Injecting Hallucinations in Autonomous Vehicles: A Component-Agnostic Safety Evaluation Framework

ALEXANDRE MOREIRA NASCIMENTO[1,*], GABRIEL KENJI GODOY SHIMANUKI[1,*], LÚCIO FLAVIO VISMARI[1], JOÃO BATISTA CAMARGO JR.[1], JORGE RADY DE ALMEIDA JR.[1], PAULO SERGIO CUGNASCA[1], ANNA CAROLINA MULLER QUEIROZ[2], JEREMY NOAH BAILENSON[3]

[1]Escola Politécnica da Universidade de São Paulo, São Paulo 05508-010, Brazil
[2]Department of Communication, University of Miami, Coral Gables, FL 33124 USA
[3]Department of Communication, Stanford University, Stanford, CA 94305-2050 USA
[*]Authors contributed equally to this work

Corresponding author: Alexandre Moreira Nascimento (e-mail: alexandremoreiranascimento@alum.mit.edu).

ABSTRACT Perception system failures in autonomous vehicles (AV) remain a critical safety concern because they are the basis for many accidents. To understand how such failures compromise safety, researchers commonly inject artificial faults into hardware or software components and observe the effects. Existing fault injection (FI) studies, however, typically focus on a single sensor or a specific machine perception (MP) module, yielding siloed, non-interoperable frameworks that are difficult to integrate into a unified simulation environment. We address this limitation by reframing perception failures as hallucinations, which are false perceptions that corrupt situational awareness of an AV and may lead to hazardous control actions. Because hallucinations capture only the observable consequences of failure, this perspective elevates the analysis to a higher level of abstraction, allowing us to ignore the idiosyncrasies of individual sensors, algorithms, or hardware modules. Instead, we can focus on how their faults manifest in the perception pipeline. Building on this paradigm, we introduce a configurable component-agnostic hallucination injection (HI) framework that induces six plausible hallucination types in an interactive, high-fidelity, open-source simulation environment. More than 18,350 simulations were executed in which hallucinations were injected while the AVs crossed an unsignalized transverse street with traffic. The resulting data were used to (i) statistically validate the framework and (ii) quantify the impact of each type of hallucination on accidents and near misses. The experiments demonstrate that certain hallucinations, such as perceptual latency and drift, significantly increase the risk of collision in the scenario tested, validating the proposed paradigm can stress the AV system safety. The framework offers a scalable, statistically validated, component agnostic, and fully interoperable toolset that simplifies and accelerates AV safety validations, even those with novel MP architectures and components. It can potentially reduce the time-to-market of AV and lay the foundation for future research on fault tolerance, and resilient AV design.

INDEX TERMS Autonomous Vehicle, Fault Injection, Hallucination, Machine Perception, Perception Systems, Safety, Simulation, Testing, Validation

I. INTRODUCTION

THE integration of artificial intelligence (AI) into autonomous vehicles (AVs) introduces unique challenges to safety assurance. Evaluating the safety of AI-based components is difficult because commonly used metrics emphasize overall capabilities while masking specific failure modes and their consequences [1]. These limitations are especially critical in machine perception (MP) systems, which rely heavily on AI techniques and directly influence safety-critical decisions [2], [3]. MP interprets the driving environment and provides the situational awareness required for AV motion planning and control. Failures in this process can cause an AV to take unsafe actions with severe consequences, as illustrated by incidents such as the 2018 Uber pedestrian fatality [4] and the 2019 Tesla Autopilot (autonomous) truck collision [5], along with other crashes and recalls [6], [7]. These examples

arXiv:2510.07749v1 [cs.RO] 9 Oct 2025

highlight the critical need for robust safety assurance in MP systems, especially since fallback mechanisms such as human intervention often suffer from delayed reactions and vigilance degradation [8], [9].

A persistent challenge in AV safety is the divide between the AI research community and the safety engineering community [10]. Advances in deep learning have enabled state-of-the-art perception, navigation, and control [11], but these advances are often driven by benchmark performance rather than system-level safety objectives [12]. Several researchers have noted that this AI-centric mindset emphasizes local robustness and narrow metrics while neglecting application-level safety assurance [13]–[17]. Even adversarial robustness research tends to isolate specific vulnerabilities, such as perturbations of images and videos [18]–[21], control flaws [22], [23], or navigation corner cases [24]–[26], without modeling how these local faults propagate through the AV stack. This gap shows why robustness is important at the system level, meaning the ability to keep operating correctly even when subsystems or components have faults [27], [28].

Robustness is therefore a key attribute for safe AVs. Koopman [29] highlights robustness testing for AI-based systems and advocates fault injection (FI) to evaluate performance under rare or unexpected conditions. More broadly, FI is a well-established dependability assessment technique in which faults are deliberately introduced into hardware or software systems, real or simulated, to reveal vulnerabilities and evaluate safety attributes [30]. FI supports both fault removal and fault forecasting, the core goals of dependability validation [31]. Classical FI approaches [32], [33] focused on physical hardware or software components to expose vulnerabilities, test redundancy mechanisms, and evaluate performance under stress conditions [34]. However, for meaningful results, the injected faults must reflect realistic system behavior [31].

Simulation-based FI offers a practical and scalable approach to evaluate AV design and safety. By realistically modeling both vehicle components and their surrounding environments, engineers can introduce controlled anomalies into specific modules, such as perception sensors or system interfaces, and systematically observe the resulting system behavior under repeatable conditions [30].

Despite advances in machine learning (ML) robustness research [35], [36], few studies have applied FI directly to AV MP systems. To address this gap, this study introduces a configurable and component-agnostic hallucination injection (HI) framework that simulates six distinct types of perception output anomalies, called hallucinations. Although the term lacks a universal definition, it has been widely associated with false but plausible outputs generated by large language and generative AI models [37]. Analogously, in AV perception, a hallucination represents a misleading observation that the MP interprets as real. As an example of perceptual distortion (Figure 1), a rear camera captured an unexpected obstruction, such as an insect landing on the lens. The MP software reconstructed the 360°view with a giant fly seemingly blocking

the environment, producing a hallucination that misrepresents reality and could mislead AV motion planning. This example highlights the importance of testing AVs against perceptual anomalies that may arise from real-world interactions, beyond traditional fault models. HI abstracts traditional *fault → error → failure* modeling [38], focusing solely on perceptual anomalies to systematically evaluate how AVs respond to corrupted situational awareness. This enables standardized, component-agnostic testing across diverse AV architectures, regardless of the specific sensors or AI modules involved. The framework is integrated into an open-source traffic simulator, supporting interactive real-time testing at scale. We demonstrated its statistical validity through more than $19,000$ simulations in an unsignalized intersection scenario, which revealed vulnerabilities that can be missed by conventional testing and offered new perspectives on AV safety under hallucinations.
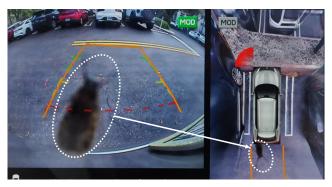

FIGURE 1. Hallucination produced by a 360°image reconstruction algorithm

The remainder of this paper is organized as follows: Section II reviews related work on FI in autonomous systems and sensor-agnostic faults in AV MP systems. Section III presents the design of the proposed HI framework, details the modeling of perception hallucination types, and explains the systematic evaluation methodology. Section IV reports the experimental results. Section V discusses the implications for AV safety assessment, and Section VI concludes with final remarks.

II. THEORETICAL BACKGROUND
A. AUTONOMOUS SYSTEMS FI FRAMEWORKS
FI has long been established as a powerful technique to evaluate the dependability of computing systems by deliberately introducing faults and observing system behavior under fault conditions [39]. Early FI research focused on low-level abstraction layers in hardware, including pin-level injection (RIFLE [33]), chip-level testing under radiation (FIST [40]), and fault modeling via simulation (FOCUS [41]). In parallel, software-level FI frameworks such as FERRARI [42], FTAPE [43], FIAT [32], Xception [44], DOCTOR [45], EXFI [46], and GOOFI [47] targeted CPUs, memory, and I/O subsystems, injecting faults using traps or event triggers. These tools enabled robustness testing, redundancy evaluation, and early

fault diagnosis with minimal hardware overhead, particularly in safety-critical domains like aerospace, embedded systems, and robotics [48], [49]. However, these approaches relied on the assumption that faults could be accurately represented and injected at the level of individual components, which restricted them to physical or deterministic failure modes.

Building on these foundations, later research has begun adapting FI for AVs, which present new challenges due to their AI-driven, highly integrated architectures. FI at the system-level has gained traction as a way to validate the robustness of AV in rare but safety-critical scenarios [29]. Frameworks such as AVFI [50] and Kayotee [51] simulate sensor failures and inject perturbations into the input of neural network to study their downstream effects on control and decision-making. Multi-agent platforms, such as presented in [52], model agent interactions, and fault-aware behaviors in dynamic environments. However, these approaches remain tied to specific modules or components and fall short in capturing the breadth of consequences such failures can have at the perception abstraction level, which limits their applicability for extensive safety validation.

Another group of studies has focused on improving MP robustness through adversarial testing at the module-level. Benchmarks have explored structured corruptions in vision models [53], [54], and tools such as AV-Fuzzer [55] and ontology-based generators [56] have been used to create dangerous edge-case scenarios. Other works target vulnerabilities in internal mechanisms of ML systems by injecting faults into neural network parameters [57]–[60] or exploring neuron-level behaviors through coverage criteria and logic inconsistencies [61]–[63]. However, these studies operate at low abstraction levels of ML, typically isolated from the entire AV stack, and offer little support to understand how such faults manifest as hazardous behaviors when propagated through perception, planning, and controlling pipelines.

Recent research efforts have aimed to close this gap by focusing on perception-level failures. However, they often remain tightly coupled to specific sensors or data fusion strategies. For example, FADE [64] injects faults directly into camera and LiDAR data to model real-world sensor degradations. PEM [65] proposes error models to abstract failure perceptions but derives these from predefined sensor setups, which limits their generalization. HydraFusion [66] improves robustness by dynamically adjusting the fusion based on environmental context. Jin *et al.* [67] focus on real-time fault detection and isolation by combining hardware and analytical redundancy. In contrast, Hou *et al.* [68] aim to classify the source of sensor faults in real-time, to allow diagnosis and remediation. Despite their contributions, these studies primarily address the origin of faults (e.g., sensor degradation, network faults, or fusion inconsistencies) rather than their observable consequences on situational awareness and decision-making.

B. FAULTS IN AV MP SYSTEMS

MP provide situation awareness to AV systems. They interpret sensor data to detect and classify critical elements in the environment (vehicles, pedestrians, signals), providing the spatial and semantic information needed for downstream decision-making. Figure 2 illustrates a typical MP system, showing its main components and an abstraction of its inputs and outputs. Although a common setup includes a camera, GPU, and ML model, perception pipelines can integrate other sensors, such as LiDAR, radar, and others [69].
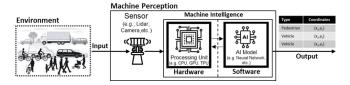

FIGURE 2. MP system model

Regardless of the sensor modality, MP remains vulnerable to faults that can compromise safety [2]. Faults are classified by their persistence (Figure 3) [38]. Persistence categories include **permanent faults**, which persist until the faulty component is repaired or replaced, for example, a processor damaged by overheating or corrupted firmware causing consistent malfunction [38]. They also include **transient faults**, which are temporary and hard to reproduce, such as a bit flip in sensor data due to electromagnetic interference [38]. In addition, **intermittent faults** [70] occur at irregular intervals while the system otherwise operates normally, such as an unstable LiDAR connection that fails under specific thermal conditions. These faults can affect MP, machine control (MC, broadly responsible for motion planning and decision-making), or machine actuator (MA, responsible for executing control commands), ultimately threatening overall vehicle safety.

Faults can also be classified by their dimension [38]), such as hardware or software (Figure 3). Hardware faults can include camera calibration drift and damage to the charge-coupled device (CCD). For example, damage to a CCD can lead to partial image corruption (Figure 4 - left) [71] or complete frame loss (Figure 4 - right) [71]. Both can compromise scene classification [72]. Beyond these examples, there is a list of other dimension fault that can jeopardize the utility of CCD sensors or even LiDARs [73], such as black pixels [74], commonly caused by contamination of sensor material, and traps [74], dark columns produced by a transfer charge barrier.

C. HALLUCINATIONS

According to Avizienis *et al.* [38], a failure occurs when the service delivered by a system deviates from its intended correct service because of an existing fault. Therefore, a faulty component of an MP system may cause an MP failure, which may cause perceptual distortion. In other words, an MP failure may cause a perceptual experience that occurs without the corresponding external stimuli. Unfortunately, in
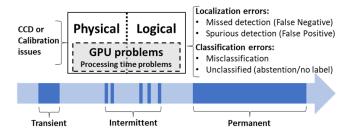
FIGURE 3. Taxonomy of dimensions and persistence of MP faults

FIGURE 4. Illustration of observable consequences of CCD partial fault (left) and complete fault (right)

an AV system, those MP failures can propagate to the MC since they can be perceived with the same force and clarity as genuine perceptions. Consequently, the MC will adapt and react according to the distorted situation awareness, which may result in a failure of the AV that compromises its safety.

From a psychological perspective, perceptual experiences that occur without corresponding external stimuli and are perceived with the same force and clarity as genuine perceptions are known as hallucinations [75]. They can manifest across sensory modalities, including auditory, visual, tactile, olfactory, and gustatory domains, with auditory hallucinations particularly associated with psychotic disorders such as schizophrenia [75], [76]. Hallucinations can alter an individual's perception of reality, affecting their situational awareness, which in turn may influence behavior and reactions [75], [76].

Inspired by the psychological domain, here we define MP hallucinations as MP failures that manifest as perceptual experiences that occur without corresponding external stimuli and are perceived with the same force and clarity as genuine perceptions. Since hallucinated output from MP systems can mislead motion planning and control, they represent a special set of AV subsystem faults that can potentially be a serious source of risk to AV safety, while conventional fault models often overlook. Thus, this definition captures an adequate abstraction level and scope of interest for a holistic and system-level safety evaluation of AVs when MP failures are present, regardless of which component is faulty and which fault mechanism is involved.

However, this novel concept requires a framework for injecting hallucinations into perception rather than focusing narrowly on component-level faults, which do not exist to the best of our knowledge. Therefore, in an effort to bridge literature gaps and introduce the proposed novel perspective of analysis, the present work adopts a component-agnostic approach, focusing on injecting and exploring the effects of

hallucinations into the AV MP, rather than explicitly modeling fault sources. This abstraction enables faults to be injected at the level of behavioral symptoms, allowing for unified evaluation across AV platforms regardless of technical architecture, components, or sensor configuration. By decoupling failure modeling from implementation details, this approach facilitates standardized and interoperable safety testing in simulations, offering new insights into the propagation and emergent system behavior resulting from MP failures. Moreover, it can accelerate AV safety research by supporting anticipated AV evaluations against novel and potential hallucination types prior to research that uncovers their root causes and fault mechanisms.

This shift from fault causation to analyzing observable effects addresses a broader need for a system-level safety assessment that extends beyond individual modules. Rather than focusing on the failure of the component, the perspective emphasizes how the system behaves when perception is degraded. By representing failures as parameterizable hallucinations, the approach enables reproducibility and quantitative evaluation in diverse scenarios, supporting a more systematic and generalizable understanding of AV safety. HI thus serves as a unifying abstraction, allowing safety analysis to reflect how AVs behave under perception anomalies, regardless of where or why those anomalies arise. This abstraction is particularly valuable for studying emergent behaviors and identifying systemic vulnerabilities that traditional FI or adversarial methods may overlook.

D. TYPES OF AV SYSTEMS HALLUCINATIONS
Although the concept of hallucinations in the AV domain was introduced in this study to raise the level of abstraction in the analysis, avoiding the need to deal with fault mechanisms and models at the component level, it is essential to note that plausible hallucinations are those supported by already uncovered or at least plausible failures or potentially underlying MP components' fault mechanisms. Therefore, for the purpose of plausibility, an explanation of the underlying fault mechanism or component involved in each hallucination type defined ahead is presented.

Hallucinations can occur across different sensing modalities, including cameras, LiDAR, radar, or sensor fusion systems, and may arise from hardware, software, or environmental interactions. For example, calibration drift, the gradual deviation of sensor outputs over time, can result from environmental factors such as changes in temperature or mechanical shock [77], [78] , shifting coordinate frames and degrading object location (Figure 5). Such hallucinations have the potential to alter the sensor reference system (O, x, y, z to O, x', y', z'), leading to a spatial misinterpretation (Figure 5).

Specific hallucinations such as recognition latency, missed detections, false classifications, and spatial drift can emerge from various causes at different system-levels [38] (Figure 6). For example, **Perception Linear Drift** and **Perception Angular Drift** can cause spatial location errors, misplacing detected objects and affecting subsequent reasoning and plan-
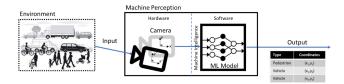
FIGURE 5. Camera calibration drift

ning [79]. Class confusion, such as misidentifying a pedestrian as a cyclist/vehicle and spurious detections, can manifest itself as **Phantom Perception** (false positives), where the system detects or classifies objects that are not present or mislabels them, compromising situational awareness and potentially leading to unsafe responses [80]. **Missed Detections** (false negatives) may result from poor generalization in ML models, degraded input quality, or synchronization errors between system components [80], [81]. In contrast, a **Blind Region** refers to an entire spatial area that the perception system fails to observe, rather than a single missed entity, such as when sensor mounting or field-of-view limitations (scene occlusion) systematically obscure parts of the environment. **Perception Latency** can result from overloaded processing units (including GPUs prone to scheduling delays, memory bottlenecks, or load spikes [82]–[84]), inefficient software pipelines, or timing problems in sensor data transmission [85]. **Unsigned**, such as temporal instability and adversarial vulnerabilities further illustrate how hallucinations span hardware and software layers [72], [86].

III. METHODOLOGY

Injecting hallucinations into safety critical systems like an AV in real situations to evaluate its reactions and safety implications would be too risky. However, other scientific fields have successfully studied the behavioral consequences of perceptual modifications in a less risky setup, which provided inspiration and guidance for the present investigation. In fact, psychological and behavioral research have been using virtual environments (VEs) to enable the systematic manipulation of a user's sensory reality. Researchers can alter visual, auditory, and haptic feedback to create perceptual experiences that would be dangerous, counterintuitive, expensive, or impossible to control in the physical world [87]. This VE capability enables direct investigation of how specific perceptual modifications affect human cognition, emotion, and behavior [88]. An illustrative example is the embodiment of avatars with visually altered features, such as an arm appearing as stone, made participants feel heavier and stiffer, move more slowly, and exhibit changes in motor cortical excitability [89]. This level of experimental control over a person's perception of self and the environment provides a unique and encouraging tool for exploring the causal links between perception and behavior.

Driving simulators represent a specialized class of VEs that extend these principles to transportation research. Such platforms allow researchers to study driver perception, emo-

FIGURE 6. Effect of distinct MP hallucinations on the AV situational awareness

tion, and decision-making under controlled and repeatable conditions. Previous studies have investigated how onboard voice interfaces influence driver affect and safety outcomes [90], [91]. Others have modeled accident risk based on visual cues, such as facial expressions [92]. Additional work has analyzed behavioral adaptation in the presence of partially autonomous driving systems [93], [94]. These examples sug-

gest that simulators are useful tools for investigating how a driver's perception and behavior are altered by advanced vehicle systems.

It is noteworthy that a critical factor for the success of such studies is the plausibility of the virtual scenario. A plausible illusion is the cognitive interpretation that events occurring in the VE are real, not in the sense of being photorealistic, but in the sense that they are happening and are coherent with the rules of the environment [95]. For a participant to react authentically to a perceptual modification, they must accept the scenario as a credible sequence of events, even if it is fantastical. Without plausibility, the user may disengage or react based on the artificiality of the setup rather than the intended stimulus. The plausibility is something the user is continuously evaluating unconsciously while in a VE.
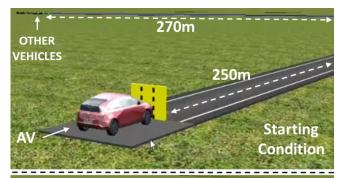
In the present study, inspired by the analogous investigations in the human sciences, a VE was used to validate a proposed HI module. Environmental plausibility in VE is crucial for reliably assessing the validity of the HI module. However, emulating in AVs the mechanism that enables humans to consistently assess environmental plausibility is a challenging task. Thus, rather than expecting the AV to be able to evaluate the environment plausibility, this was considered a VE requirement by design. All the other information regarding experimental design choices, methods, protocols, and materials used to support the present study is presented in the following subsections.

A. EXPERIMENTAL USE CASE

A simulated intersection scenario (Figure 7) was used as a testbed due to its high accident potential and relevance to the evaluation of safety research [96]. Intersections are critical points in the road network, accounting for about a quarter of traffic fatalities and nearly half of all injuries in the U.S. [97]. Their complexity makes them relevant for testing fault tolerance in AVs.

In the simulated scenario, the AV needed to cross a one-way street intersected by five other vehicles that have the right of way. These vehicles start approximately 270 meters from the intersection and accelerate from rest to a maximum speed of 54 km/h. The AV starts 250 meters from the intersection and must identify a safe time window, called the candidate window (CW), to cross without violating traffic rules. CW are defined as a predicted time-space gap in traffic during which the AV can safely cross the intersection computed by the control.

Each simulation execution ends when the AV either (1) crosses safely, (2) causes a collision, or (3) halts indefinitely due to the absence of safe opportunities. Any simulation error, such as a crash in the simulator or runtime failure, was treated as an invalid execution, and the corresponding log was discarded. This procedure ensured reproducibility by including only fully valid executions in the analysis. The simulations were repeated until the configured number of valid executions was reached.

FIGURE 7. Snapshots of starting condition (top) and execution of simulation (bottom)

B. SIMULATION ENVIRONMENT

All simulations were conducted using the real-time component of the USP54 framework [98], a simulation environment previously used in AV safety research [98]–[100] to evaluate complex traffic scenarios (Figure 8) that supports the plausibility requirement. To support the objectives of this study, additional implementations were developed to extend the framework and enable HI experiments. The USP54 framework integrates VEINS [101], SUMO [102], OMNET++ [103], and OpenDS [104], providing real-time and accelerated simulations with vehicle-to-everything (V2X) communication. These tools were selected because they are well known [105], widely adopted [105], open source, and validated in previous studies. Consequently, the implementations in this work extended the environment to support a newer and wider scope of future investigations.

FIGURE 8. Simulator console

Figure 9 presents the feature of real-time simulation (ReTS) of USP54. Traffic scenarios were simulated in OpenDS, which handled environmental elements, vehicle behavior, and perception modules. The MP system and actuators were also implemented in OpenDS. The control algorithms were executed in MATLAB, which received OpenDS real-time updates (matrix D_{RT}), which contains the position coordinates of e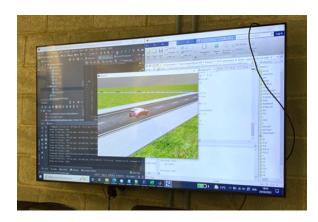ach vehicle and the returned control commands (matrix U_{AV}), which specifies the AV's throttle, brake, and steering wheel angle. Communication between OpenDS and MATLAB was handled via a socket interface.
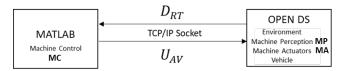

FIGURE 9. System components distribution and integration over OpenDS and Matlab

C. HALLUCINATION INJECTION MODULE

To support this study, the proposed HI module was implemented (Figure 10). This module allows simulation of various hallucination scenarios through configurable properties described in Table 1. The HI module was implemented in MATLAB. The following subsections detail each HI property implemented as an HI variable.

TABLE 1. HI module's properties

Property	Variable	Description	Type*
Active	*ModuleActivation*	Setting the HI module to ON activates the injection of the configured signal, while setting it to OFF deactivates it	C
Type	*HallucinationType*	Type of hallucination injected into the AV during a specific simulation execution	C
Domain	*AffectedDomain*	Affected domain by the hallucination injected during a specific simulation execution	C
Configuration	*HallucinationConfiguration*	Configuration parameters of the hallucination injected into the AV during a specific simulation execution	C
Probability	*HallucinationProbability*	Probability of the hallucination occurrence during a specific simulation execution	N
Persistence	*HallucinationPersistence*	Hallucination Persistence configured for a specific simulation execution	C

* Type: C = Categoric, N = Numeric

The *ModuleActivation* variable is a categorical indicator that specifies whether a hallucination is injected during a given simulation run. It takes the values *ON* or *OFF*, allowing for straightforward comparison between simulations with and without hallucination injection.

The categorical variable *HallucinationType* represents six types of representative hallucinations that can be injected into
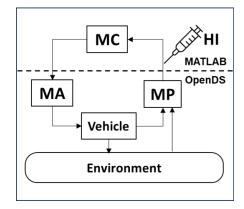

FIGURE 10. Updated USP54 ReTS simulation environment with HI module

the MP system. Table 2 lists all six plausible hallucinations implemented. They are illustrated in Figure 11.

TABLE 2. *HallucinationType* categories

Id	Category	Description
LinDrift	*PerceptionLinearDrift*	Damages the accuracy of the detect vehicle position
Phant	*PhantomPerception*	Non-existent vehicles are detected (false positives)
Missed	*MissedDetection*	Existing vehicles are not detected (false negatives)
AngDrift	*PerceptionAngularDrift*	Angular deviation introduced in the detected vehicle position
Blind	*BlindRegion*	Occluded regions in field-of-view that may obstruct object detection
Latency	*PerceptionLatency*	Delays are introduced into the perception mechanisms, leading to delayed situational awareness sent to MC

The *PerceptionLinearDrift* hallucination induces inaccuracies in the position of recognized objects, potentially causing inappropriate vehicle behavior like sudden braking or oversight of hazards [2], [106]–[108].The *PhantomPerception* hallucination introduces fictitious objects into the output of MP, prompting reactions to non-existent entities, which can lead to unnecessary or dangerous maneuvers. By understanding how an AV control system reacts to these injected false positives, researchers can better calibrate detection algorithms to discriminate between genuine and spurious stimuli, ultimately enhancing the reliability and safety of AV navigation. This tests the system's ability to reject spurious detections. The *MissedDetection* hallucination suppresses data from actual vehicles, simulating missed detections that risk unsafe decisions due to incorrect assumptions of a clear path [109]. The *PerceptionAngularDrift* hallucination simulates angular miscalibration by applying a coordinate transformation to detected object centroids based on a configured angular offset. The MP system processes misaligned data without mathematical compensation, potentially degrading
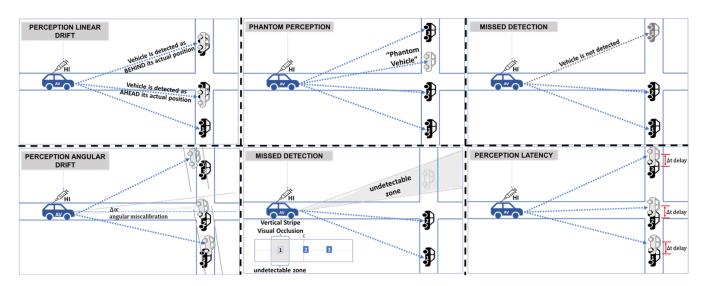
FIGURE 11. HI framework's hallucination types

decision quality [108]. As a result, compromised information forwarded to the MC system may precipitate incorrect assessments and subsequent decision-making, increasing the potential for safety-related risks in AV operations. *BlindRegion* hallucination simulates sensor noise artifacts that obscure parts of the image, leading to degraded object detection. Previous work shows that it can significantly affect ANN-based perception [110], [111], increasing the risk of false positives or negatives. By introducing structured occlusions into simulated perception inputs, researchers can assess the resilience of AV perception systems and develop strategies to mitigate the impact of such image quality problems, thereby supporting safer navigation and decision-making by AVs. The *PerceptionLatency* hallucination simulates latency in the MP system by freezing detection data for a configurable number of cycles, during which new sensor inputs are buffered. After this period, the buffered data are sequentially replayed, emulating the AV's perception of a delayed environment. This attempts to mirror real-world processing or transmission delays that impair real-time decision-making, potentially leading to outdated situational awareness and an increased risk of navigation errors [112]. Previous work has addressed such delays through edge computing to reduce latency [113], whereas [85] emphasizes the need for efficient architectures to ensure safety in dynamic conditions. By implementing *PerceptionLatency*, this study enables controlled analysis of timing anomalies in autonomous systems.

The *AffectedDomain* (Table 3) variable is categorical and indicates which perception domain is targeted by a hallucination introduced in a simulation run [38]. It enables safety analysis per domain. The *ObjectPosition* category refers to hallucinations that affect the precise localization of objects, including *PerceptionLinearDrift* and *PerceptionAngularDrift*. The *ObjectRecognition* category encompass to hallucinations impacting the ability to identify objects, such as *PhantomPerception*, *MissedDetection*, and *BlindRegion*. The *Informa-*

tionTiming category refer to hallucinations affecting the timing of the situational awareness, such as *PerceptionLatency*. It should be noted that no unsigned cases of hallucinations were implemented in the current version of the HI framework. Although they represent a very important type of hallucination, they are going to be implemented in future work since those hallucinations are consequences of cyberattacks and belong to another well-explored and complex field of domain, which is not the scope of the present research.

TABLE 3. *AffectedDomain* categories

Id	Category	Affected Domain
Pos	*ObjectPosition*	Position (coordinates) of detected object
Rec	*ObjectRecognition*	Object detection
Time	*InformationTiming*	Information timing

The categorical variable *HallucinationConfiguration* (Table 4) encompasses the different possible configuration parameters of hallucination injected during a simulation run. Different configurations are associated for each *Hallucination Type*. When no additional configuration is required by the *Hallucination Type*, none is assigned as its value. *PerceptionLinearDrift* is uniquely identified by the *Location* category. Hallucination types *MissedDetection* and *PhantomPerception* require specifying which car is affected or used for phantom perception. The id *x*, corresponds to cars 1, 2 and 3, respectively. For *PerceptionAngularDrift*, the configuration defines the camera rotation angle relative to the longitudinal axis of the AV, with standard values of 5°, 10°, 20°, and 25°, either to the left (L) or to the right (R), resulting in categories such as *ANG05L* (5°rotation to the left) and *ANG25R* (25°rotation to the right). Similarly, the *BlindRegion* hallucination is configured by the angle in an angular coordinate system centered on the AV, corresponding to the

position where the visual occlusion artifact appears. In this work, values of 40°, 50°, and 60°to the left are used, producing categories such as *BLIND50L* (occlusion at 50°to the left). The occlusion stripes are restricted to the left side because the vehicles approach from that direction, as illustrated in the Blind part of Figure 11. In the case of *PerceptionLatency*, the configuration corresponds to the number of cycles that the information is delayed in the AV system. Two standardized values were used: 20 and 40 execution cycles. They were denoted by *LAT20* and *LAT40* respectively. These predefined configurations reduce the variable dimensionality and facilitate pairwise analyses and framework validation. It should be noted that future studies will expand the scope of possible configurations following the current work that focuses on HI framework validation.

TABLE 4. *HallucinationConfiguration* categories

ID & Category	Description
Location	A systemic linear drift affecting the perceived position of all detected vehicle
Car1, Car2, Car3	Selects which crossing car whose perceived attributes will be altered in the AV MP
Ang05L, Ang10L, Ang20L, Ang25L, Ang05R, Ang10R, Ang20R, Ang25R	Sets the camera miscalibration rotating 5°, 10°, 20°, 25°to the left (L) or right (R) side of the AV, considering 0°the axle parallel to the AV trajectory
Blind40L, Blind50L, Blind60L	Position the center of the occlusion stripe at 40°, 50°, 60°to the left side of the AV in the direction of its trajectory. 0°is parallel to the AV trajectory
Lat20, Lat40	Selects the delay to be introduced between the simulation status and the MP in number of simulation cycles (20 or 40)

The *HallucinationProbability* variable defines the probability that a hallucination occurs during a given simulation run, with values ranging from 0 to 1. In the simulations conducted for this study, the probabilities of 1%, 5%, 10%, 25%, and 50% were used to represent a broad spectrum of hallucination probabilities, ranging from rare to frequent events.

The *HallucinationPersistence* variable represents the hallucination manifestation mode during each simulation run and is categorical. When no hallucinations are injected, the simulation is categorized as *Baseline*. If the hallucination is intermittent, the category *Intermittent* is used, whereas a permanent injected hallucination is denoted by *Permanent*.

D. MACHINE PERCEPTION MODULE

To achieve the goals of this study, it was not necessary to replicate the complexity of physical sensors or to implement AI-based MP models. Rather than that, it focused on establishing a controlled and transparent representation of the MP for safety analysis. To this end, we adopted the MP

module developed in [99], using a bypass strategy in which the MP module directly accessed ground truth data from the simulation environment, including the real-time positions and velocities of all vehicles. In its default configuration, this perception sensor has an unlimited range and 360°coverage, effectively providing the complete state of the environment. To reproduce the directional constraints of real perception systems, this information was subsequently filtered through a virtual field of view (FOV) abstraction (Figure 12). The resulting perception layer produced structured outputs equivalent to those illustrated in Figure 12, allowing systematic HI and controlled experimentation. By abstracting the complexities of sensor noise and uncertainty, a reliable baseline was established to support the safety implications of hallucinations. At the same time, it preserved a foundation for integrating more sophisticated MP models in future research.

FIGURE 12. Implementation of the virtual camera FOV in USP54 ReTS from [99]

E. MACHINE CONTROL MODULE

A simplified MC module was implemented to support the current study. Basically, it works by receiving data from the MP to obtain situational awareness and continuously estimate the future occupancy of the intersection by other vehicles and identify CWs, during which the AV can cross safely (Figure 7). This approach mirrors the classical "gap acceptance" concept in traffic engineering, where drivers decide whether to accept a temporal gap before performing maneuvers [114]. Each CW represents a predicted opportunity when no other vehicle is expected to enter the intersection. Thus, MC was implemented based on the paradigms of autonomous intersection management, in which vehicles exploit minimal-time windows to achieve efficient and coordinated intersection passage [115].

For each CW, the MC assesses whether the AV can reach its center without violating the speed limits. Infeasible CWs are discarded. Among the feasible ones, the closest valid window is selected and the longitudinal control (throttle α_{th} and brake α_{br}) is adjusted to match the time of arrival of the AV at the intersection with the CW time. This MC strategy aligns with reservation-based trajectory optimization methods, which plan conflict-free intersection entries by reserving spatio-temporal resources [116]. This MC method was chosen because it provides higher interpretability and predictability than black-box models. That reduces the uncertainty for the analysis and validation of the HI module.

F. RESEARCH QUESTIONS AND INVESTIGATED HYPOTHESES

A set of research questions (RQs) was formulated to evaluate the HI module and quantify its implications for the safety of AV. Each RQ was mapped to a corresponding hypothesis. Each of those hypotheses were unfolded into fine-graned testable hypotheses. For clarity and ease of reference, the list of hypotheses discussed below are consolidated in Table 5

TABLE 5. Summary of granular hypotheses

Hypothesis	Independent Variable	Expected Influence	Dependent Variable
$H_{1.1}$	*ModuleActivation*	↑↓	
$H_{2.1}$	*HallucinationType*	↑↓	
$H_{3.1}$	*AffectedDomain*	↑↓	*Accident*
$H_{4.1}$	*HallucinationConfiguration*	↑↓	*Probability*
$H_{5.1}$	*HallucinationProbability*	↑	
$H_{6.1}$	*HallucinationPersistence*	↑↓	
$H_{1.2}$	*ModuleActivation*	↑↓	
$H_{2.2}$	*HallucinationType*	↑↓	
$H_{3.2}$	*AffectedDomain*	↑↓	*Minimum*
$H_{4.2}$	*HallucinationConfiguration*	↑↓	*Distance*
$H_{5.2}$	*HallucinationProbability*	↓	
$H_{6.2}$	*HallucinationPersistence*	↑↓	

Note: An up arrow (↑) indicates the independent variable is expected to increase the dependent variable. A down arrow (↓) indicates the value is expected to decrease (e.g., a smaller minimum distance). A double arrow (↑↓) indicates that the direction of the influence (increase or decrease) is not known a priori.

RQ1) Do hallucinations increase safety risk?

The first hypothesis (H_1) states that *hallucinations increase the safety risk*. This reflects the primary assumption that the introduction of hallucinations increases the likelihood of accidents compared to baseline condition. Despite the diversity of metrics [117] in the literature to operationalize safety risk, we used two proxy variables: (i) accident and (ii) minimum distance to the nearest vehicle that crosses the intersection. An accident (collision) provides an unambiguous oracle for critical safety failure, whereas the minimum distance is widely used as a continuous risk proxy to assess near-miss scenarios [26], [118]–[121]. These metrics are also adopted in the fitness functions used in search-based testing frameworks to systematically uncover safety-critical scenarios [26], [118], [121]. By decomposing safety into quantifiable, testable components, the proposed methodology aligns with the literature for a rigorous, evidence-based safety evaluation of AVs in diverse operational scenarios [122], [123]. Thus, H_1 was decomposed into:

- **$H_{1.1}$** The injected hallucinations influences the likelihood of an accident occurring;
- **$H_{1.2}$** The injected hallucinations influences the minimum distance between the AV and the nearest vehicle.

RQ2) How do distinct types of hallucinations impact the system safety?

The second hypothesis (H_2) asserts that *the type of hallucination significantly influence the safety risk*. This investigates whether different types of hallucination produce statistically distinguishable results. Previous work argues that ignoring failure distinctions can mask critical risk variations and compromise system safety [124]. Thus, H_2 was derived as:

- **$H_{2.1}$**: The type of hallucination influences the likelihood of an accident occurring;
- **$H_{2.2}$**: The type of hallucination influences the minimum distance between the AV and the nearest vehicle.

RQ3) Do hallucinations targeting distinct perception domains affect system safety differently?

MP spans multiple domains, including object position, object recognition, and information timing. Each supports distinct safety-critical functions. The third hypothesis (H_3) holds that *changes in the perception domains targeted by hallucinations affect system safety differently*. For example, object position hallucinations distort trajectories, while object recognition hallucinations can suppress yielding behavior. H_3 was decomposed into the following components:

- **$H_{3.1}$**: The perception domain targeted by hallucinations influences the likelihood of an accident occurring;
- **$H_{3.2}$**: The perception domain targeted by hallucinations influences the minimum distance between the AV and the nearest vehicle.

RQ4) How do distinct configuration of hallucination influence system safety?

The HI module was designed to parameterize hallucinations by type and configuration. Although H_2 focused on the type, the fourth hypothesis (H_4) states that *different hallucinations configurations impact system safety differently*. Therefore H_4 evaluates whether these configurations lead to measurable differences in the likelihood of an accident and the proximity of crossing vehicles, supporting the validation of the HI module. Thus, H_4 was unfolded as:

- **$H_{4.1}$** The configuration of hallucinations influences the likelihood of an accident occurring;
- **$H_{4.2}$** The configuration of injected hallucinations influences the minimum distance between the AV and the nearest vehicle.

RQ5) How does the probability of hallucination occurrence impact system safety?

Higher probabilities are expected to introduce more noises into the MC, which are expected to produce higher noisy reactions, affecting the AV safety. The fifth hypothesis (H_5) asserts that *a higher probability of hallucination occurrence significantly increases the risk of system safety*. Then, H_5 was unfolded into:

- **$H_{5.1}$** Higher probabilities of hallucinations increase the likelihood of an accident occurring;
- **$H_{5.2}$** Higher probabilities of hallucinations reduces minimum distances between the AV and the nearest vehicle.

RQ6) How does hallucination persistence over time affect system safety?

Finally, the sixth hypothesis (H_6) states that *hallucination persistence (permanent vs. intermittent) significantly influences the likelihood of an accident occurring*. Persistence defines duration: intermittent hallucinations may allow recovery, whereas permanent ones accumulate risk. Reliability studies highlight persistence as a critical factor in safety assessment [70]. H_6 was decomposed into the following components:

- **$H_{6.1}$** Persistence influences the likelihood of an accident occurring compared to intermittent hallucinations;
- **$H_{6.2}$** Persistence influences the minimum distances between the AV and the nearest vehicle compared to intermittent hallucinations.

G. EXPERIMENTAL PROCEDURES

The proposed HI module was evaluated using the USP54 ReTS environment (Figure 13, element 1). A total of 201 distinct experimental conditions were designed. They encompassed a baseline condition (HI module off) and 200 conditions covering all possible combinations of HI (type, configuration, probability, and persistence) defined by element 2 in Figure 13 (see Section III-C). Approximately 50 simulations were executed for each of the 200 HI conditions (totaling around $10,000$ runs) and approximately $9,000$ runs for the baseline configuration.

During the simulation execution, log files recorded metrics measurements for each run (Table 6). After invalid simulations logs were discarded (see Section III-A for criteria), the results of $18,356$ valid log files ($8,695$ HI OFF and $9,661$ HI ON), were consolidated with scripts (element 3) into a dataset (element 4, Table 7) used to investigate the study hypotheses (Section III-F). Each entry dataset corresponds to an individual simulation encompassing the used HI setup and the safety outcomes (Accident flag and Minimum Distance between vehicles). Finally, this dataset was imported into R Studio (element 5), where the statistical analyzes detailed in Section III-H were performed to test the hypotheses and generate the research results (element 6).

TABLE 6. Simulation log content

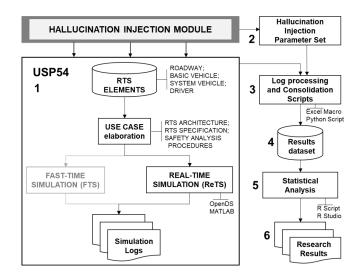| Metric (unit)	Description
Time (ms)	Timestamp
$x_{av}[m], z_{av}[m],$ $v_{av}[km/h]$	Position and speed of AV at the moment indicated by the timestamp
Steering Wheel Position $\sigma(t)$	Position of the steering wheel ranging from -1 to 1 commanded by the MC
Throttle Pedal Position $\alpha_{th}(t)$	Position of throttle pedal pressure ranging from 0 to 1. 0 indicates no actuation and 1 represents full pressure
Brake Pedal Position $\alpha_{br}(t)$	Position of brake pedal pressure ranging from 0 to 1. 0 indicates no actuation and 1 represents full pressure
$x_n[m], z_n[m],$ $v_n[km/h]$	Position and speed of each vehicle n at the transversal street at the moment indicated by the timestamp; $n = 1$ to 5

FIGURE 13. Elements used to support the experimental procedures

TABLE 7. Dataset Features Used to Verify Hypotheses H_1 to H_6

Variable	Description
Accident	Indicates whether a collision occurred in a given simulation run.
MinimalDistance	The minimum distance between the AV and the nearest crossing vehicle at the intersection.
HI Variables*	Define the characteristics and context of the injected hallucination for each run.

* Includes *ModuleActivation*, *HallucinationType*, *AffectedDomain*, *HallucinationConfiguration*, *HallucinationProbability*, and *HallucinationPersistence*.

H. ANALYSIS PROCEDURES

A combination of statistical approaches was used to test the hypotheses. Providing an adequate sample size to investigate each hypothesis is fundamental. Although the total sample size was $18,356$, ensuring a robust investigation of H_1, an analysis was performed to measure the representativeness of each category of condition evaluated by H_2 to H_6. Figure 14 illustrates the number of valid simulations consolidated per category of each variable of the HI module when HI was ON. It is noteworthy a robust sample size ($N \geq 450$) was achieved for a rigorous comparison with the baseline for all conditions evaluated by H_2 to H_6. To evaluate the impact on the probability of accident (collision or no collision), since the accident was a binary variable, logistic regression with a binomial distribution was used. This model is appropriate for binary data because it estimates how predictors affect the odds of a collision occurring [125]. Thus, logistic regression was used to test the hypotheses of $H_{1.1}$, $H_{1.2}$, $H_{2.1}$, $H_{2.2}$, $H_{3.1}$, $H_{3.2}$, $H_{4.1}$, $H_{4.2}$, $H_{5.1}$, $H_{5.2}$, $H_{6.1}$, $H_{6.2}$.

In contrast, for the continuous Minimum Distance variable, a standard linear regression model was applied. This approach provides clear estimates of the strength and direction of an association, allowing for a direct evaluation of how different factors influenced the physical safety buffer between vehicles [125]. Thus, linear regression was used to test the hypotheses
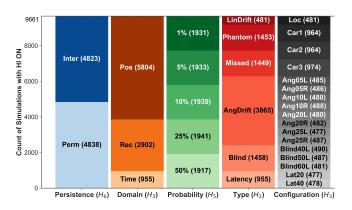
FIGURE 14. Breakdown of sample size per HI property's category

of $H_{1.1}$, $H_{1.2}$, $H_{2.1}$, $H_{2.2}$, $H_{3.1}$, $H_{3.2}$, $H_{4.1}$, $H_{4.2}$, $H_{5.1}$, $H_{5.2}$, $H_{6.1}$, $H_{6.2}$.

Then, to evaluate the impact of categorical predictors, an Analysis of Variance (ANOVA) was performed on the models to assess the overall significance of each factor on safety risk [126]. This approach is more informative than examining individual regression coefficients, which are limited to comparisons against a baseline.

All statistical analyzes were performed in RStudio, using established libraries for regression modeling, ANOVA, and hypothesis testing [127]. In all analyses, the significance threshold was set at $\alpha = 0.05$., consistent with standard practice in safety and reliability research [128].

I. EXPERIMENTAL HARDWARE INFRASTRUCTURE

The experiments were performed using a consumer-grade laptop equipped with an Intel Core i7 processor, 8 GB of RAM, and NVIDIA GPU GeForce MX150. The same computer was used to execute Python scripts to process the logs. Another similar laptop was used to run R Studio to perform the statistical analysis.

IV. RESULTS

The effect of hallucination injection on the system safety (H_1). A significant effect of injecting hallucinations on AV safety. First, the AV accident probability was found to significantly increase when hallucinations were injected (HI *ON*) compared to the baseline (HI *OFF*) (Wald $\chi^2 = 126.7$, $p < 0.001$, Table 8). In fact, the probability of an accident increases 3.09 ($p < 0.001$) times when HI is *ON* compared to when HI is *OFF*, demonstrating the effectiveness of the HI module in stressing the system. Thus, $H_{1.1}$ was accepted. Moreover, the minimum distance of the AV from the closest vehicle during the crossing was found to be significantly affected when HI is *ON* compared to the baseline (HI *OFF*) ($F(1, 18354) = 6,989$, $p < .001$, partial $\eta^2 = 0.28$, Table 9). In fact, the minimum distance was shortened on average by 18.6% when the hallucinations were injected, which demonstrates the HI created riskier situations during the crossings, even in situations when no accidents happened. That reinforces the HI module's ability to stress the AV

system. Therefore, H_1 was accepted since $H_{1.2}$ was also accepted.

The effect of hallucination type on the system safety (H_2). At a higher level of abstraction, the HI property Type, as a single construct, was found to significantly impact the AV safety when hallucinations were injected. In fact, it was found to significantly impact the accident likelihood (Wald $\chi^2 = 186.29$, $p < 0.001$, Table 8) ($H_{2.1}$ accepted) and the minimum distance between AV and the closest vehicle at the crossing ($F(6, 18349) = 1,223$, $p < .001$, partial $\eta^2 = 0.29$, Table 9) ($H_{2.2}$ accepted). Therefore, H_2 was accepted.

However, not all hallucination types were found to be equally dangerous. Regarding accident likelihood, as shown by the chart $H_{2.1}$ in Figure 15 and Table 11, missed detection ($OR = 5.20$, $p < 0.001$) and blind region ($OR = 4.92$, $p < 0.001$) hallucinations were the most significant, increasing the odds of a collision approximately five times. They were followed by angular drift ($OR = 2.52$, $p < 0.001$), Phantom ($OR = 2.23$, $p < 0.001$), and Latency ($OR = 1.81$, $p = 0.012$), which increased the odds of a collision by approximately 2.5, 2.2, and 1.8 times, respectively. However, linear drift, as a single category, was the only hallucination type whose effect on accident likelihood was not statistically significant ($p = 0.278$), although it still showed a 46% increase in accident probability. Considering the large standard deviation of the odds ratio shown in the chart $H_{2.1}$ in Figure 15, the linear drift hallucination is probably significant for some combinations of HI properties and not for others.

Regarding the minimum distance between the AV and the closest vehicle while crossing the transversal street, as shown by the chart $H_{3.1}$ in Figure 15 and Table 17, the bling region ($p < 0.001$) and angular drift hallucinations ($p < 0.001$) caused the most drastic reduction of the safety buffer, corresponding to an average of $1.85m$ and $1.72m$ in reduction, respectively. They were followed by significant reductions of $1.60m$ and $1.52m$ in the safety buffer, caused by Phanton ($p < 0.001$) and Linear Drift Hallucinations ($p < 0.001$), respectively. That result demonstrates that the linear drift hallucination also significantly stressed the system's safety. Finally, missed detection ($p < 0.001$) and Latency hallucinations ($p < 0.001$) caused significant reductions of the minimum distance measure between the AV and the closest vehicle on the transversal street during the crossing of $1.42m$ and $1.11m$, respectively.

The effect of the domain affected by the hallucination on the system safety (H_3). The Domain affected by perception in the HI had a significant impact on system safety when analyzed as a single HI property construct. First, it was found to significantly impact the accident likelihood (Wald $\chi^2 = 137.88$, $p < 0.001$, Table 8) ($H_{3.1}$ accepted) and the minimum distance between AV and the closest vehicle at the crossing ($F(3, 18352) = 2,428$, $p < .001$, partial $\eta^2 = 0.28$, Table 9) ($H_{3.2}$ accepted). Therefore, H_3 was accepted.

The analysis by domain category revealed that all domains targeted by the hallucinations had a significant impact on system safety. The recognition ($p < 0.001$), object position
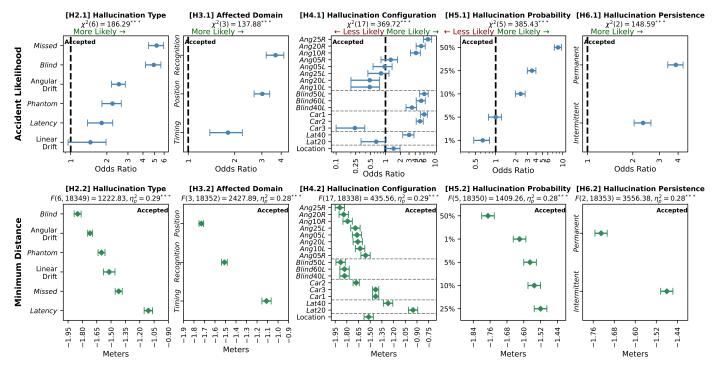
FIGURE 15. Testing results of supporting hypotheses (H_2–H_6). Five HI properties (Type, Domain, Configuration, Probability, Persistence) were tested on two variables. The top row shows odds ratios from logistic regression for *Accident Likelihood*, and the bottom row shows linear regression coefficients (β) for *Minimum Distance*, where negative values indicate a smaller safety buffer. Error bars show 95% confidence intervals

Note. For details on H_4, see Table 4. *** denotes significance level < 0.001.

($p < 0.001$), and information timing ($p = 0.012$) domains significantly increased the accident likelihood by 3.68, 3.02, and 1.81 times, respectively (chart $H_{3.1}$ in Figure 15 and Table 12). Moreover, the object position ($p < 0.001$), object recognition ($p < 0.001$), and information timing ($p < 0.001$) domains reduced significantly the minimum measured distance between the AV and the closest vehicle during the crossing by $1,73m$, $1.51m$, and $1.11m$ on the average, respectively (chart $H_{3.2}$ in Figure 15 and Table 18).

The effect of hallucination configuration on the system safety (H_4). The HI property Configuration had a significant impact on system safety when analyzed as a single construct. In fact, it was found to significantly impact the accident likelihood (Wald $\chi^2 = 369.72, p < 0.001$, Table 8) ($H_{4.1}$ accepted) and the minimum distance between AV and the closest vehicle at the crossing ($F(17, 18338) = 436, p < 0.001$, partial $\eta^2 = 0.29$, Table 9) ($H_{4.2}$ accepted). The likelihood of an accident was increased from 0.24 ($p = 0.014$) to 7.23 ($p < 0.001$) times when compared to the baseline (HI OFF) when car 3 had its detection missed, and an angular drift of 25°to the right of the AV was applied to miscalibrate the MP perception, respectively. The minimum distance between the AV and the closest vehicle was reduced from $8.62m$ ($p < 0.001$) to $7.67m$ ($p < 0.001$) on average when compared to the baseline (HI OFF), with a latency of 20 simulation periods added to the time it took for the MP information to reach the MC. Therefore, H_3 was accepted.

It is noteworthy that, although the HI property Configuration is an abstract construct that encompasses a distinct group of configurations associated with different hallucination Types, understanding its impact on the system safety demonstrates the HI properties' consistency in stressing the AV system.

A segmented analysis revealed the most damaging configurations related to each hallucination type, in terms of system safety. The angular drifts to the right resulted in higher accident likelihoods compared to the angular drifts to the left (chart $H_{4.1}$ in Figure 15 and Table 11). As shown in Figure 15 ($H_{4.1}$), the higher the angular drift to the right, the higher the likelihood of an accident. Moreover, they also reduced the minimum distance between the AV and the closest vehicle during the crossing (Figure 15 – $H_{4.2}$). That is, the higher the angular drift to the right, the shorter the minimum distance, which means a reduction in the safety buffer. On the other hand, for the angular drift to the left, it was not possible to identify a direct relationship between the magnitude order and the accident likelihood. In fact, the lowest angle (5°) to the left is associated with the highest accident likelihood ($OR = 0.96, p = 0.923$), followed by the highest angle (25°) ($OR = 0.81, p = 0.650$), which seems counterintuitive. The same was observed with the minimum distance. The highest angle (25°) drift to the left ($1.69m, p < 0.001$) caused the highest minimum distance reduction, followed by the lowest angle (5°) drift to the left ($1.67m, p < 0.001$), while $20°(1.66m, p < 0.001)$ and $10°(1.63m, p < 0.001)$ caused

the lowest minimum distance reductions. These results appear to be related to the specific characteristics of the simulated use case, such as the direction of the vehicles' flow on the transversal street, speed, and other road configurations. The same effect, probably strongly related to the use case characteristics, was observed with the blind region since positioning it at 40°, 50°, and 60°to the left caused the likelihood of an accident to increase by 3.43 ($p < 0.001$), 6.12 ($p < 0.001$), and 5.28 ($p < 0.001$), respectively, and the minimum distance to reduce by 1.83m ($p < 0.001$), 1.88m ($p < 0.001$), and 1.83m ($p < 0.001$), respectively. Thus, a direct relationship between the angular positioning of the blind strip and the safety metrics could not be established either. It is noteworthy that for both types of hallucinations, the higher the likelihood of accidents caused by each specific configuration tested, the lower the standard deviation observed. On the other hand, more consistent standard deviations can be observed for the minimum distance reductions caused by the distinct configurations tested for those hallucinations.

Furthermore, switching the vehicle targeted by the missed detection and phantom hallucinations on the transversal street also contributed significantly to increasing the accident likelihood and reducing the minimum distance between the AV and the closest vehicle. These hallucinations targeting the first (car 1), second (car 2), and third (car 3) vehicle increased the accident likelihood by 6.09 ($p < 0.001$), 5.00 ($p < 0.001$), and 0.24 ($p = 0.014$), respectively, and the minimum distance to reduce by 1.43m ($p < 0.001$), 1.68m ($p < 0.001$), and 1.43m ($p < 0.001$), respectively. However, it is noteworthy that the standard deviation of the accident likelihood was considerably higher when the third car was targeted, compared to the others. On the other hand, the differences in the standard deviation of the minimum distance were smoother among all the cars. That is another result that is probably closely related to the characteristics of the simulated use case, since changing the initial speed of the vehicles and their position would probably affect those findings.

However, when the hallucination causes a delay in the MP information reaching the MC, the results seem to follow intuition more closely and are less dependent on the use case configuration. In fact, the higher the latency, the higher the system safety risk. In fact, the information delays of 20 and 40 simulation cycles increased the likelihood of an accident by 0.65 ($p = 0.396$), and 3.00 ($p < 0.001$), respectively, and reduced the minimum distance by 0.95m ($p < 0.001$), and 1.27m ($p < 0.001$), respectively. Although a delay of 40 simulation cycles increases the accident likelihood by 4.61 times compared to a 20-cycle delay, the resulting likelihood exhibits a considerably narrower standard deviation. On the other hand, both cause a minimum distance reduction with a similar standard deviation. Finally, the location configuration for linear drift hallucination caused an odds ratio of 1.46 compared to the baseline ($p = 0.278$) and reduced the minimum distance between the AV and the closest vehicle at the crossing region to 1.52m ($p < 0.001$). In conclusion, H_4 was accepted since $H_{4.1}$ and $H_{4.2}$ were accepted.

The effect of hallucination probability on the system safety (H_5). The hallucination probability significantly compromised system safety by both increasing the likelihood of accidents (Wald $\chi^2 = 385.43, p < 0.001$, Table 8) and reducing the safety buffer (minimum distance) of the AV ($F(5, 18350) = 1,409, p < .001$, partial $\eta^2 = 0.28$, Table 9) ($H_{5.2}$ accepted). Figure 15 - $H_{5.1}$ illustrates that the higher the probability of hallucination, the higher the odds ratio of accident. In fact, while the hallucination probability of 1% decreases the accident probability by 0.64 times ($p = 0.097$) compared to the baseline, a 50% probability increases by 8.53 times ($p < 0.001$) with a narrower standard deviation. All tested rates caused a substantial reduction of approximately 1.5m to 1.8m (Figure 15 - $H_{5.2}$), suggesting that any level of hallucination occurrence can compromise vehicle spacing. Surprisingly, the smallest reduction in the safety buffer was observed with a hallucination probability of 25% (distance 1.52m, $p < 0.001$) rather than 1% (distance 1.62m, $p < 0.001$). Although the largest reduction was caused by the probability of 50% (distance 1.77m, $p < .001$), the second largest reduction was caused by the smallest simulated probability of hallucination, that is, 1%. That was an unexpected effect, possibly caused by the emerging system behavior related to the specificities of the tested use case. However, unlike the odds ratio, the minimum distance standard deviations were more consistent. In conclusion, since $H_{5.1}$ and $H_{5.2}$ were accepted, H_5 was accepted.

The effect of hallucination persistence on the system safety (H_6). The persistence of hallucination significantly impacted the system's safety. It impacted both the accident likelihood (Wald $\chi^2 = 148.59, p < 0.001$, Table 8) ($H_{6.1}$ accepted) and the minimum distance ($F(2, 18353) = 3,556$, $p < .001$, partial $\eta^2 = 0.28$, Table 9) ($H_{6.2}$ accepted). While intermittent hallucinations increased the accident likelihood by 2.34 times ($p < 0.001$) compared to the baseline, the permanent hallucinations increased it by 3.86 times ($p < 0.001$). This finding was consistent with the effect of persistence on the minimum distance, as permanent hallucinations caused a more severe reduction in the safety buffer (1.73m) compared to intermittent ones (1.48m). Since both $H_{6.1}$ and $H_{6.2}$ were accepted, H_6 was accepted.

V. DISCUSSION

The results of this study demonstrate that HI module and all of its properties significantly impacted the AV system safety by stressing the system in the experimental use case. By abstracting perception failures as high-level hallucinations, the HI framework provides a component-agnostic method to reveal safety issues and quantify their consequences. Acceptance of all hypotheses supports that HI has the potential to be both an effective perturbation mechanism and a statistically grounded tool for AV safety analysis. The experiments revealed that hallucination injection, as a general condition, more than tripled the probability of an accident (OR = 3.09) and severely degraded operational safety by reducing the mean minimum vehicle distance by 1.60m. This establishes a clear causal

link between perception-level hallucinations and system-level risk.

Moreover, the HI module provides a versatile and flexible framework for exploring new potential threads in AV systems, even before research on the fault modes of new sensors or fault mechanisms has been consolidated. In fact, the researcher only needs to propose and implement new hallucination configurations or types and execute simulations to understand how critical they are for the AV system's safety. This helps research prioritize and deepen its focus on the most critical mechanisms. It also helps to guide the development of the most critical protection mechanisms, which can potentially accelerate the AV development cycle.

The results indicate that not all hallucinations contribute equally to the degradation of safety. Failures causing complete information loss produced the most severe impacts. Specifically, missed detections and blind region hallucinations increased the probability of accident approximately five times. This result was corroborated in the minimum distance analysis, in which blind region hallucinations induced the most dangerous behavior, reducing the safety buffer by an average of $1.85m$. This is consistent with previous evidence showing that physical degradation of LiDARs or cameras significantly increases the risk of accidents [64]. Although less severe, hallucinations that simply distort information still represent a significant safety threat. For example, phantom objects more than doubled the odds of an accident, confirming that these hallucinations should not be ignored. This distinction is decisive because it reinforces the need to prioritize the most impactful hallucinations in both design-time validation and runtime monitoring.

Real-world data supports the experimental hierarchy of hallucination risks observed in this study. Across the major AV operators, missed detections are the most frequent failure mode. Reported cases include Waymo's failures to detect low-profile obstacles such as chains, sidewalks, and poles [129], [130], Cruise's collisions with stationary or partially occluded pedestrians [131], [132], Zoox's intermittent tracking errors that resulted in recalls [7], [133], and Tesla's diverse accidents [134], [135]. Phantom objects occur less often, but are frequently observed in Tesla's camera-based driverless system, where phantom braking events have caused several accidents [136], [137]. These findings suggest that the prevalence of each hallucination type depends on the architecture of the MP itself. [138]. Considering both the frequency of these failures and their estimated odds ratios, missed detections and blind region hallucinations represent the most critical categories. Although phantom objects seem to be more common in MP systems relying solely on cameras, such as Tesla's incidents, their lower odds ratio suggests a smaller contribution to accident risk compared to missed detections, which are less frequent but more severe. Hence, these evidences indicate that safety assessments of AVs should consider both the likelihood and severity of each hallucination type to guide the development of effective mitigation strategies.

Another important aspect uncovered was that the hallucination domain matters. Hallucinations that affect object recognition and position significantly increased the probability of accidents much more than the other domains. The impact on the minimum distance was particularly severe, with position-related hallucinations causing the largest average reduction in vehicle spacing ($1.73m$). This observation has direct implications for the design of AI-based perception models. The achievement of high accuracy for an object's position and its classification is a top priority, a point reinforced by recent safety incidents related to perception failures [139], [140]. Architectural choices, algorithms, training datasets, and test/validations must reflect the disproportionate safety impact of these specific perceptual dimensions.

Some specific operational configurations can create extreme risks in the use case evaluated. One of the most dangerous conditions identified was an angular drift of 25°to the right, which increased the probability of an accident by a factor of seven and was among the worst offenders for reducing the minimum distance ($1.89m$). This highlights how miscalibrations can produce disproportionate safety impacts and shows the value of the HI method for systematically identifying high-risk corner cases. This approach is consistent with other frameworks that use systematic perception error injection for virtual safety validation [65]. Furthermore, identifying these specific, high-impact failures is a critical prerequisite for developing real-time safety monitors that can assess the danger of a given failure based on the vehicle's current plan [141].

The analysis also revealed that both the probability and persistence of hallucinations significantly modify their safety impact. This finding contributes to a growing body of research showing that perception failures cannot be treated as isolated events, because their safety implications depend heavily on their dynamic and temporal characteristics. A more holistic understanding of component health requires reasoning about diagnostic information as it evolves over time [142]. The goal of a monitoring system is not merely to detect every error, but to identify the task-relevant failures that pose a genuine risk to the vehicle's current plan and to do so quickly enough to enable a safe recovery maneuver [143]. Achieving this needs continuous, real-time monitoring that provides swift alerts to facilitate a rapid response [68]. Therefore, the results highlight the need for continuous health and predictive fault detection monitoring systems that account for both the frequency and duration of failures, rather than treating them as isolated events.

The relationship between hallucination probability and accident risk was complex. Counterintuitively, a very low fault rate (1%) was associated with a slight reduction in accident risk compared to baseline, while a 5% rate did not show significant difference. This suggests that the AV's planner might react to minimal perceptual noise by adopting a more cautious behavior. However, any protective effect was quickly negated as the fault rate increased. As the hallucination rate increased, the danger grew substantially, making an accident at the 50% probability level over eight times more likely (OR

= 8.53). The persistence of a hallucination was also directly correlated with its impact on safety. Permanent hallucinations were substantially more dangerous than intermittent ones, increasing the probability of accidents more significantly (OR = 3.86 vs. 2.34) and causing a greater reduction in the minimum distance ($1.73m$ vs. $1.48m$).

These findings illustrate the value of an effect-centric strategy for AV safety. By focusing on observable system-level outcomes rather than the internal mechanics of specific sensors or algorithms, the HI framework provides a reusable, sensor-agnostic method for safety analysis across different AV architectures. The proposed framework that incorporates five-dimensional analysis (hallucination type, affected domain, configuration, probability, and persistence) offers a standardized taxonomy that can be adopted for simulation-based testing across AV platforms.

The implications extend beyond research. Regulators and standards bodies could employ hallucination-based tests as part of safety assurance pipelines, complementing performance benchmarks with explicit evaluation of resilience to perception failures. This would move AV testing closer to the safety infrastructures established in aviation and other high-reliability industries. However, unlike aviation, where a relatively small number of actors operate under unified and enforceable regulatory frameworks, the AV ecosystem remains fragmented and competitive, often constrained by proprietary incentives [25], [144]. This fragmentation, reinforced by what has been termed the "AV-IP problem" [56], has slowed the development of interoperable safety infrastructure and left systematic safety evaluation trailing the pace of AI innovation [10], [12]. The HI framework can potentially smooth this gap by offering a scalable and statistically validated toolkit that can support both industry practice and regulatory oversight.

It is noteworthy that some of the results might be not generalizable to other configurations and use cases. Thus, additional investigations are needed. Changing the initial condition and other conditions in the specific scenario could be a first sped to understand how generalizable those results are. Moreover, testing with various use cases systematically is also important to validate the generalization of the findings.

VI. CONCLUDING REMARKS

This study presents a simulation-based HI framework for systematically evaluating how perception failures (hallucinations) affect the safety of AVs. By modeling six types of sensor-agnostic hallucination and performing $18,356$ simulations in a high-risk intersection scenario, the analysis demonstrates that all HI module properties influence accident risk and minimum safe distances. The results quantify how different hallucination types and other properties impact safety-critical metrics, enhancing a previously developed framework.

The findings reveal that specific hallucinations pose disproportionately severe risks. These hallucinations can originate from both the software layer (i.e., machine learning (ML) perception models) and the hardware layer (i.e., sensors,

GPUs), reinforcing the value of the framework's component-agnostic design. The results offer actionable information for AV developers, regulators, and safety engineers by identifying high-impact failure scenarios that require careful testing and monitoring.

This study provides a foundational analysis in a controlled environment, a design choice that also defines the boundaries of the findings. The experiments were conducted in a single, unsignalized crossing with an experimental motion controller. Those, until a larger generalization effort is conducted, it is safer to consider the findings context-dependent. The risk profile for a given hallucination is not absolute. It will almost certainly change in different road geometries, traffic conditions, or with more sophisticated AV control systems. This context dependence defines the path forward. The next step is to apply the HI framework to a much broader set of driving scenarios, from complex urban intersections to highways, and in different AV architectures. Extending this investigation will be necessary to build a robust map of safety vulnerabilities, advance the design of robust AV safety systems, and contribute to the creation of industry validation protocols and future regulatory standards.

REFERENCES

[1] R. Ren, S. Basart, A. Khoja, A. Gatti, L. Phan, X. Yin, M. Mazeika, A. Pan, G. Mukobi, R. Kim *et al.*, "Safetywashing: Do ai safety benchmarks actually measure safety progress?" *Advances in Neural Information Processing Systems*, vol. 37, pp. 68 559–68 594, 2024.

[2] J. Van Brummelen, M. O'brien, D. Gruyer, and H. Najjaran, "Autonomous vehicle perception: The technology of today and tomorrow," *Transportation research part C: emerging technologies*, vol. 89, pp. 384–406, 2018.

[3] Y. Ma, Z. Wang, H. Yang, and L. Yang, "Artificial intelligence applications in the development of autonomous vehicles: A survey," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 2, pp. 315–329, 2020.

[4] National Transportation Safety Board, "Collision Between Vehicle Controlled by Developmental Automated Driving System and Pedestrian; Tempe, Arizona; March 18, 2018," National Transportation Safety Board, Washington, D.C., Highway Accident Report HAR1903, 2019. [Online]. Available: {https://www.ntsb.gov/investigations/AccidentReports/Reports/HAR1903.pdf}

[5] L. Tung, "Fatal tesla crash: Car was on autopilot when it hit truck, say investigators," *ZDNet*, May 2019. [Online]. Available: https://www.zdnet.com/article/fatal-tesla-crash-car-was-on-autopilot-when-it-hit-truck-say-investigators/

[6] S. Rivers. (2025) Deadly tesla crash raises questions about vision-based self-driving systems. CarScoops. [Online]. Available: https://www.carscoops.com/2025/06/tesla-fatal-crash-arizona-fsd-vision-only-safety-investigation/

[7] A. Palmer. (2025) Amazon's zoox robotaxi unit issues second software recall in a month after san francisco crash. CNBC. [Online]. Available: https://www.cnbc.com/2025/05/23/amazons-zoox-issues-software-recall-again-after-san-francisco-crash.html

[8] N. H. Mackworth, "The breakdown of vigilance during prolonged visual search," *Quarterly journal of experimental psychology*, vol. 1, no. 1, pp. 6–21, 1948.

[9] K. L. Lichstein, B. W. Riedel, and S. L. Richman, "The mackworth clock test: A computerized version," *The Journal of psychology*, vol. 134, no. 2, pp. 153–161, 2000.

[10] A. M. Nascimento, L. F. Vismari, P. S. Cugnasca, J. Camargo, J. de Almeida, R. Inam, E. Fersman, A. Hata, and M. Marquezini, "Concerns on the differences between ai and system safety mindsets impacting autonomous vehicles safety," in *Computer Safety, Reliability, and Security: SAFECOMP 2018 Workshops, ASSURE, DECSoS, SASSUR,*

STRIVE, and WAISE, Västerås, Sweden, September 18, 2018, Proceedings 37. Springer, 2018, pp. 481–486.

[11] A. M. Nascimento, L. F. Vismari, C. B. S. T. Molina, P. S. Cugnasca, J. B. Camargo, J. R. de Almeida, R. Inam, E. Fersman, M. V. Marquezini, and A. Y. Hata, "A systematic literature review about the impact of artificial intelligence on autonomous vehicle safety," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 12, pp. 4928–4946, 2019.

[12] Y. Bengio, G. Hinton, A. Yao, D. Song, P. Abbeel, T. Darrell, Y. N. Harari, Y.-Q. Zhang, L. Xue, S. Shalev-Shwartz *et al.*, "Managing extreme ai risks amid rapid progress," *Science*, vol. 384, no. 6698, pp. 842–845, 2024.

[13] K. Muhammad, A. Ullah, J. Lloret, J. Del Ser, and V. H. C. De Albuquerque, "Deep learning for safe autonomous driving: Current challenges and future directions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4316–4336, 2020.

[14] S. Rismani, R. Shelby, A. Smart, E. Jatho, J. Kroll, A. Moon, and N. Rostamzadeh, "From plane crashes to algorithmic harm: applicability of safety engineering frameworks for responsible ml," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023, pp. 1–18.

[15] S. Abrecht, A. Hirsch, S. Raafatnia, and M. Woehrle, "Deep learning safety concerns in automated driving perception," *IEEE Transactions on Intelligent Vehicles*, 2024.

[16] F. Mirzarazi, S. Danishvar, and A. Mousavi, "The safety risks of ai-driven solutions in autonomous road vehicles," *World Electric Vehicle Journal*, vol. 15, no. 10, p. 438, 2024.

[17] T. Miller, I. Durlik, E. Kostecka, P. Borkowski, and A. Łobodzińska, "A critical ai view on autonomous vehicle navigation: The growing danger," *Electronics*, vol. 13, no. 18, p. 3660, 2024.

[18] Y. Tian, K. Pei, S. Jana, and B. Ray, "Deeptest: Automated testing of deep-neural-network-driven autonomous cars," in *Proceedings of the 40th international conference on software engineering*, 2018, pp. 303–314.

[19] S. Wang and Z. Su, "Metamorphic object insertion for testing object detection systems," in *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering*, 2020, pp. 1053–1065.

[20] J.-A. Bolte, A. Bar, D. Lipinski, and T. Fingscheidt, "Towards corner case detection for autonomous driving," in *2019 IEEE Intelligent vehicles symposium (IV)*. IEEE, 2019, pp. 438–445.

[21] K. N. Kumar, C. Vishnu, R. Mitra, and C. K. Mohan, "Black-box adversarial attacks in autonomous vehicle technology," in *2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*. IEEE, 2020, pp. 1–7.

[22] H. Sun, S. Feng, X. Yan, and H. X. Liu, "Corner case generation and analysis for safety assessment of autonomous vehicles," *Transportation research record*, vol. 2675, no. 11, pp. 587–600, 2021.

[23] M. Koren, S. Alsaif, R. Lee, and M. J. Kochenderfer, "Adaptive stress testing for autonomous vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1–7.

[24] Q. Song, K. Tan, P. Runeson, and S. Persson, "Critical scenario identification for realistic testing of autonomous driving systems," *Software Quality Journal*, vol. 31, no. 2, pp. 441–469, 2023.

[25] G. Shimanuki, A. Nascimento, L. Vismari, J. Camargo, J. A. Almeida, and P. Cugnasca, "Navigating the edge with the state-of-the-art insights into corner case identification and generation for enhanced autonomous vehicle safety," *arXiv preprint arXiv:2503.00077*, 2025.

[26] ——, "CORTEX-AVD: A framework for CORner case testing and EXploration in autonomous vehicle development," *arXiv preprint arXiv:2504.03989*, 2025.

[27] S. Chen, Y. Liao, F. Wang, G. Wang, L. Wang, Y. Wang, and X. Zhu, "Toward the robustness of autonomous vehicles in the ai era," *The Innovation*, vol. 6, no. 3, 2025.

[28] P. Koopman and M. Wagner, "Autonomous vehicle safety: An interdisciplinary challenge," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 90–96, 2017.

[29] P. Koopman, "Practical experience report: Automotive safety practices vs. accepted principles," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2018, pp. 3–11.

[30] J. V. Carreira, D. Costa, and J. G. Silva, "Fault injection spot-checks computer system dependability," *IEEE Spectrum*, vol. 36, no. 8, pp. 50–55, 1999.

[31] D. Avresky, J. Arlat, J.-C. Laprie, and Y. Crouzet, "Fault injection for formal testing of fault tolerance," *IEEE Transactions on Reliability*, vol. 45, no. 3, pp. 443–455, 2002.

[32] Z. Segall, D. Vrsalovic, D. Siewiorek, D. Ysskin, J. Kownacki, J. Barton, R. Dancey, A. Robinson, and T. Lin, "Fiat-fault injection based automated testing environment," in *Twenty-Fifth International Symposium on Fault-Tolerant Computing, 1995,'Highlights from Twenty-Five Years'*. IEEE, 1995, p. 394.

[33] H. Madeira, M. Rela, F. Moreira, and J. G. Silva, "Rifle: A general purpose pin-level fault injector," in *Dependable Computing—EDCC-1: First European Dependable Computing Conference Berlin, Germany, October 4–6, 1994 Proceedings 1*. Springer, 1994, pp. 197–216.

[34] J. H. Lala and R. E. Harper, "Architectural principles for safety-critical real-time applications," *Proceedings of the IEEE*, vol. 82, no. 1, pp. 25–40, 1994.

[35] S. Dey and S.-W. Lee, "Multilayered review of safety approaches for machine learning-based systems in the days of ai," *Journal of Systems and Software*, vol. 176, p. 110941, 2021.

[36] T. Wu, Y. Dong, Z. Dong, A. Singa, X. Chen, and Y. Zhang, "Testing artificial intelligence system towards safety and robustness: State of the art." *IAENG International Journal of Computer Science*, vol. 47, no. 3, 2020.

[37] N. Maleki, B. Padmanabhan, and K. Dutta, "Ai hallucinations: a misnomer worth clarifying," in *2024 IEEE conference on artificial intelligence (CAI)*. IEEE, 2024, pp. 133–138.

[38] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE transactions on dependable and secure computing*, vol. 1, no. 1, pp. 11–33, 2004.

[39] H. Ziade, R. A. Ayoubi, R. Velazco *et al.*, "A survey on fault injection techniques," *Int. Arab J. Inf. Technol.*, vol. 1, no. 2, pp. 171–186, 2004.

[40] J. Karlsson, P. Liden, P. Dahlgren, R. Johansson, and U. Gunneflo, "Using heavy-ion radiation to validate fault-handling mechanisms," *IEEE micro*, vol. 14, no. 1, pp. 8–23, 2002.

[41] P. Civera, L. Macchiarulo, M. Rebaudengo, M. S. Reorda, and A. Violante, "Exploiting fpga for accelerating fault injection experiments," in *Proceedings Seventh International On-Line Testing Workshop*. IEEE, 2001, pp. 9–13.

[42] G. A. Kanawati, N. A. Kanawati, and J. A. Abraham, "Ferrari: A tool for the validation of system dependability properties." in *FTCS*, 1992, pp. 336–344.

[43] T. K. Tsai, R. K. Iyer, and D. Jewitt, "An approach towards benchmarking of fault-tolerant commercial systems," in *Proceedings of annual symposium on fault tolerant computing*. IEEE, 1996, pp. 314–323.

[44] J. Carreira, H. Madeira, and J. G. Silva, "Xception: A technique for the experimental evaluation of dependability in modern computers," *IEEE Transactions on Software Engineering*, vol. 24, no. 2, pp. 125–136, 1998.

[45] S. Han, K. G. Shin, and H. A. Rosenberg, "Doctor: An integrated software fault injection environment for distributed real-time systems," in *Proceedings of 1995 IEEE International Computer Performance and Dependability Symposium*. IEEE, 1995, pp. 204–213.

[46] A. Benso, P. Prinetto, M. Rebaudengo, and M. S. Reorda, "Exfi: a low-cost fault injection system for embedded microprocessor-based boards," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 3, no. 4, pp. 626–634, 1998.

[47] J. Aidemark, J. Vinter, P. Folkesson, and J. Karlsson, "Goofi: Generic object-oriented fault injection tool," in *2001 International Conference on Dependable Systems and Networks*. IEEE, 2001, pp. 83–88.

[48] A. L. Christensen, R. O'Grady, M. Birattari, and M. Dorigo, "Fault detection in autonomous robots based on fault injection and learning," *Autonomous Robots*, vol. 24, pp. 49–67, 2008.

[49] T. Vardanega, P. David, J.-F. Chane, W. Mader, R. Messaros, and J. Arlat, "On the development of fault-tolerant on-board control software and its evaluation by fault injection," in *Twenty-Fifth International Symposium on Fault-Tolerant Computing. Digest of Papers*. IEEE, 1995, pp. 510–515.

[50] S. Jha, S. S. Banerjee, J. Cyriac, Z. T. Kalbarczyk, and R. K. Iyer, "Avfi: Fault injection for autonomous vehicles," in *2018 48th annual ieee/ifip international conference on dependable systems and networks workshops (dsn-w)*. IEEE, 2018, pp. 55–56.

[51] S. Jha, T. Tsai, S. Hari, M. Sullivan, Z. Kalbarczyk, S. W. Keckler, and R. K. Iyer, "Kayotee: A fault injection-based system to assess the safety and reliability of autonomous vehicles to faults and errors," *arXiv preprint arXiv:1907.01024*, 2019.

[52] D. Garrido, L. Ferreira, J. Jacob, and D. C. Silva, "Fault injection, detection and treatment in simulated autonomous vehicles," in *Computational Science–ICCS 2020: 20th International Conference, Amsterdam, The*

Netherlands, June 3–5, 2020, Proceedings, Part I 20. Springer, 2020, pp. 471–485.

[53] M. Elgharbawy, A. Schwarzhaupt, G. Scheike, M. Frey, and F. Gauterin, "A generic architecture of adas sensor fault injection for virtual tests," in *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*. IEEE, 2016, pp. 1–7.

[54] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," *arXiv preprint arXiv:1903.12261*, 2019.

[55] G. Li, Y. Li, S. Jha, T. Tsai, M. Sullivan, S. K. S. Hari, Z. Kalbarczyk, and R. Iyer, "Av-fuzzer: Finding safety violations in autonomous driving systems," in *2020 IEEE 31st international symposium on software reliability engineering (ISSRE)*. IEEE, 2020, pp. 25–36.

[56] Z. Tahir and R. Alexander, "Intersection focused situation coverage-based verification and validation framework for autonomous vehicles implemented in carla," in *International Conference on Modelling and Simulation for Autonomous Systems*. Springer, 2021, pp. 191–212.

[57] Y. Liu, L. Wei, B. Luo, and Q. Xu, "Fault injection attack on deep neural network," in *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2017, pp. 131–138.

[58] J. Breier, X. Hou, D. Jap, L. Ma, S. Bhasin, and Y. Liu, "Practical fault attack on deep neural networks," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 2204–2206.

[59] A. S. Rakin, Z. He, J. Li, F. Yao, C. Chakrabarti, and D. Fan, "T-bfa: Targeted bit-flip adversarial weight attack," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7928–7939, 2021.

[60] S. Laskar, M. H. Rahman, and G. Li, "Tensorfi+: a scalable fault injection framework for modern deep learning neural networks," in *2022 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. IEEE, 2022, pp. 246–251.

[61] K. Pei, Y. Cao, J. Yang, and S. Jana, "Deepxplore: Automated whitebox testing of deep learning systems," in *proceedings of the 26th Symposium on Operating Systems Principles*, 2017, pp. 1–18.

[62] A. Chernikova, A. Oprea, C. Nita-Rotaru, and B. Kim, "Are self-driving cars secure? evasion attacks against deep neural networks for steering angle prediction," in *2019 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2019, pp. 132–137.

[63] A. Boloor, K. Garimella, X. He, C. Gill, Y. Vorobeychik, and X. Zhang, "Attacking vision-based perception in end-to-end autonomous driving models," *Journal of Systems Architecture*, vol. 110, p. 101766, 2020.

[64] H. Tian, W. Ding, X. Han, G. Wu, A. Guo, J. Zhang, W. Chen, J. Wei, and T. Zhang, "Testing the fault-tolerance of multi-sensor fusion perception in autonomous driving systems," *Proc. ACM Softw. Eng.*, vol. 2, no. ISSTA, Jun. 2025. [Online]. Available: https://doi.org/10.1145/3728910

[65] A. Piazzoni, J. Cherian, J. Dauwels, and L.-P. Chau, "Pem: Perception error model for virtual testing of autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 1, pp. 670–681, 2023.

[66] A. V. Malawade, T. Mortlock, and M. A. Al Faruque, "Hydrafusion: Context-aware selective sensor fusion for robust and efficient autonomous vehicle perception," in *2022 ACM/IEEE 13th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2022, pp. 68–79.

[67] T. Jin, C. Zhang, Y. Zhang, M. Yang, and W. Ding, "A hybrid fault diagnosis method for autonomous driving sensing systems based on information complexity," *Electronics*, vol. 13, no. 2, p. 354, 2024.

[68] W. Hou, W. Li, and P. Li, "Fault diagnosis of the autonomous driving perception system based on information fusion," *Sensors*, vol. 23, no. 11, p. 5110, 2023.

[69] J. Vargas, S. Alsweiss, O. Toker, R. Razdan, and J. Santos, "An overview of autonomous vehicles sensors and their vulnerability to weather conditions," *Sensors*, vol. 21, no. 16, p. 5397, 2021.

[70] H. Qi, S. Ganesan, and M. Pecht, "No-fault-found and intermittent failures in electronic products," *Microelectronics Reliability*, vol. 48, no. 5, pp. 663–674, 2008.

[71] O. Hainaut, "Ccd artifacts," n.d. [Online]. Available: \url{https://www.eso.org/~ohainaut/ccd/CCD_artifacts.html}

[72] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.

[73] Z. Jin, X. Ji, Y. Cheng, B. Yang, C. Yan, and W. Xu, "Pla-lidar: Physical laser attacks against lidar-based 3d object detection in autonomous vehicle," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 1822–1839.

[74] Oxford Instruments Andor, "Ccd blemishes and non-uniformities," n.d., accessed via Andor Learning. [Online]. Available: https://andor.oxinst.com/learning/view/article/ccd-blemishes-and-non-uniformities

[75] American Psychiatric Association, *American Psychiatric Association: Diagnostic and statistical manual of mental disorders*, 5th ed. American Psychiatric Association, Arlington, VA, 2013.

[76] F. Waters, J. D. Blom, K. Hugdahl, and I. E. Sommer, "Auditory hallucinations, not necessarily a hallmark of psychotic disorder," *Psychological medicine*, vol. 48, no. 4, pp. 529–536, 2018.

[77] K. Talluru, V. Kulandaivelu, N. Hutchins, and I. Marusic, "A calibration technique to correct sensor drift issues in hot-wire anemometry," *Measurement Science and Technology*, vol. 25, no. 10, p. 105304, 2014.

[78] B. J. Halkon and S. J. Rothberg, "Establishing correction solutions for scanning laser doppler vibrometer measurements affected by sensor head vibration," *Mechanical Systems and Signal Processing*, vol. 150, p. 107255, 2021.

[79] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough, and A. Mouzakitis, "A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 829–846, 2018.

[80] J. Janai, F. Güney, A. Behl, A. Geiger *et al.*, "Computer vision for autonomous vehicles: Problems, datasets and state of the art," *Foundations and trends® in computer graphics and vision*, vol. 12, no. 1–3, pp. 1–308, 2020.

[81] D. Hoiem, Y. Chodpathumwan, and Q. Dai, "Diagnosing error in object detectors," in *European conference on computer vision*. Springer, 2012, pp. 340–353.

[82] T. Amert, N. Otterness, M. Yang, J. H. Anderson, and F. D. Smith, "Gpu scheduling on the nvidia tx2: Hidden details revealed," in *2017 IEEE Real-Time Systems Symposium (RTSS)*. IEEE, 2017, pp. 104–115.

[83] K. Hsieh, E. Ebrahimi, G. Kim, N. Chatterjee, M. O'Connor, N. Vijaykumar, O. Mutlu, and S. W. Keckler, "Transparent offloading and mapping (tom) enabling programmer-transparent near-data processing in gpu systems," *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 204–216, 2016.

[84] M. Yang, "Avoiding pitfalls when using nvidia gpus for real-time tasks in autonomous systems," in *Proceedings of the 30th Euromicro Conference on Real-Time Systems*, 2018.

[85] S. Xu, H. Peng, and Y. Tang, "Preview path tracking control with delay compensation for autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 2979–2989, 2020.

[86] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[87] J. Bailenson, *Experience on demand: What virtual reality is, how it works, and what it can do*. WW Norton & Company, 2018.

[88] J. M. Box-Steffensmeier, J. Burgess, M. Corbetta, K. Crawford, E. Duflo, L. Fogarty, A. Gopnik, S. Hanafi, M. Herrero, Y.-y. Hong *et al.*, "The future of human behaviour research," *Nature Human Behaviour*, vol. 6, no. 1, pp. 15–24, 2022.

[89] K. A. Buetler, J. Penalver-Andres, Ö. Özen, L. Ferriroli, R. M. Müri, D. Cazzoli, and L. Marchal-Crespo, ""tricking the brain" using immersive virtual reality: modifying the self-perception over embodied avatar influences motor cortical excitability and action initiation," *Frontiers in human neuroscience*, vol. 15, p. 787487, 2022.

[90] C. Nass, I.-M. Jonsson, H. Harris, B. Reaves, J. Endo, S. Brave, and L. Takayama, "Improving automotive safety by pairing driver emotion and car voice emotion," in *CHI'05 extended abstracts on Human factors in computing systems*, 2005, pp. 1973–1976.

[91] H. Harris and C. Nass, "Emotion regulation for frustrating driving contexts," in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2011, pp. 749–752.

[92] M. Jabon, J. Bailenson, E. Pontikakis, L. Takayama, and C. Nass, "Facial expression analysis for predicting unsafe driving behavior," *IEEE Pervasive Computing*, vol. 10, no. 4, pp. 84–95, 2010.

[93] K. J. Lee, Y. K. Joo, and C. Nass, "Partially intelligent automobiles and driving experience at the moment of system transition," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2014, pp. 3631–3634.

[94] J. Koo, J. Kwac, W. Ju, M. Steinert, L. Leifer, and C. Nass, "Why did my car just do that? explaining semi-autonomous driving actions to improve driver understanding, trust, and performance," *International Journal on Interactive Design and Manufacturing (IJIDeM)*, vol. 9, no. 4, pp. 269–275, 2015.

[95] M. Slater, "Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 364, no. 1535, pp. 3549–3557, 2009.

[96] E. D. Swanson, F. Foderaro, M. Yanagisawa, W. G. Najm, P. Azeredo, A. John *et al.*, "Statistics of light-vehicle pre-crash scenarios based on 2011–2015 national crash data," United States. Department of Transportation. National Highway Traffic Safety ..., Tech. Rep., 2019.

[97] Federal Highway Administration (FHWA). (2024) About intersection safety. U.S. Department of Transportation. [Online]. Available: https://highways.fhwa.dot.gov/safety/intersection-safety/about

[98] L. Vismari, C. Molina, J. Camargo, J. Almeida, R. Inam, E. Fersman, and M. Marquezini, "A simulation-based safety analysis framework for autonomous vehicles—assessing impacts on road transport system's safety and efficiency," in *Safety and Reliability–Safe Societies in a Changing World*. CRC Press, 2018, pp. 2067–2075.

[99] C. B. S. T. Molina, L. F. Vismari, T. Fuji, J. Camargo, J. de Almeida, R. Inam, E. Fersman, A. Hata, and M. Marquezini, "Enhancing sensor capabilities of open-source simulation tools to support autonomous vehicles safety validation," in *Computer Safety, Reliability, and Security: SAFECOMP 2018 Workshops, ASSURE, DECSoS, SASSUR, STRIVE, and WAISE, Västerås, Sweden, September 18, 2018, Proceedings 37*. Springer, 2018, pp. 353–364.

[100] J. K. Naufal, J. B. Camargo, L. F. Vismari, J. R. de Almeida, C. Molina, R. I. R. González, R. Inam, and E. Fersman, "A 2 cps: A vehicle-centric safety conceptual framework for autonomous transport systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 6, pp. 1925–1939, 2017.

[101] C. Sommer, D. Eckhoff, A. Brummer, D. S. Buse, F. Hagenauer, S. Joerer, and M. Segata, "Veins: The open source vehicular network simulation framework," *Recent advances in network simulation: the OMNeT++ environment and its ecosystem*, pp. 215–252, 2019.

[102] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of sumo-simulation of urban mobility," *International journal on advances in systems and measurements*, vol. 5, no. 3&4, 2012.

[103] A. Varga, "Discrete event simulation system," in *Proc. of the European Simulation Multiconference (ESM'2001)*, vol. 17, 2001.

[104] R. Math, A. Mahr, M. M. Moniri, and C. Müller, "Opends: A new open-source driving simulator for research," *GMM-Fachbericht-AmE 2013*, vol. 2, 2013.

[105] S. V. Balkus, H. Wang, B. D. Cornet, C. Mahabal, H. Ngo, and H. Fang, "A survey of collaborative machine learning using 5g vehicular communications," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 1280–1303, 2022.

[106] Y. Zhao, C. Lei, Y. Shen, Y. Du, and Q. Chen, "Improving autonomous vehicle visual perception by fusing human gaze and machine vision," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 11, pp. 12 716–12 725, 2023.

[107] S. Kochanthara, T. Singh, A. Forrai, and L. Cleophas, "Safety of perception systems for automated driving: A case study on apollo," *ACM Transactions on Software Engineering and Methodology*, vol. 33, no. 3, pp. 1–28, 2024.

[108] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, "Sensor and sensor fusion technology in autonomous vehicles: A review," *Sensors*, vol. 21, no. 6, p. 2140, 2021.

[109] P. Koopman and M. Wagner, "Challenges in autonomous vehicle testing and validation," *SAE International Journal of Transportation Safety*, vol. 4, no. 1, pp. 15–24, 2016.

[110] T.-W. Weng, H. Zhang, P.-Y. Chen, J. Yi, D. Su, Y. Gao, C.-J. Hsieh, and L. Daniel, "Evaluating the robustness of neural networks: An extreme value theory approach," *arXiv preprint arXiv:1801.10578*, 2018.

[111] K. Geng, G. Dong, and W. Huang, "Robust dual-modal image quality assessment aware deep learning network for traffic targets detection of autonomous vehicles," *Multimedia Tools and Applications*, vol. 81, no. 5, pp. 6801–6826, 2022.

[112] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: the next computing revolution," in *Proceedings of the 47th design automation conference*, 2010, pp. 731–736.

[113] G. Premsankar, M. Di Francesco, and T. Taleb, "Edge computing for the internet of things: A case study," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1275–1284, 2018.

[114] M. A. Pollatschek, A. Polus, and M. Livneh, "A decision model for gap acceptance and capacity at intersections," *Transportation Research Part B: Methodological*, vol. 36, no. 7, pp. 649–663, 2002.

[115] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *Journal of artificial intelligence research*, vol. 31, pp. 591–656, 2008.

[116] M. W. Levin, S. D. Boyles, and R. Patel, "Paradoxes of reservation-based intersection controls in traffic networks," *Transportation Research Part A: Policy and Practice*, vol. 90, pp. 14–25, 2016.

[117] L. Westhofen, C. Neurohr, T. Koopmann, M. Butz, B. Schütt, F. Utesch, B. Neurohr, C. Gutenkunst, and E. Böde, "Criticality metrics for automated driving: A review and suitability analysis of the state of the art," *Archives of Computational Methods in Engineering*, vol. 30, pp. 1886–1784, 2023.

[118] H. Tian, Y. Jiang, G. Wu, J. Yan, J. Wei, W. Chen, S. Li, and D. Ye, "Mosat: finding safety violations of autonomous driving systems using multi-objective genetic algorithm," in *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2022, pp. 94–106.

[119] H. Ebadi, M. H. Moghadam, M. Borg, G. Gay, A. Fontes, and K. Socha, "Efficient and effective generation of test cases for pedestrian detection-search-based software testing of baidu apollo in svl," in *2021 IEEE International Conference on Artificial Intelligence Testing (AITest)*. IEEE, 2021, pp. 103–110.

[120] D. Kaufmann, L. Klampfl, F. Klück, M. Zimmermann, and J. Tao, "Critical and challenging scenario generation based on automatic action behavior sequence optimization: 2021 ieee autonomous driving ai test challenge group 108," in *2021 IEEE International Conference On Artificial Intelligence Testing (AITest)*. IEEE, 2021, pp. 118–127.

[121] F. Klück, Y. Li, J. Tao, and F. Wotawa, "An empirical comparison of combinatorial testing and search-based testing in the context of automated and autonomous driving systems," *Information and Software Technology*, vol. 160, p. 107225, 2023.

[122] Å. Svensson and C. Hydén, "Estimating the severity of safety related behaviour," *Accident Analysis & Prevention*, vol. 38, no. 2, pp. 379–385, 2006.

[123] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?" *Transportation research part A: policy and practice*, vol. 94, pp. 182–193, 2016.

[124] R. Borgovini, S. Pemberton, and M. Rossi, "Failure mode, effects, and criticality analysis (fmeca)," 1993.

[125] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to linear regression analysis*, 6th ed. John Wiley & Sons, 2021.

[126] P. D. O'connor and A. V. Kleyner, *Practical reliability engineering*, 4th ed. john wiley & sons, 2012.

[127] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2021. [Online]. Available: https://www.R-project.org/

[128] M. H. Kutner, C. J. Nachtsheim, W. Wasserman, and J. Neter, *Applied Linear Statistical Models*, 5th ed. McGraw-Hill Irwin, 2005.

[129] D. Shepardson and D. Sophia, "Waymo recalls 1,200 self-driving vehicles in US after minor collisions," https://www.reuters.com/business/autos-transportation/alphabets-waymo-recalls-over-1200-vehicles-after-collisions-with-roadway-2025-0 may 2025.

[130] National Highway Traffic Safety Administration, "Standing General Order on Crash Reporting," https://www.nhtsa.gov/laws-regulations/standing-general-order-crash-reporting, 2025.

[131] J. Gilboy, "California Says Cruise Lied About Robotaxi Crash Footage, Suspends Operations," https://www.thedrive.com/news/california-suspends-cruise-robotaxi-operations-over-risk-to-public-safety, 2023.

[132] J. Parker, "How Many Waymo, Cruise Driverless Cars Have Crashed?" https://www.govtech.com/transportation/how-many-waymo-cruise-driverless-cars-have-crashed, 2023.

[133] D. Shepardson, "Amazon's robotaxi unit Zoox agrees to software recall after self-driving Las Vegas crash," https://www.reuters.com/technology/amazons-robotaxi-unit-zoox-recalls-vehicles-after-self-driving-las-vegas-crash-202 may 2025.

[134] M. Mekelburg, "Tesla Spars in Court Over Autopilot Alert 2 Seconds Before Crash," https://www.bloomberg.com/news/articles/2025-07-18/tesla-spars-in-court-over-autopilot-alert-2-seconds-before-crash, jul 2025.

[135] BBC News, "Tesla in fatal California crash was on Autopilot," https://www.bbc.com/news/world-us-canada-43604440, March 2018.

[136] G. Kay, "Video shows 8-car pileup after a Tesla allegedly using Full Self-Driving stopped in a highway tunnel," https://www.businessinsider.com/tesla-stops-tunnel-pileup-accidents-driver-says-fsd-enabled-video-2023-1, jan 2023.

[137] Goksedef, Ece, "Tesla recalls more than 1.6 million cars in China over steering software issues," https://www.bbc.com/news/technology-67891080, jan 2024, accessed on: 2025-10-06.

[138] Y. Tang, H. He, Y. Wang, Z. Mao, and H. Wang, "Multi-modality 3d object detection in autonomous driving: A review," *Neurocomputing*, vol. 553, p. 126587, 2023.

[139] E. Carter, "Tesla's 'self-driving' software fails at train crossings, some car owners warn," *NBC News*, 09 2025, accessed: 2025-09-17. [Online]. Available: https://www.nbcnews.com/tech/elon-musk/tesla-full-self-driving-fails-train-crossings-drivers-warn-railroad-rcna225558

[140] Reuters, "Xiaomi will cooperate with investigation into fatal EV crash, says founder," *Reuters*, 04 2025, accessed: 2025-08-20. [Online]. Available: https://www.reuters.com/world/china/chinas-xiaomi-says-actively-cooperating-with-police-after-fatal-accident-2025-04-02/

[141] P. Antonante, S. Veer, K. Leung, X. Weng, L. Carlone, and M. Pavone, "Task-aware risk estimation of perception failures for autonomous vehicles," *arXiv preprint arXiv:2305.01870*, 2023.

[142] P. Antonante, H. G. Nilsen, and L. Carlone, "Monitoring of perception systems: Deterministic, probabilistic, and learning-based fault detection and identification," *Artificial Intelligence*, vol. 325, p. 103998, 2023.

[143] K. Chakraborty, Z. Feng, S. Veer, A. Sharma, B. Ivanovic, M. Pavone, and S. Bansal, "System-level safety monitoring and recovery for perception failures in autonomous vehicles," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 12 885–12 891.

[144] L. F. A. León and Y. Aoyama, "Industry emergence and market capture: The rise of autonomous vehicles," *Technological Forecasting and Social Change*, vol. 180, p. 121661, 2022.

···

APPENDIX A
SUPPLEMENTARY STATISTIC TABLES

All supplementary statistical tables supporting the analyses discussed in the main text are presented below for completeness and transparency.

A. ANOVA

Tables 8 and 9 present the detailed results from the ANOVA tests, summarizing the F-statistic, degrees of freedom (df), p-value, and partial eta squared (η_p^2) for each predictor variable.

TABLE 8. ANOVA results for the effects of HI properties on Accident Probability

HI Properties	LR χ^2	Df	p
$H_{1.1}$: Module Activation	126.7	1	< .001
$H_{2.1}$: Hallucination Type	186.29	6	< .001
$H_{3.1}$: Affected Domain	137.88	3	< .001
$H_{4.1}$: Hallucination Configuration	369.72	17	< .001
$H_{5.1}$: Hallucination Probability	385.43	5	< .001
$H_{6.1}$: Hallucination Persistence	148.59	2	< .001

Note: LR χ^2: Likelihood Ratio Chi-Squared, Df: Degrees of freedom.

TABLE 9. ANOVA results for the effects of HI properties on Minimum Distance

Predictor	SS	df	MS	F	p	η_p^2	90% CI [η_p^2]
$H_{1.2}$: Module Activation	11,785	1	11,785	6,989	<.001	0.28	[0.27, 0.28]
$H_{2.2}$: Hallucination Type	12,206	6	2,034	1,223	<.001	0.29	[0.28, 0.29]
$H_{3.2}$: Affected Domain	12,141	3	4,047	2,428	<.001	0.28	[0.28, 0.29]
$H_{4.2}$: Hallucination Configuration	12,292	17	723	436	<.001	0.29	[0.28, 0.30]
$H_{5.2}$: Hallucination Probability	11,856	5	2,371	1,409	<.001	0.28	[0.27, 0.29]
$H_{6.2}$: Hallucination Persistence	11,936	2	5,968	3,556	<.001	0.28	[0.27, 0.29]

Note: SS: Sum of Squares, df: Degrees of Freedom, MS: Mean Square, F: F-statistic, p: p-value, pη^2: partial η^2, pη^2 90%: partial η^2 90% CI [LL, UL], CI: Confidence Interval, LL: Lower Limit, UL: Upper Limit

B. ODDS RATIO

The following tables (10 to 15) detail the results of the OR analyzes performed to evaluate hypotheses H_1 through H_6. The OR quantifies the strength of the association between the predictor variables (various characteristics of injected hallucinations) and the binary outcome of an accident occurring. For each hypothesis, the corresponding table presents the calculated OR, its 95% confidence interval, and statistical significance, to provide a statistical summary of the implications discussed in the main text.

TABLE 10. Odds ratio results for predictor Module Activation (Hypothesis $H_{1.1}$)

Parameter	Odds Ratio	SE	95% CI	z	p
(Intercept)	0.01	$1.24e^{-3}$	[0.01, 0.02]	-45.62	< .001
FailureInjected [Yes]	3.09	0.34	[2.51, 3.85]	10.39	< .001

Note: SE: Standard Error, CI: Confidence Interval

TABLE 11. Odds ratio results for predictor Hallucination Type (Hypothesis $H_{2.1}$)

Parameter	Odds Ratio	SE	95% CI	z	p
(Intercept)	0.01	$1.24e^{-3}$	[0.01, 0.02]	-45.62	< .001
Linear Drift	1.46	0.51	[0.68, 2.74]	1.08	0.278
Phantom	2.23	0.41	[1.53, 3.17]	4.33	< .001
Missed Detection	5.20	0.75	[3.91, 6.88]	11.47	< .001
Latency	1.81	0.43	[1.11, 2.81]	2.51	0.012
Angular Drift	2.52	0.33	[1.94, 3.27]	7.00	< .001
Blind Region	4.92	0.72	[3.69, 6.54]	10.96	< .001

Note: SE: Standard Error, CI: Confidence Interval

TABLE 12. Odds ratio results for predictor Affected Domain (Hypothesis $H_{3.1}$)

Parameter	Odds Ratio	SE	95% CI	z	p
(Intercept)	0.01	$1.24e^{-3}$	[0.01, 0.02]	-45.62	< .001
Position	3.02	0.35	[2.40, 3.81]	9.42	< .001
Recognition	3.68	0.48	[2.85, 4.76]	10.02	< .001
Timing	1.81	0.43	[1.11, 2.81]	2.51	0.012

Note: SE: Standard Error, CI: Confidence Interval

TABLE 13. Odds ratio results for predictor Hallucination Configuration (Hypothesis $H_{4.1}$)

Parameter	Odds Ratio	SE	95% CI	z	p
(Intercept)	0.01	1.24e-03	[0.01, 0.02]	-45.62	< .001
Location	1.46	0.51	[0.68, 2.74]	1.08	0.278
Car1	6.09	0.95	[4.47, 8.25]	11.60	< .001
Car2	5.00	0.82	[3.60, 6.87]	9.77	< .001
Car3	0.24	0.14	[0.06, 0.63]	-2.46	0.014
Ang05L	0.96	0.40	[0.37, 2.01]	-0.10	0.923
Ang05R	1.28	0.47	[0.57, 2.48]	0.67	0.500
Ang10L	0.48	0.28	[0.12, 1.28]	-1.24	0.214
Ang10R	4.14	0.94	[2.60, 6.34]	6.28	< .001
Ang20L	0.48	0.28	[0.12, 1.28]	-1.24	0.214
Ang20R	5.27	1.10	[3.45, 7.83]	7.96	< .001
Ang25L	0.81	0.37	[0.29, 1.80]	-0.45	0.650
Ang25R	7.23	1.36	[4.96, 10.36]	10.56	< .001
Blind40L	3.43	0.83	[2.08, 5.40]	5.08	< .001
Blind50L	6.12	1.21	[4.10, 8.92]	9.17	< .001
Blind60L	5.28	1.10	[3.45, 7.85]	7.97	< .001
Lat20	0.65	0.33	[0.20, 1.55]	-0.85	0.396
Lat40	3.00	0.77	[1.75, 4.85]	4.25	< .001

Note: SE: Standard Error, CI: Confidence Interval

TABLE 14. Odds ratio results for predictor Hallucination Probability (Hypothesis $H_{5.1}$)

Parameter	Odds Ratio	SE	95% CI	z	p
(Intercept)	0.01	$1.24e^{-3}$	[0.01, 0.02]	-45.62	< .001
1%	0.64	0.17	[0.36, 1.05]	-1.66	0.097
5%	1.00	0.22	[0.64, 1.53]	0.02	0.985
10%	2.36	0.39	[1.70, 3.24]	5.25	< .001
25%	3.47	0.51	[2.59, 4.61]	8.48	< .001
50%	8.53	1.04	[6.73, 10.86]	17.60	< .001

Note: SE: Standard Error, CI: Confidence Interval

TABLE 15. Odds ratio results for predictor Hallucination Persistence (Hypothesis $H_{6.1}$)

Parameter	Odds Ratio	SE	95% CI	z	p
(Intercept)	0.01	$1.24e^{-3}$	[0.01, 0.02]	-45.62	< .001
Inter	2.34	0.30	[1.83, 3.01]	6.67	< .001
Perm	3.86	0.45	[3.08, 4.87]	11.59	< .001

Note: SE: Standard Error, CI: Confidence Interval

C. REGRESSION ANALYSIS

The following tables (16 to 21) present the statistical output of the linear regression analyzes. These models were developed to quantify the magnitude and direction of the relationship between different hallucination predictors and the minimum distance between the AV and other vehicles. Each table corresponds to a specific hypothesis and details the estimated model coefficients (β), their standard errors, and overall statistical significance, offering a summary of the implications discussed in the main text.

TABLE 16. Linear model results for predictor Module Activation (Hypothesis $H_{1.2}$)

Parameter	Coefficient	SE	95% CI	t(18354)	p
(Intercept)	8.62	0.01	[8.59, 8.65]	619.12	< .001
FailureInjected [Yes]	-1.60	0.02	[-1.64, -1.57]	-83.60	< .001

Note: SE: Standard Error, CI: Confidence Interval

TABLE 17. Linear model results for predictor Hallucination Type (Hypothesis $H_{2.2}$)

Parameter	Coefficient	SE	95% CI	t(18349)	p
(Intercept)	8.62	0.01	[8.59, 8.65]	623.29	< .001
Linear Drift	-1.52	0.06	[-1.64, -1.41]	-25.24	< .001
Phantom	-1.60	0.04	[-1.67, -1.53]	-43.80	< .001
Missed Detection	-1.42	0.04	[-1.49, -1.35]	-38.83	< .001
Latency	-1.11	0.04	[-1.19, -1.02]	-25.21	< .001
Angular Drift	-1.72	0.02	[-1.76, -1.67]	-68.80	< .001
Blind Region	-1.85	0.04	[-1.92, -1.78]	-50.63	< .001

Note: SE: Standard Error, CI: Confidence Interval, $t(18349)$ = t-statistic with 18349 degrees of freedom

TABLE 18. Linear model results for predictor Affected Domain (Hypothesis $H_{3.2}$)

Parameter	Coefficient	SE	95% CI	t(18352)	p
(Intercept)	8.62	0.01	[8.59, 8.65]	622.68	< .001
Position	-1.73	0.02	[-1.78, -1.69]	-79.20	< .001
Recognition	-1.51	0.03	[-1.57, -1.46]	-54.60	< .001
Timing	-1.11	0.04	[-1.19, -1.02]	-25.19	< .001

Note: SE: Standard Error, CI: Confidence Interval, $t(18352)$ = t-statistic with 18352 degrees of freedom

TABLE 19. Linear model results for predictor Hallucination Configuration (Hypothesis $H_{4.2}$)

Parameter	Coefficient	SE	95% CI	t(18339)	p
(Intercept)	8.62	0.01	[8.59, 8.65]	623.98	< .001
Location	-1.52	0.06	[-1.64, -1.41]	-25.27	< .001
Car1	-1.43	0.04	[-1.51, -1.34]	-32.60	< .001
Car2	-1.68	0.04	[-1.76, -1.59]	-38.34	< .001
Car3	-1.43	0.04	[-1.52, -1.35]	-32.89	< .001
Ang05L	-1.67	0.06	[-1.79, -1.56]	-27.84	< .001
Ang05R	-1.56	0.06	[-1.67, -1.44]	-25.92	< .001
Ang10L	-1.63	0.06	[-1.74, -1.51]	-26.93	< .001
Ang10R	-1.79	0.06	[-1.91, -1.67]	-29.84	< .001
Ang20L	-1.66	0.06	[-1.78, -1.54]	-27.43	< .001
Ang20R	-1.84	0.06	[-1.96, -1.73]	-30.59	< .001
Ang25L	-1.69	0.06	[-1.81, -1.57]	-27.89	< .001
Ang25R	-1.89	0.06	[-2.00, -1.77]	-31.46	< .001
Blind40L	-1.83	0.06	[-1.95, -1.71]	-30.59	< .001
Blind50L	-1.88	0.06	[-2.00, -1.77]	-31.39	< .001
Blind60L	-1.83	0.06	[-1.95, -1.71]	-30.34	< .001
Lat20	-0.95	0.06	[-1.07, -0.83]	-15.65	< .001
Lat40	-1.27	0.06	[-1.39, -1.15]	-20.96	< .001

Note: SE: Standard Error, CI: Confidence Interval, $t(18338)$ = t-statistic with 18338 degrees of freedom

TABLE 20. Linear model results for predictor Hallucination Probability
(Hypothesis $H_{5.2}$)

Parameter	Coefficient	SE	95% CI	t(18350)	p
(Intercept)	8.62	0.01	[8.59, 8.65]	619.77	< .001
1%	-1.62	0.03	[-1.68, -1.55]	-49.53	< .001
5%	-1.57	0.03	[-1.63, -1.51]	-48.13	< .001
10%	-1.55	0.03	[-1.62, -1.49]	-47.63	< .001
25%	-1.52	0.03	[-1.59, -1.46]	-46.74	< .001
50%	-1.77	0.03	[-1.83, -1.70]	-53.95	< .001

Note: SE: Standard Error, CI: Confidence Interval, $t(18350)$ =
t-statistic with 18350 degrees of freedom

TABLE 21. Linear model results for predictor Hallucination Persistence
(Hypothesis $H_{6.2}$)

Parameter	Coefficient	SE	95% CI	t(18353)	p
(Intercept)	8.62	0.01	[8.59, 8.65]	620.61	< .001
Inter	-1.48	0.02	[-1.53, -1.43]	-63.63	< .001
Perm	-1.73	0.02	[-1.77, -1.68]	-74.43	< .001

Note: SE: Standard Error, CI: Confidence Interval, $t(18353)$ =
t-statistic with 18353 degrees of freedom