvers, while AWSIM-Scripts was extended to facilitate the specification of these maneuvers. In addition, the new version of AWSIM-Script enables multiple scenarios to be loaded and executed sequentially without restarting either the ADS or the simulator, thereby improving experimental efficiency.

b) Re-implementation of Runtime Monitor: A runtime monitor was introduced in [16] to record execution data in the AWSIM-Labs–Autoware simulation environment and export it as trace files for post-simulation analysis. This monitor was implemented inside AWSIM-Labs and functioned solely as a passive recorder, without the ability to check desired safety requirements or intervene at runtime. To address this limitation, we re-implemented the monitor as an independent and extensible tool, named AW-Runtime-Monitor, fully decoupled from AWSIM-Labs. AW-Runtime-Monitor supports recording control commands generated by the control module, as well as video captured from the camera mounted on the vehicle, in addition to the text-based data. The independent design further allows integration of online runtime verification components, such as a safety shield.

The integration of these tools and the overall experimental workflow are illustrated in Fig. 5. From the reference benchmarks presented in the last section, we specified a suite of AWSIM-Script scenarios. When a scenario is fed into AWSIM-Script, it initializes Autoware to perform the AD task (including the initial position and desired destination) and AWSIM-Labs to simulate the behavior of other traffic participants. AW-Runtime-Monitor records dynamic information of all actors and Autoware's internal data (e.g., detected objects, camera images, and control commands), storing them as trace files at the end of each run.

Shielded Autoware: In addition to the baseline Autoware system, we evaluate a variant of Autoware integrated with a controller safety shield using the same experimental setup.

The shield is implemented within AW-RUNTIME-MONITOR and operates at runtime by monitoring control commands and intervening when safety constraints are violated. For each control command produced by the control module and each detected object provided by the perception module, the shield estimates the minimum time-to-collision (TTC) with the object. When the estimated TTC falls below a threshold of 2 seconds, the shield immediately triggers automatic emergency braking (AEB). This shielded Autoware serves to assess whether the safety reference benchmarks can validate safety improvements in a modified system.

C. End-to-end Learning-based AD Agents

To complement the evaluation of a modular, real-world ADS, we also conducted experiments with six end-to-end learning-based AD agents in the CARLA simulator [21]: InterFuser [17], Learning from All Vehicles (LAV) [18], TransFuser [19], Latent TransFuser [20], Late Fusion [19], and Geometric Fusion [19]. All six agents are implemented using deep neural networks (DNNs) and learn driving policies in an end-to-end manner, mapping raw or intermediate sensor representations directly to driving commands or waypoints, without an explicit modular decomposition into perception, planning, and control. To run these agents, users must manually provide a sequence of waypoints guiding the ego vehicle from the initial position to the desired destination. This differs from Autoware, where only the initial position and goal are specified, and the route is generated automatically by the planning module. Except for Late Fusion and Geometric Fusion, all these end-to-end solutions were submitted to the CARLA Leaderboard [40], which hosts some AD challenges based on the CARLA simulator.

Experiments with these six agents were conducted using the PCLA framework [41], which supplies the pretrained models and a unified interface for executing these agents in the CARLA simulator. ScenarioRunner [42], a Python interface for scenario specification in CARLA, was used to specify the U-turn and swerve maneuvers.

D. Experiment Setup

We conducted experiments with representative scenarios that are safety-critical and classified as collision-avoidable with respect to our safety reference benchmarks. As illustrated Fig. 3, these scenarios correspond to the green points located near the zigzag boundary separating the collision-avoidable (green) and collision-unavoidable (red) regions. As a minimum safety requirement, Autoware and all AI agents should be able to avoid collisions in all such scenarios.

In all experiments, the oncoming vehicle traveled at either 10 or 15 km/h, representing typical low-speed U-turn and swerve maneuvers in the real world. The maximum achievable speeds of the six end-to-end AD agents are generally limited; for example, InterFuser is capped at approximately 18 km/h (5 m/s), while TransFuser and Latent Transfuser are at 14.4 km/h (4 m/s). For this reason, for experiments involving the six AD agents, the ego-vehicle target speed was fixed at 14 km/h. For Autoware, which supports higher operating speeds, we additionally evaluated ego speeds of 20, 30, and 40 km/h in the swerve scenarios, and 20, 25, 30, 35, and 40 km/h in the U-turn scenarios.

The U-turn experiments consist of two configurations: (1) the ego vehicle traveling in the innermost lane (i.e., rightmost lane under left-hand traffic and leftmost lane otherwise), and (2) the ego vehicle traveling in the lane adjacent to the innermost lane. For the swerve scenarios, the lateral velocity of the oncoming vehicle was set to 1.0, 1.2, and 1.4 m/s, corresponding to the most commonly observed values reported in the JAMA standard [13].

To account for non-determinism in simulation and AI components, each concrete scenario was executed three times. For each run, execution traces were collected and analyzed to determine whether a collision occurred. For collision-free runs, we further estimated the minimum TTC value from the post-simulation execution traces. This TTC was computed at each simulation frame under the assumption that both vehicles continue with their instantaneous velocities. The minimum TTC over the entire run serves as an indicator of how close to a collision, thereby providing a quantitative measure of safety margin even in the absence of an actual collision.

E. Results

Tables II–IV summarize the experimental results for the U-turn scenarios, while Tables V and VI report the results for the swerve scenarios. Recall that v_e and v_o are the speeds of the ego and the oncoming vehicles (in km/h), v_y the lateral velocity of the NPC (in m/s), and d_{x_0} the longitudinal distance between the two vehicles when the maneuver starts (in m). A check mark ✓ denotes non-collision, while a cross mark ✗ indicates a collision. For each collision-free run, the value in parentheses reports the minimum TTC in seconds.

RQ1.1. Collisions were observed in both U-turn and swerve scenarios across all evaluated systems. In total, 176 out of 426 runs resulted in collisions, showing that the proposed benchmarks are effective at exposing unsafe behaviors. Notably, Autoware, despite being a production-grade ADS, exhibited a substantial number of safety weaknesses, particularly in swerve scenarios and U-turn scenarios where the ego vehicle traveled in the lane adjacent to the innermost lane. This highlights the need for further systematic analysis and safety improvement studies on Autoware, for which safety-focused research remains limited compared to competing platforms such as Baidu Apollo [27] (e.g., [26], [43]–[47]).

RQ1.2. At the same time, for every scenario considered, covering both U-turn and swerve maneuvers, and low to high speeds, there existed at least one collision-free execution. For example, Autoware consistently avoided collisions in scenarios where the ego vehicle traveled in the innermost lane, while shielded Autoware could do so in the swerve scenarios. This empirical evidence confirms that scenarios classified as collision-avoidable by the safety reference benchmarks do admit collision-free behavior in practice, supporting the soundness of the presented avoidability oracle.

TABLE II: Results of U-turn scenarios on Autoware and Shielded Autoware with ego speeds of 20–40 km/h. Experiments on Shielded Autoware with ego traveling in the innermost lane were omitted, as the original Autoware behaved safely in these scenarios.

Ego in innermost lane

			No Collision (Min TTC)	
v_o	v_e	d_{x_0}	Autoware	Shielded Autoware
10	20	17	✓✓✓ (0.9 0.93 0.89)	N/A
	25	21	✓✓✓ (1.02 1.05 1.04)	
	30	26	✓✓✓ (1.27 1.29 1.33)	
	35	31	✓✓✓ (1.49 1.32 1.47)	
	40	35	✓✓✓ (1.22 1.32 1.27)	
15	20	15	✓✓✓ (0.93 0.92 0.93)	
	25	18	✓✓✓ (1.0 1.0 0.99)	
	30	21	✓✓✓ (1.05 1.05 1.05)	
	35	24	✓✓✓ (1.13 1.12 1.13)	
	40	28	✓✓✓ (1.16 1.16 1.16)	

Ego in adjacent lane

			No Collision (Min TTC)	
v_o	v_e	d_{x_0}	Autoware	Shielded Autoware
10	20	17	✓✓✓ (0.51 0.52 0.91)	✓✓✓ (0.76 0.94 1.32)
	25	21	✓✗✓ (0.29 0.32)	✓✓✓ (1.07 1.11 0.67)
	30	25	✗✗✗	✓✓✓ (1.15 1.08 1.18)
	35	30	✓✗✗ (0.31)	✓✓✓ (1.09 1.29 1.17)
	40	35	✗✗✗	✓✓✓ (1.19 1.55 1.07)
15	20	15	✓✓✓ (0.79 0.8 0.79)	✓✓✓ (0.79 0.87 0.92)
	25	19	✓✓✓ (0.94 0.95 0.95)	✓✓✓ (0.89 0.96 0.95)
	30	22	✓✓✓ (0.95 0.95 0.97)	✓✓✓ (1.07 1.02 0.89)
	35	26	✓✓✓ (1.06 1.07 1.07)	✓✓✓ (1.16 1.12 1.18)
	40	31	✓✓✓ (1.14 1.17 0.83)	✓✓✓ (1.23 1.33 1.27)

TABLE III: Results of U-turn scenarios on Autoware, InterFuser, LAV, TransFuser, and Latent TransFuser with ego speed of 14 km/h.

				No Collision (Min TTC)				
v_e	Lane	v_o	d_{x_0}	Autoware	InterFuser	LAV	TransFuser	Latent TransFuser
14	Innermost	10	12	✓✓✓ (0.68 0.7 0.69)	✓✓✓ (0.66 0.62 0.65)	✓✓✓ (0.91 1.08 1.08)	✓✓✓ (0.7 0.69 0.72)	✓✓✓ (0.73 0.77 0.68)
		15	10	✓✓✓ (0.7 0.71 0.71)	✓✓✓ (1.02 0.77 1.03)	✓✓✓ (0.47 0.53 0.54)	✓✓✓ (1.03 0.79 0.81)	✓✓✓ (1.01 1.02 1.01)
	Adjacent	10	12	✓✓✓ (0.4 0.41 0.42)	✗✗✗	✗✗✗	✓✓✓ (0.97 0.95 0.96)	✗✗✗
		15	10	✓✓✓ (0.64 0.64 0.65)	✗✗✗	✗✗✗	✓✓✗ (0.63 0.63)	✗✗✗

TABLE IV: Results of U-turn scenarios on Late Fusion and Geometric Fusion with ego speed of 14 km/h.

				No Collision (Min TTC)	
v_e	Lane	v_o	d_{x_0}	Late Fusion	Geometric Fusion
14	Innermost	10	12	✓✓✓ (0.83 0.79 0.78)	✓✓✓ (0.91 0.91 0.86)
		15	10	✓✓✓ (1.03 0.81 1.03)	✓✓✓ (0.85 0.73 0.83)
	Adjacent	10	12	✗✗✗	✗✗✗
		15	10	✗✗✗	✗✗✗

RQ1: The benchmarks expose unsafe behaviors in Autoware and multiple end-to-end learning-based agents, even in scenarios classified as collision-avoidable. At the same time, collision-free executions are observed in such scenarios for at least one system (e.g., shielded Autoware), confirming that the avoidability classification is achievable in practice rather than purely theoretical.

RQ2.1. In the U-turn scenarios, Autoware outperformed the six end-to-end AD agents in terms of collision avoidance when the ego vehicle traveled in the lane adjacent to the innermost lane at a low speed of 14 km/h. When the ego vehicle traveled in the innermost lane, this difference is no longer observed, as all systems were able to avoid collisions. However, it should be noted that the relatively low maximum speed limits of the six AD agents may mask potential safety issues that could arise at higher ego speeds.

In the swerve scenarios with the ego speed of 14 km/h, five of the six AD agents failed to avoid collisions in all runs, whereas the LAV agent consistently avoided collisions across all 18 runs. Notably, LAV even outperformed Autoware, which exhibited collisions in scenarios with low lateral swerve velocities. This performance advantage was reflected not only in the number of collision-free runs but also in the higher minimum TTC values observed in LAV's collision-free executions.

RQ2.2. Integrating the controller safety shield completely eliminated the collisions observed in all U-turn scenarios. A similar effect was observed in the swerve scenarios with an NPC speed of 10 km/h. For swerve scenarios with an NPC speed of 15 km/h, a small number of collisions (three runs) still occurred at high ego speeds; nevertheless, the shield substantially reduced the overall number of collisions compared to the unshielded system.

RQ2: When evaluated under the same reference benchmarks, the production-grade Autoware system and end-to-end learning-based agents exhibited distinct safety weakness patterns, with neither consistently dominating across all scenarios. Notably, Autoware did not always surpass simpler learning-based agents in safety. Integrating a controller safety shield into Autoware significantly reduced collisions under the same scenarios.

RQ3. In the U-turn scenarios, while both Autoware and the six AD agents could operate safely when the ego vehicle traveled in the innermost lane, they generally struggled to avoid collisions when the ego vehicle traveled in the adjacent lane. This might be because when traveling in a non-innermost

TABLE V: Results of swerve scenarios on Autoware and six AD agents with ego speed of 14 km/h. Latent TF stands for Latent TransFuser.

v_e	v_o	v_y	d_{x_0}	No Collision (Min TTC)						
				Autoware	InterFuser	LAV	TransFuser	Latent TF	Late Fusion	Geometric Fusion
14	10	1.0	18	✗✗✗	✗✗✗	✓✓✓ (0.76 0.80 0.77)	✗✗✗	✗✗✗	✗✗✗	✗✗✗
		1.2	17	✓✓✓ (0.09 0.23 0.07)	✗✗✗	✓✓✓ (0.84 0.84 0.84)	✗✗✗	✗✗✗	✗✗✗	✗✗✗
		1.4	15	✓✓✓ (0.16 0.26 0.37)	✗✗✗	✓✓✓ (0.81 0.82 0.82)	✗✗✗	✗✗✗	✗✗✗	✗✗✗
	15	1.0	23	✓✗✗ (0.13)	✗✗✗	✓✓✓ (0.57 0.54 0.54)	✗✗✗	✗✗✗	✗✗✗	✗✗✗
		1.2	20	✓✓✓ (0.09 0.19 0.14)	✗✗✓ (0.34)	✓✓✓ (0.55 0.55 0.57)	✗✗✗	✗✗✗	✗✗✗	✗✗✗
		1.4	18	✓✓✓ (0.19 0.24 0.26)	✗✗✗	✓✓✓ (0.49 0.57 0.54)	✗✗✗	✗✗✗	✗✗✗	✗✗✗

TABLE VI: Results of swerve scenarios on Autoware and Shielded Autoware with ego speeds of 14–40 km/h.

NPC speed (v_o): 10 km/h

v_e	v_y	d_{x_0}	No Collision (Min TTC)	
			Autoware	Shielded Autoware
14	1.0	18	✗✗✗	✓✓✓ (0.13 0.84 1.23)
	1.2	17	✓✓✓ (0.09 0.23 0.07)	✓✓✓ (0.51 1.12 1.28)
	1.4	15	✓✓✓ (0.16 0.26 0.37)	✓✓✓ (0.2 1.39 1.17)
20	1.0	23	✗✗✗	✓✓✓ (0.32 1.44 1.92)
	1.2	20	✓✓✗ (0.02 0.02)	✓✓✓ (1.24 0.99 1.23)
	1.4	18	✓✗✓ (0.01 0.11)	✓✓✓ (0.95 0.92 1.11)
30	1.0	31	✓✓✓ (0.06 0.11 0.09)	✓✓✓ (0.54 0.02 0.33)
	1.2	27	✓✓✓ (0.03 0.09 0.03)	✓✓✓ (0.42 0.36 0.67)
	1.4	24	✓✓✓ (0.06 0.01 0.17)	✓✓✓ (0.74 0.1 0.39)
40	1.0	39	✗✗✗	✓✓✓ (0.16 0.1 0.17)
	1.2	34	✗✗✗	✓✓✓ (0.37 0.21 0.37)
	1.4	30	✗✓✓ (0.07 0.06)	✓✓✓ (0.22 0.03 0.30)

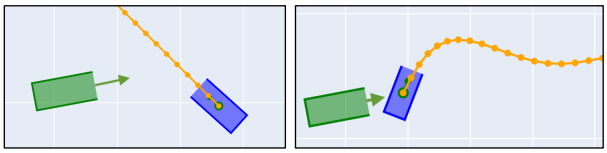
NPC speed (v_o): 15 km/h

v_e	v_y	d_{x_0}	No Collision (Min TTC)	
			Autoware	Shielded Autoware
14	1.0	23	✓✗✗ (0.13)	✓✓✓ (0.93 0.9 0.63)
	1.2	20	✓✓✓ (0.09 0.19 0.14)	✓✓✓ (1.0 0.21 1.05)
	1.4	18	✓✓✓ (0.19 0.24 0.26)	✓✓✓ (0.81 0.39 0.26)
20	1.0	27	✗✗✗	✓✓✓ (0.36 0.56 0.71)
	1.2	23	✗✗✗	✓✓✓ (0.7 0.66 1.01)
	1.4	20	✗✗✗	✓✓✓ (0.44 0.43 0.47)
30	1.0	35	✗✗✗	✓✗✓ (0.21 0.27)
	1.2	29	✗✗✗	✓✓✓ (0.02 0.26 0.26)
	1.4	26	✗✗✗	✓✓✓ (0.22 0.35 0.23)
40	1.0	43	✗✗✗	✓✓✓ (0.38 0.43 0.28)
	1.2	36	✗✗✗	✗✓✓ (0.24 0.28)
	1.4	31	✗✗✗	✓✗✓ (0.08 0.09)

lane, the systems may pay less attention to the oncoming vehicle during decision-making; while when the ego vehicle travels in the innermost lane, where the oncoming vehicle is geometrically closer and poses a more immediate threat, the systems appear to respond more conservatively. In addition, Autoware primarily exhibited weaknesses at high ego speeds.

The swerve scenarios were generally more challenging for Autoware and for all learning-based agents except LAV. At low ego speeds (e.g., 14 km/h), scenarios with low lateral swerve velocities (1.0 m/s) were more problematic than those with higher lateral velocities. This can be explained as under a fixed speed, the higher the lateral velocity, the quicker the oncoming vehicle swerves and returns to its original lane. At higher ego speeds, Autoware increasingly struggled to maintain safety. Note that even in scenarios where collisions were avoided (e.g., ego/NPC speeds of 30/10 km/h), the minimum TTC values were very low (≤ 0.1 s), indicating near-collision situations. This observation was further validated by the shielded Autoware, where collisions still occurred in three runs at ego speeds of 30 and 40 km/h.

RQ3: Unsafe behaviors were observed more frequently in swerve scenarios than in U-turn scenarios. They occurred most often when the ego vehicle traveled in the lane adjacent to the innermost lane in U-turn scenarios, in swerve maneuvers with low lateral velocity, and in both U-turn and swerve scenarios at higher ego speeds.



(a) 1.7 seconds prior to impact (b) 0.7 seconds prior to impact

Fig. 6: Visualization of a U-turn collision case illustrating prediction errors: ego vehicle (green), perceived vehicle (blue), and predicted path (orange) at 1.7 s and 0.7 s before impact.

Factors Contributing to Autoware Collisions: We analyzed the post-simulation execution traces from Autoware collision runs and found that perception inaccuracy could be a primary factor leading to these collisions. The perception module generally detected the NPC with fairly accurate positions as the ego vehicle approached. However, the predicted travel paths were often unrealistic, and both the estimated speeds and shapes were underestimated until the ego was very close. For instance, in the U-turn collision case with the ego and NPC speeds of 35 km/h and 10 km/h, respectively, the predicted path **1.7 seconds prior to impact** lacked a U-turn shape, but instead followed a diagonal trajectory crossing multiple lanes and eventually leaving the road. This prediction was only corrected **1 second** later (i.e., 0.7 seconds before the collision). The situation is illustrated in Fig. 6.

This behavior contrasts with the cut-in, cut-out, and de-

celeration scenarios examined in prior work [16], where the NPC traveled in the same direction as the ego vehicle. In those cases, Autoware at least was able to predict the general direction of motion correctly. The oncoming traffic scenarios studied in this work reveal a more fundamental limitation: Autoware failed to recognize the traveling intent of oncoming vehicles until the last moment. A similar issue was observed in the shape estimation: until the final 1–2 seconds before impact, the perceived object shape did not conform to a realistic vehicle shape (box), and the perceived size was smaller than the true vehicle dimensions. Additional illustrations for the swerve scenarios and further discussion are provided in [23].

F. Threats to Validity

1) Internal Validity: The experiments are inherently subject to non-determinism arising from multiple sources, including the simulators, middleware communication between simulators and ADSs, and stochastic components in learning-based models. Such non-determinism may lead to variability across runs. To mitigate this threat, we executed each scenario three times and reported the results.

The collision-avoidability classification depends on the assumptions of the reference driver model, including reaction times and braking capability. In practice, human drivers exhibit variability in these factors, and different parameter choices may shift the boundary between avoidable and unavoidable collisions. In this work, the parameters are chosen to represent a reasonably attentive driver with good braking capability, and so the benchmark should be interpreted as evaluating avoidability under this baseline model rather than capturing the full range of possible human driving behaviors. Exploring the sensitivity of the avoidability boundary to alternative reference models or parameter ranges is left for future work.

Another potential threat arises from the long-running execution of Autoware, AWSIM-Labs, and CARLA, which may occasionally exhibit instability in these systems. To address this, we carefully monitored each run and restarted the simulation whenever abnormal behaviors were observed, ensuring that the reported collisions were not caused by system-level failures.

2) External Validity: All experiments were conducted in simulated environments (AWSIM-Labs and CARLA), which may not fully capture the complexities of real-world driving conditions. In addition, the benchmark focuses on two safety-critical scenario classes, which, while representative and motivated by safety standards, do not cover the full diversity of real-world traffic situations. We emphasize, however, that the underlying methodology is inherently extensible; the same framework can be applied to a broader range of scenario classes by defining the corresponding reference driver models and scenario-specific parameterizations.

Another external validity threat concerns the use of simplified vehicle and behavior models, e.g., fixed swerving trajectories and constant vehicle speeds. We note that these abstractions follow the JAMA's methodology [13], which systematically decomposes the driving condition space to enable manageable scenario-based evaluation. Nevertheless, real-world deviations from these assumptions may lead to different avoidability outcomes, and thus, the results should be interpreted within the scope of the modeled conditions.

VI. Conclusion

This paper presented safety reference benchmarks for safety-critical oncoming traffic scenarios and demonstrated their effectiveness in evaluating ADSs. The benchmarks formally characterize collision avoidability and delineate clear boundaries between collision-avoidable and collision-unavoidable scenarios in the scenario space. Within the scenario set defined by the benchmarks, safety-critical scenarios can be systematically identified, enabling the construction of highly challenging scenarios for safety evaluation and testing ADSs.

Using these benchmarks, extensive simulation-based experiments revealed safety weaknesses not only in end-to-end learning-based agents, but also in a production-grade modular ADS—Autoware—highlighting the need for safety improvement. The experimental results confirm that the benchmarks provide a sound and practical basis for comparative safety evaluation across different ADS implementations. They also yield several valuable insights: a modular, production-grade ADS (Autoware) does not always outperform simpler end-to-end agents (e.g., LAV) in terms of safety; swerve scenarios are generally more challenging than U-turn scenarios; and integrating a controller safety shield can substantially improve Autoware's safety performance.

This study highlights the value of benchmark-driven safety evaluation for systematic ADS comparison and for guiding the development of safer, more reliable ADSs. Although this work focuses on two representative classes of oncoming traffic scenarios, the underlying framework is general and not limited to these specific cases. The process of formalizing traffic participant behaviors, identifying relevant parameters, and applying a reference model to establish avoidability boundaries can likewise be transferred to other scenario categories. Expanding the benchmark to cover a broader set of scenarios is an important direction for future work.

References

[1] S. Tang, Z. Zhang, Y. Zhang, J. Zhou, Y. Guo, S. Liu, S. Guo, Y. Li, L. Ma, Y. Xue, and Y. Liu, "A survey on automated driving system testing: Landscapes and trends," *ACM Trans. Softw. Eng. Methodol.*, vol. 32, no. 5, pp. 124:1–124:62, 2023.

[2] D. Kaufmann, L. Klampfl, F. Klück, M. Zimmermann, and J. Tao, "Critical and challenging scenario generation based on automatic action behavior sequence optimization," in *2021 IEEE International Conference on Artificial Intelligence Testing, AITest 2021*, pp. 118–127, 2021.

[3] X. Zheng, H. Liang, B. Yu, B. Li, S. Wang, and Z. Chen, "Rapid generation of challenging simulation scenarios for autonomous vehicles based on adversarial test," in *2020 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 1166–1172, 2020.

[4] K. Viswanadha, F. Indaheng, J. Wong, E. Kim, E. Kalvan, Y. Pant, D. J. Fremont, and S. A. Seshia, "Addressing the IEEE AV test challenge with scenic and verifai," in *2021 IEEE International Conference on Artificial Intelligence Testing, AITest 2021*, pp. 136–142, 2021.

[5] A. Gambi, M. Müller, and G. Fraser, "Automatically testing self-driving cars with search-based procedural content generation," in *28th ACM SIGSOFT International Symposium on Software Testing and Analysis, ISSTA 2019, China, July 15-19, 2019*, pp. 318–328, ACM, 2019.

[6] Y. Zhou, Y. Sun, Y. Tang, Y. Chen, J. Sun, C. M. Poskitt, Y. Liu, and Z. Yang, "Specification-based autonomous driving system testing," *IEEE Trans. Software Eng.*, vol. 49, no. 6, pp. 3391–3410, 2023.

[7] S. Goyal, A. Griggio, J. Kimblad, and S. Tonetta, "Automatic generation of scenarios for system-level simulation-based verification of autonomous driving systems," in *FMAS@iFM 2023*, vol. 395 of *EPTCS*, pp. 113–129, 2023.

[8] G. Li, Y. Li, S. Jha, T. Tsai, M. B. Sullivan, S. K. S. Hari, Z. Kalbarczyk, and R. K. Iyer, "AV-FUZZER: finding safety violations in autonomous driving systems," in *31st International Symposium on Software Reliability Engineering, Portugal, Oct 12-15, 2020*, pp. 25–36, 2020.

[9] S. Lin, F. Chen, L. Xi, G. Wang, R. Xi, Y. Sun, and H. Zhu, "Tm-fuzzer: fuzzing autonomous driving systems through traffic management," *Autom. Softw. Eng.*, vol. 31, no. 2, 2024.

[10] Q. Jin, T. Wu, Y. Dong, Z. Ding, and Y. Xu, "Reinseed: Reinforcement fuzz testing with multiphase seed optimization for autonomous driving systems," *IET Software*, vol. 2025, no. 1, p. 8657455, 2025.

[11] A. Calò, P. Arcaini, S. Ali, F. Hauer, and F. Ishikawa, "Generating avoidable collision scenarios for testing autonomous driving systems," in *13th IEEE International Conference on Software Testing, Validation and Verification, Portugal, October 24-28, 2020*, pp. 375–386, 2020.

[12] A. Calò, P. Arcaini, S. Ali, F. Hauer, and F. Ishikawa, "Simultaneously searching and solving multiple avoidable collisions for testing autonomous driving systems," in *Genetic and Evolutionary Computation Conference, Mexico, July 8-12, 2020*, pp. 1055–1063, 2020.

[13] Japan Automobile Manufacturers Association, "Automated driving safety evaluation framework ver 4.0," tech. rep., December 2025.

[14] S. Kato et al., "Autoware on board: enabling autonomous vehicles with embedded systems," in *The 9th ACM/IEEE International Conference on Cyber-Physical Systems, ICCPS 2018*, pp. 287–296, 2018.

[15] Autoware Foundation, "AWSIM-Labs." https://github.com/autowarefoundation/AWSIM-Labs. Accessed: 2025-08-30.

[16] D. D. Tran, T. Tomita, and T. Aoki, "Safety analysis of autonomous driving systems: A simulation-based runtime verification approach," *IEEE Transactions on Reliability*, vol. 74, no. 4, pp. 4574–4588, 2025.

[17] H. Shao, L. Wang, R. Chen, H. Li, and Y. Liu, "Safety-enhanced autonomous driving using interpretable sensor fusion transformer," in *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, vol. 205 of *Proceedings of Machine Learning Research*, pp. 726–737, PMLR, 2022.

[18] D. Chen and P. Krähenbühl, "Learning from all vehicles," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, USA, June 18-24, 2022*, pp. 17201–17210, IEEE, 2022.

[19] A. Prakash, K. Chitta, and A. Geiger, "Multi-modal fusion transformer for end-to-end autonomous driving," in *IEEE Conference on Computer Vision and Pattern Recognition, June 19-25, 2021*, pp. 7077–7087, 2021.

[20] K. Chitta, A. Prakash, B. Jaeger, Z. Yu, K. Renz, and A. Geiger, "Transfuser: Imitation with transformer-based sensor fusion for autonomous driving," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 12878–12895, 2023.

[21] A. Dosovitskiy, G. Ros, F. Codevilla, A. M. López, and V. Koltun, "CARLA: an open urban driving simulator," in *1st Annual Conference on Robot Learning, CoRL 2017*, vol. 78, pp. 1–16, 2017.

[22] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, "Safe reinforcement learning via shielding," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), USA, Feb 2-7, 2018*, pp. 2669–2678, AAAI Press, 2018.

[23] Anonymous Authors, "Safety Reference Benchmarks with Avoidability Criteria for Evaluating Autonomous Driving Systems - Supporting Material." https://github.com/dtanony/ADS-Safety-Reference-Benchmark, 2026.

[24] United Nations Economic Commission for Europe (UNECE), "Un regulation no 157 – uniform provisions concerning the approval of vehicles with regards to automated lane keeping systems [2021/389]," Mar 2021.

[25] ISO - International Organization for Standardization, "ISO34502: Road vehicles - Test scenarios for automated driving systems - Scenario based safety evaluation framework," standard, 2022.

[26] Z. Hu, S. Guo, Z. Zhong, and K. Li, "Coverage-based scene fuzzing for virtual autonomous driving testing," *CoRR*, vol. abs/2106.00873, 2021.

[27] Baidu Apollo team (2017), "Apollo: Open Source Autonomous Driving." https://github.com/ApolloAuto/apollo. Accessed: 2025-08-30.

[28] J. Zhang, C. Xu, and B. Li, "Chatscene: Knowledge-enabled safety-critical scenario generation for autonomous vehicles," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 15459–15469, IEEE, 2024.

[29] C. Chang, S. Wang, J. Zhang, J. Ge, and L. Li, "LLMScenario: Large language model driven scenario generation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 11, pp. 6581–6594, 2024.

[30] S. Lin, F. Chen, L. Xi, K. Xie, Y. Zheng, H. Fei, Y. Sun, and H. Zhu, "ScenarioFuzz-LLM: Enhancing diversity in autonomous driving scenario fuzzing with llms," in *28th International Conference on Computer Supported Cooperative Work in Design, France, May 5-7, 2025*, pp. 1581–1586, 2025.

[31] A. Gambi, T. Huynh, and G. Fraser, "Generating effective test cases for self-driving cars from police reports," in *ESEC/SIGSOFT FSE 2019*, pp. 257–267, 2019.

[32] S. K. Bashetty, H. B. Amor, and G. Fainekos, "Deepcrashtest: Turning dashcam videos into virtual crash tests for automated driving systems," in *2020 IEEE International Conference on Robotics and Automation, ICRA 2020*, pp. 11353–11360, 2020.

[33] J. P. Paardekooper et al., "Automatic identification of critical scenarios in a public dataset of 6000 km of public-road driving," in *26th International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, no. 19-0255, 2019.

[34] A. Gambi, V. Nguyen, J. Ahmed, and G. Fraser, "Generating critical driving scenarios from accident sketches," in *IEEE International Conference On Artificial Intelligence Testing, AITest 2022, Newark, CA, USA, August 15-18, 2022*, pp. 95–102, 2022.

[35] Z. Wei, H. Huang, G. Zhang, R. Zhou, X. Luo, S. Li, and H. Zhou, "Interactive critical scenario generation for autonomous vehicles testing based on in-depth crash data using reinforcement learning," *IEEE Trans. Intell. Veh.*, vol. 10, no. 3, pp. 1471–1482, 2025.

[36] G. Zhang, H. Huang, R. Zhou, S. Li, and J. Bian, "High-risk trajectories generation for safety testing of autonomous vehicles based on in-depth crash data," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 8, pp. 11619–11630, 2025.

[37] E. T. Barr, M. Harman, P. McMinn, M. Shahbaz, and S. Yoo, "The oracle problem in software testing: A survey," *IEEE Trans. Software Eng.*, vol. 41, no. 5, pp. 507–525, 2015.

[38] R. C. Coulter, "Implementation of the pure pursuit path tracking algorithm," tech. rep., Carnegie Mellon University, Robotics Institute, 1992.

[39] On-Road Automated Driving (ORAD) Committee, *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, Apr. 2021.

[40] CARLA Team, "CARLA Autonomous Driving Leaderboard." https://leaderboard.carla.org/. Accessed: 2025-10-30.

[41] M. J. Tehrani, J. Kim, and P. Tonella, "PCLA: A framework for testing autonomous agents in the CARLA simulator," in *The 33rd ACM International Conference on the Foundations of Software Engineering, Norway, June 23-28, 2025*, pp. 1040–1044, 2025.

[42] CARLA team, "ScenarioRunner for CARLA." https://github.com/carla-simulator/scenario_runner. Accessed: 2025-09-02.

[43] Y. Tang, Y. Zhou, F. Wu, Y. Liu, J. Sun, W. Huang, and G. Wang, "Route coverage testing for autonomous vehicles via map modeling," in *IEEE International Conference on Robotics and Automation, ICRA 2021*, pp. 11450–11456, 2021.

[44] Y. Tang, Y. Zhou, T. Zhang, F. Wu, Y. Liu, and G. Wang, "Systematic testing of autonomous driving systems using map topology-based scenario classification," in *36th IEEE/ACM International Conference on Automated Software Engineering, ASE 2021*, pp. 1342–1346, 2021.

[45] C. Xia, S. Huang, Y. Yao, C. Zheng, and Y. Wang, "Generating autonomous driving safety violation scenarios based on multi-objective optimization," in *23rd International Conference on Software Quality, Reliability, and Security, Thailand, Oct 22-26, 2023*, pp. 509–515, 2023.

[46] R. Zhou, H. Huang, G. Zhang, H. Zhou, and J. Bian, "Crash-based safety testing of autonomous vehicles: Insights from generating safety-critical scenarios based on in-depth crash data," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 10, pp. 15616–15630, 2025.

[47] R. Zhou, W. Gui, H. Huang, X. Liu, Z. Wei, and J. Bian, "Diffcrash: leveraging denoising diffusion probabilistic models to expand high-risk testing scenarios using in-depth crash data," *Expert Syst. Appl.*, vol. 287, p. 128140, 2025.