Safety Interventions against Adversarial Patches in an Open-Source Driver Assistance System

Cheng Chen, Grant Xiao[†], Daehyun Lee [‡], Lishan Yang[‡], Evgenia Smirni[§], Homa Alemzadeh [†], Xugui Zhou
Louisiana State University, Baton Rouge, LA 70803 {cchen72, xuguizhou}@lsu.edu
[†]University of Virginia, Charlottesville, VA 22904 {yeq8mf, ha4d}@virginia.edu
[‡]George Mason University, Fairfax, VA 22030 {dlee217, lyang28}@gmu.edu
[§]William & Mary, Williamsburg, VA 23187 {esmirni}@cs.wm.edu

***Abstract*—Drivers are becoming increasingly reliant on advanced driver assistance systems (ADAS) as autonomous driving technology becomes more popular and developed with advanced safety features to enhance road safety. However, the increasing complexity of the ADAS makes autonomous vehicles (AVs) more exposed to attacks and accidental faults. In this paper, we evaluate the resilience of a widely used ADAS against safety-critical attacks that target perception inputs. Various safety mechanisms are simulated to assess their impact on mitigating attacks and enhancing ADAS resilience. Experimental results highlight the importance of timely intervention by human drivers and automated safety mechanisms in preventing accidents in both driving and lateral directions and the need to resolve conflicts among safety interventions to enhance system resilience and reliability. [Code Available at https://doi.org/10.6084/m9.figshare.28691090]**

***Index Terms*—ADAS, Sensor Attack, Adversarial Patch, Fault Injection, Safety Intervention, Driver, Autonomous Vehicles (AVs)**

I. Introduction

Advanced driver assistance systems (ADAS) offer Level-2 autonomous driving features [1] such as automatic lane centering (ALC) and adaptive cruise control (ACC). They are equipped on over 300 million passenger vehicles worldwide [2] and are estimated to reach a market value of USD 73.74 billion by 2031 [3]. This proliferation stems from the rapid advances in sensing and computing technologies and machine learning, which also raises concerns about the safety of vehicles and human drivers due to the increasing complexity and connectivity of ADAS. For example, recent studies show that Tesla's vehicles have the highest fatal accident and crash rates among all car brands in the U.S. [4], [5].

The core features of ADAS are ACC and ALC, which autonomously regulate a vehicle's speed and steering angle to maintain lane centering and a safe following distance from lead vehicles. To achieve these functionalities, ADAS relies on deep learning (DL) models to detect leading objects and lane markings from camera inputs. However, DL models are inherently vulnerable to input perturbations [6]–[9], and failures in object or lane line detection can lead to catastrophic safety risks [9], [10]. Consequently, studying the resilience of ADAS to camera input perturbations is critically important, particularly for vision-centric systems like Tesla Autopilot [11] and Subaru EyeSight [12].

Previous work has shown DL-based ADAS are vulnerable to perception attacks, including digital attacks that manipulate live camera feeds by directly compromising control software [13] and physical attacks employing adversarial stickers on road signs [14], the road surface [15], or camera lenses [16], as well as adversarial patches projected or printed on vehicles ahead [10], [17]. However, these works largely overlook the role of human-driver intervention and the impact of integrated safety mechanisms within the ADAS control loop, such as advanced emergency braking systems (AEBS) and lane departure warnings. These advanced safety features, designed to enhance road safety, are increasingly standard in modern ADAS and are critical to evaluating the resilience of ADAS.

While some studies have evaluated safety features like automatic emergency braking (AEB), these efforts are often limited to component-level analysis [18] without systematically testing the safety mechanisms and ADAS as a whole. Others focus narrowly on a single feature, such as ACC [9], neglecting the interdependencies among ADAS functionalities. Recent advancements in machine learning have shown promise in enhancing the resilience of robotic vehicles against sensor attacks [19]–[21]. However, these methods have yet to be evaluated on commercial ADAS.

To fill this gap, we systematically assess the resilience of a widely used open-source commercial ADAS, OpenPilot [22], against perception attacks (adversarial patches) while considering the interventions of existing safety mechanisms (such as AEB) and human drivers. Given that real-world road tests require significant time and resources and pose substantial risks to human safety and vehicle integrity during collision scenarios, especially when studying safety-critical attacks that could lead to crashes, we have developed a simulation platform for realistic experimental studies. Our platform integrates real-world control software, a physical-world simulator, a driver reaction simulator, and key ADAS safety mechanisms (AEB and FCW), all implemented in accordance with international standards and government-issued transportation guidelines.

The main contributions of the paper are as follows:

- Analyzing the effectiveness of basic ADAS safety interventions in mitigating the adversarial effect of physical patches on the ALC and ACC operation.
- Developing an open-source platform for realistic simulation of attacks and different levels of safety intervention.
- Studying the conflicts or coordination among different safety interventions, including automated mechanisms

and human drivers.
- Comparing basic safety mechanisms with an automated ML-based mitigation method.

Our study demonstrates that OpenPilot is highly vulnerable to adversarial patch attacks targeting ACC and ALC, often failing to detect the front vehicle at close range and exhibiting unsafe, aggressive speed control even in benign conditions, such as applying a hard brake only at a very close distance when approaching the lead vehicle. The findings highlight that basic safety mechanisms and human intervention may be more effective than certain ML-based automated mitigation methods in preventing accidents across both driving and lateral directions in the simulated scenarios. Lateral attacks, in particular, remain challenging to mitigate, though highly alert drivers achieve notably better prevention rates. Furthermore, the results underscore the critical role of independent sensors for AEBS, the potential of AEB to prevent lateral accidents, and the pressing need to resolve conflicts among safety features to enhance overall system reliability and resilience.

II. Preliminaries

A. Advanced Driver Assistance System

Advanced Driver Assistance Systems (ADAS) are technologies now widely used in vehicles to enhance road safety and driving comfort by freeing up driver attention and reducing human error. With a number of assistive features and semi-autonomous driving designs, it effectively enhances driving safety and driver comfort over long distances. ADAS systems rely on sensors and cameras to collect real-time data about the vehicle's surroundings, which can be used to enable predictive action and alert the driver to potential hazards.

Fig. 1 shows the control structure of a typical ADAS. Key ADAS functions include ACC, ALC, AEB, Blind Spot Detection, Traffic Sign Recognition, and Driver Monitoring Systems, among others. ADAS technologies are designed to automate driving tasks by gradually automating driving tasks and ultimately achieving fully automated driving functions. These systems significantly improve safety by reducing the number of accidents caused by human error, while increasing the overall comfort and convenience of driving. As technology advances, ADAS integration in vehicles is becoming more common, resulting in smarter and safer roads.

Autonomous Driving Levels: Autonomous driving technologies are categorized into different levels according to the vehicle's automation capabilities and the need for human intervention, ranging from Level 0 (no automation) to Level 5 (full automation). Current commercial ADAS operate at Level 2, where human drivers must continuously monitor the driving environment and ensure safety at all times [1].

Safety Mechanisms: Safety mechanisms are an integral part of ADAS to prevent accidents and protect vehicle occupants through a range of functions.

AEB is an important safety feature that automatically activates the brakes when it detects an impending collision, sometimes before the driver can react. This system is particularly

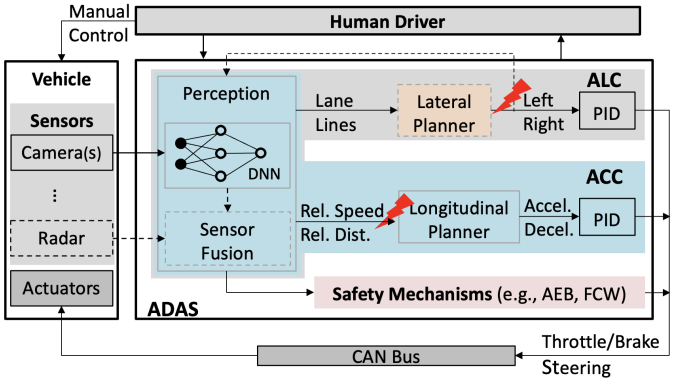

Fig. 1: Overview of the Control Structure of an ADAS.

effective at low speeds and in urban driving environments and can significantly reduce rear-end accidents. Closely working with AEB, Forward Collision Warning (FCW) monitors the road ahead with a forward-facing camera and radar, and when it detects a risk of a collision ahead, the system warns the driver to take evasive action. In many cases, the FCW system can warn the driver several seconds in advance, giving enough time to adjust speed or avoid obstacles.

These mechanisms mark a milestone in automotive safety, improving driving and road safety and paving the way for further ADAS innovations.

Openpilot: OpenPilot [22] is an open-source ADAS driving software from Comma.ai, Inc. When the OpenPilot software runs on Comma.ai's hardware and connects to a car, it becomes an autonomous driving system that can control the car, allowing for upgrades to original systems that lack ADAS features. OpenPilot supports more than 300 vehicle models [23] and has over 10,000 users, with more than 100 million miles driven. OpenPilot utilizes a system-level end-to-end design that uses DL models to predict the information necessary to avoid risks and plan the car's trajectory, based directly on images captured by the camera on the Comma hardware. This is a departure from traditional self-driving solutions, which rely on separate units of perception, prediction, and planning that operate in conjunction with each other.

B. MetaDrive Driving Simulator

MetaDrive [24] is an open-source simulator platform for autonomous driving research, specifically designed to provide an efficient and scalable simulation environment. It is the official simulator for OpenPilot by Comma.ai. MetaDrive provides a controlled-risk environment for researchers to test and optimize autonomous driving algorithms, including perception, decision making, and motion control strategies. We do not use CARLA [25], another widely used physical-world simulator, due to its incompatible Python environment with the OpenPilot (v0.9.7) control software.

III. Approach

A. Threat Model

We assume the attacker has the capability to launch efficient physical-world attacks on the camera inputs of ADAS by

introducing adversarial patches on the rear of a lead vehicle (LV) [26] or directly on the road [10] with the goal of compromising the ACC and ALC functionalities. These capabilities may include prior information about the victim vehicle's driving path and full knowledge of the target ADAS (e.g., ALC and ACC) implementation details through reverse-engineering a rented or purchased vehicle with the same model [15] or researching open-source ADAS materials [22]. The assumption of attack capability is realistic and similar to previous work in the literature [9], [10], [26]–[29]. The resulting perturbations compromise the ADAS perception module (see Fig. 1), leading to incorrect predictions of critical information, such as the lead vehicle's distance from the ego vehicle (EV) and lane line positions. Consequently, these errors propagate to the ADAS control software, causing malfunctions in the ALC and ACC modules, which can result in safety hazards or accidents, such as collisions or lane departures.

In this paper, we focus on analyzing the impact and effectiveness of existing safety mechanisms in mitigating physical-world perception attacks.

B. Attack Design and Implementation

In this paper, we examine attacks targeting ALC and ACC features of ADAS.

For ALC attacks, the attacker aims to deviate the vehicle's driving direction by altering predictions of lane lines or desired curvature. Prior work has demonstrated the feasibility of physical adversarial perturbations on camera inputs to achieve this goal [10]. Here, we adopt a method that compromises ALC functionality by deploying a well-crafted patch on the ego vehicle's driving path [10] (see Fig. 2). The attack is activated when the ego vehicle drives over the area containing the patch.

For ACC attacks, we utilize a method explored in previous studies [9], [26], [30], which involves displaying or projecting a physical patch on the rear of a lead vehicle (see top images in Fig. 2). This attack is triggered when the patch is detected by the ego vehicle within an effective range (e.g., less than 80 meters). It disrupts the prediction of the lead vehicle's distance, a critical input for the ACC decision-making module.

Note that only a portion of the adversarial patches lead to DNN (deep neural network) misprediction and further propagate to cause unsafe driving behaviors. To better evaluate the impact of safety interventions against these adversarial patches, we directly emulate the effect of the patches by injecting attacks into the DNN output and getting the range of attack values of mispredictions corresponding to adversarial patches from previous work [9], [10].

C. Safety Mechanisms Design

Passenger vehicles are increasingly equipped with advanced safety mechanisms such as AEB, FCW, lane departure warning, and firmware safety checking. In this paper, we implement three-level safety mechanisms to realistically assess the impact of these existing safety mechanisms on improving ADAS resilience to perception attacks (adversarial patches) and driving

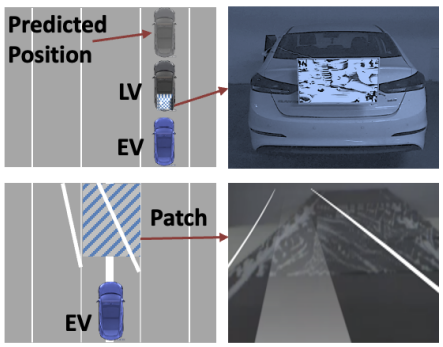

Fig. 2: Example physical attacks against ACC and ALC by adding an adversarial patch on the rear of the lead vehicle (LV) [26] or on the road [10].

safety, including basic-level AEBS, application-level safety checking, and human-level driver interventions.

AEBS: We adhere to the established guidelines and regulations for AEBS [31]–[33] and implement a time-to-collision-based phase-controlled AEBS, following the approach outlined in previous studies [9], [34]. Specifically, the AEBS operates by calculating the time to collision (TTC) between the ego vehicle and the lead vehicle, defined as the ratio of Relative Distance (RD) to Relative Speed (RS):

$$ttc = RD/RS \tag{1}$$

To determine the braking response, the system first estimates the time required for the ego vehicle to stop under human driver braking:

$$T_{stop} = V_{Ego}/a_{driver} \tag{2}$$

where, V_{Ego} is the ego vehicle's speed and a_{driver} is the driver's assumed deceleration.

An FCW is triggered to alert the driver of potential collision risks when the estimated time to collision is insufficient for the driver to respond (T_{react}) and stop the vehicle safely.

$$t_{fcw} = T_{react} + T_{stop} \tag{3}$$

In this study, we assume an average human reaction time of 2.5 seconds, as reported in the literature [9], [10], [35]. If the driver fails to react to the FCW alert in time, the AEBS activates phased braking based on speed-dependent TTC thresholds:

$$t_{pb1} = \frac{V_{Ego}}{3.8}, \quad t_{pb2} = \frac{V_{Ego}}{5.8}, \quad t_{fb} = \frac{V_{Ego}}{9.8} \tag{4}$$

Specifically, when ttc falls below the cascade thresholds for first-stage braking time (t_{pb1}), second-stage braking time (t_{pb2}) and full-force braking time (t_{fb}), the system applies 90% braking, 95% braking and full braking, respectively. The activation sequence of FCW alerts and corresponding AEBS actions are shown in Table I.

TABLE I: AEBS Actions.

TTC	$[t_{fcw}, t_{pb1}]$	$[t_{pb1}, t_{pb2}]$	$[t_{pb2}, t_{fb}]$	$[t_{fb}, 0]$
Action	FCW Alert	90% Brake	95% Brake	100% Brake

Note that in some car models, the AEBS may also rely on compromised DNN predictions as inputs, potentially degrading its functionality, whereas other models might be equipped with separate sensors to ensure reliable AEBS operation [9]. Furthermore, certain vehicles may completely disable AEBS functionality when ADAS is activated [23]. To account for these variations and realistically simulate AEBS implementation on actual passenger cars, we design the AEBS mechanism to operate under three distinct configurations: (1) AEBS is disabled, (2) AEBS is activated but relies on compromised sensor data, and (3) AEBS is activated and utilizes inputs from an independent, secure data source.

Firmware Safety Checks: In addition to safety checking in the control software, OpenPilot also implements firmware safety checking on the output control commands through a CAN interface device, PANDA [36], which is a universal OBD adapter developed by Comma.ai, providing access to almost all vehicle sensors over the CAN bus. However, this PANDA safety checking is not available in the simulation. To align our safety mechanism with real-world implementations, we replicate the logic from PANDA and design a software-based safety constraint checker that detects if command values are within a predefined safe range, thereby blocking unsafe control commands. For example, to prevent potential hazards, the maximum acceleration and deceleration of the vehicle should be between $2m/s^2$ and -3.5m/s^2 [37] (the exact thresholds set in PANDA with a more conservative design based on ISO 22179 [38]), respectively, and only control commands that fit within the safe range can be sent to the simulation platform for execution.

Human Driver Reactions: At Level 2 autonomy [1], drivers are required to monitor driving safety at all times and intervene in emergencies to prevent hazards. We implement a driver reaction simulator to evaluate the impact of driver interventions on longitudinal and lateral control.

As shown in Table II, when receiving an FCW alert, noticing unexpected acceleration, unsafe following distance (e.g., less than a vehicle length), or unsafe cruising speed (exceeding 10% of the speed limit [39]), or identifying a vehicle attempting to cut in from an adjacent lane, the driver initiates an emergency brake after the reaction time to avoid collisions. We implement the emergency braking following a driver brake behavior study [40]. For hazard mitigation in the lateral direction, if a lane departure warning (LDW) is triggered or the vehicle approaches a lane line too closely (predicted distance less than 0.5 meters), the driver steers the vehicle back to the center of the lane, following the reaction time.

We use a fixed average driver reaction time of 2.5 seconds for these interventions unless otherwise specified in our simulations, based on various government-issued transportation policy guidelines [41], [42] and existing literature [9], [10]. To account for the variability in drivers' reaction times and assess their impact on hazard mitigation, we also conduct additional experiments in Section IV-E4.

TABLE II: Driver Reaction Simulator.

Activation Condition	Driver Reaction	Reaction Time
FCW Alerts Unsafe Cruise Speed Unexpected Acceleration Unsafe Following Distance Other Vehicle Cutting in	Emergency Brake & Zero Throttle & No changes in the steering angle	2.5 seconds
Lane Departure Warning Unsafe Distance to Lane Lines	Steer vehicle back to the center of the lane	2.5 seconds

IV. Experiments

We develop a closed-loop simulation platform (see Fig. 3) to assess ADAS resilience and evaluate the effectiveness of various safety mechanisms. The platform integrates OpenPilot control software with the MetaDrive physical-world driving simulator and incorporates a fault injection engine alongside a safety intervention model as designed in Section III-C. To address conflicts among safety interventions, we assign different priorities to the various safety mechanisms in our simulations, with AEB having the highest priority and safety checking the lowest [43] [44].

Experiments are conducted on two Ubuntu 20.04 LTS machines, one with an NVIDIA RTX 3070 graphics card and the other with an NVIDIA RTX 3090 graphics card. The platform utilizes OpenPilot v0.9.7 (the latest stable version) and MetaDrive 0.4.2.3. Each OpenPilot simulation comprises 10,000 time steps, with each step lasting approximately 10 ms, resulting in a total time of 100 seconds per simulation.

A. Driving Scenarios

We simulate the following driving scenarios, which are identified as high-risk in the National Highway Traffic Safety Administration's (NHTSA) pre-collision scenario typology report [45]. In these scenarios, the ego vehicle cruises at 50 mph and approaches the lead vehicle from an initial distance of 60 or 230 meters. These distances are chosen to ensure the ego vehicle catches up with the lead vehicle on straight and curvy roads within our simulation.

- S1: The lead vehicle cruises at a constant speed (30 mph).
- S2: The lead vehicle cruises at 30 mph and then accelerates to 40 mph.
- S3: The lead vehicle cruises at 40 mph and then decelerates to 30 mph.
- S4: The lead vehicle cruises at 30 mph and suddenly brakes to a stop due to an obstacle.
- S5: The lead vehicle cruises at 30 mph, and another vehicle from the neighboring lane cuts into the ego vehicle's driving lane.
- S6: Two lead vehicles cruise at a constant speed (30 mph) in the same lane; then, the second lead vehicle (the one closer to the ego vehicle) changes lanes and moves into an adjacent lane.

A visualization of each designed driving scenario is also shown in Fig. 4. In the MetaDrive simulator, we choose a dry highway map for all scenarios. The default environment is set to a bright morning with stable lighting and clear visibility.

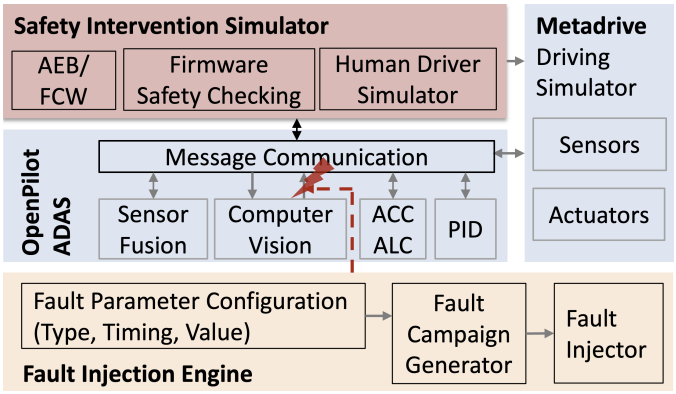

Fig. 3: Overview of closed-loop simulation platform.

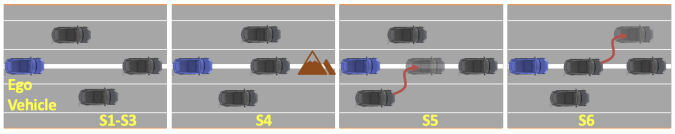

Fig. 4: Driving Scenarios.

B. Fault Injection

To simulate the effects of physical-world perception attacks on ACC and ALC (see Section III-A), we design a source-level fault injection (FI) engine capable of directly manipulating the outputs of the perception module by amounts consistent with those demonstrated in previous studies [9], [10]. Specifically, the perception attack on ACC targets the DNN's predictions of the lead vehicle's position. To replicate this effect, the fault injection engine directly alters the predicted relative distance (RD) to the lead vehicle [9]. Similarly, the fault injection engine modifies the desired curvature predicted by the perception module to simulate sensor attacks on ALC [10] (see Fig. 2). The desired curvature, defined as the reciprocal of the turning radius, dictates the sharpness of the vehicle's turns. In this study, we consider fault injection targeting a single variable and multiple variables.

For each attack type listed in Table III, the fault injection engine defines four parameters: (i) the target state variable, (ii) the magnitude of the error to inject, (iii) the trigger condition for the error, and (iv) the duration of the injected fault. Fault injections affecting the relative distance (RD) are triggered when the relative distance to the lead vehicle is less than 80 m, mimicking the perception of an adversarial patch on the rear of the lead vehicle by the ego vehicle. The injected fault values are set to 10 m, 15 m, and 38 m when the relative distance is within 80 m, 25 m, and 20 m, respectively, as reported in an existing work [9]. For attacks on ALC, the fault injection is activated when the ego vehicle crosses an adversarial patch on the road, introducing a 3% deviation in curvature output predictions. This deviation results in a lateral path offset corresponding to a steering angle adjustment of up to 10 degrees, which falls within a reasonable range reported in the literature [10]. Each configuration is repeated 10 times, resulting in 360 simulations (3 fault types $\times$2 initial positions $\times$6 driving scenarios).

TABLE III: Fault Injection Parameters.

Type	Target Variable	Attack Timing	Attack Value
Single	Relative Distance	RD < 80m	38-10m
	Desired Curvature	Ego vehicle drives over road patch area	3%
Mixed	RD & Curvature	RD < 80m or ego vehicle drives across patch	Same as above

C. Hazard and Accident

We consider two types of hazards and accidents:

- **A1**: Forward collision with the lead vehicle.
- **A2**: Driving out of the lane or colliding with side vehicles.
- **H1**: Ego vehicle violates maintaining safety distance with the lead vehicle, which may result in A1.
- **H2:** Ego vehicle drives too close to the lane lines (e.g., 0.1m), which may lead to A2.

D. Baselines

Machine learning (ML) techniques have demonstrated significant advancements in hazard recovery for drones and unmanned aerial vehicles [20], [21], yet their application to autonomous vehicles (AVs) remains relatively underexplored. To evaluate the effectiveness of ML-based automated methods in mitigating perception or sensor attacks against Level-2 ADAS, we develop a basic end-to-end ML baseline. This model takes as input the ego vehicle's speed, relative distance to the leading vehicle, lane line positions, and historical gas and steering values from previous control cycles to directly predict the expected gas and steering outputs.

We train the model on fault-free data spanning 20 control cycles (0.2 seconds at a 100Hz control frequency) and explore various configurations of a two-layer LSTM model, including 256-128, 256-64, 256-32, 128-64, 128-32, and 64-32 hidden units. The best-performing LSTM model uses 128 and 64 hidden units. Since adding a third layer does not improve performance, we retain the two-layer model as the ML baseline.

As shown in Algorithm 1, the mitigation mode is activated when the accumulated error between ML model predictions and OpenPilot controller outputs exceeds a preset threshold τ [46]. A bias parameter $b(t) > 0$ is introduced to ensure that no error accumulates in $S(t)$ under normal conditions.

Under attack, we assume the ML model has access to fault-free input data from an independent or redundant sensor measurement, following the design of previous works [20] that isolate compromised sensors. The ML model then generates a mitigation output, y_{ML}, which is executed by the actuator. The mitigation mode remains active until the error between ML outputs and OpenPilot outputs falls within threshold $b(t)$.

E. Results

1) ADAS Driving Performance: We first evaluate OpenPilot's ability to drive normally across six designed scenarios without any fault injection and present the results in Table IV.

Algorithm 1: Baseline ML-based Hazard Mitigation.

Input: Current state x(t) based on fault-free sensor data, threshold τ, bias parameter $b(t)$
Output: Control output to actuator $y(t)$

$S(t) = 0$ ▷ Initialize accumulated error
$b(t) = b_0 > 0$ ▷ Initialize bias parameter
while *Vehicle is operating* **do**
  $X_t \leftarrow [x(t-19), x(t-18), ..., x(t)]$ ▷ 20 continuous frames
  $Y_t \leftarrow [y(t-20), y(t-19), ..., y(t-1)]$ ▷ Historical outputs
  $y_{OP} \leftarrow$ OpenPilot output
  $y_{ML} \leftarrow$ MLmodel.predict(X_t, Y_t)
  $\delta \leftarrow |y_{ML} - y_{OP}|$
  $S(t+1) = max(0, S(t) + \delta - b(t))$ ▷ Keep S(t) Non-negative
  if $S(t+1) > \tau$ **then**
    recovery_mode ← True
  if *recovery_mode* **then**
    $y(t) \leftarrow y_{ML}$
    if $\delta \leq b(t)$ **then**
      recovery_mode ← False
      $S(t) \leftarrow 0$ ▷ Reset S(t) when exiting recovery mode
  else
    $y(t) \leftarrow y_{OP}$
end

TABLE IV: Hardest Brake Value in Different Scenarios.

Driving Scenario	Hazard	Accident	Following Distance(m)	Hard Brake Value	min. TTC(s)	min. t_{fcw}(s)
S1	1/20	0/20	26.02	32.7%	5.70	4.42
S2	1/20	0/20	29.15	15.7%	5.27	4.38
S3	2/20	1/20	29.88	46.7%	3.71	4.39
S4	**10**/20	**10**/20	**23.72**	**86.7%**	**0.85**	**3.24**
S5	2/20	1/20	29.42	58.0%	2.33	3.90
S6	3/20	0/20	28.15	30.3%	5.44	4.46

We observe that OpenPilot performs well in controlling the vehicle without accidents in most scenarios, including cut-in situations (S5). However, in Scenario S4, the OpenPilot fails to prevent a collision even in the absence of an attack, particularly when the lead vehicle brakes abruptly on a curve. The ego vehicle, still approaching at high speed without prior speed control before the curve, collides due to an insufficient emergency braking distance, despite triggering the FCW alarm.

Although OpenPilot maintains a larger following distance with the lead vehicle (LV) during a stable cruise stage compared to a previous version [9], OpenPilot exhibits aggressive braking behavior when approaching the lead vehicle, even in the absence of any attacks. This overly aggressive braking, while not causing immediate hazards, can adversely affect ride smoothness and increase the risk of rear-end collisions in congested traffic conditions. As shown in Fig. 5, taking Scenario 1 as an example, when the ego vehicle approaches the lead vehicle, the ego vehicle's speed suddenly drops from about 21.7m/s to 9.6m/s, a decrease of 55.8% within 4.7 seconds, followed by similar speed fluctuations.

Furthermore, ALC struggles to maintain the vehicle in the center of the lane, as evidenced by the minimal distance to lane lines, particularly during high-speed turns (see Table V). This increases the risk of side collisions and highlights insufficient coordination between the ALC and ACC modules. Additionally, the vehicle's aggressive speed control on curves with excessive angles underscores the need for improved lateral and longitudinal stability to ensure safe driving performance.

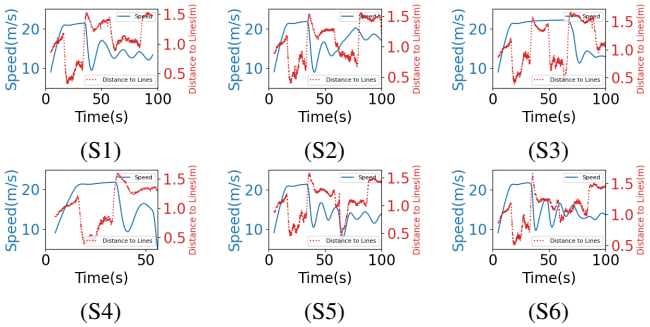

Fig. 5: Speed and Distance to Lane Lines when Approaching LV.

TABLE V: Minimal Distance to Lane Lines.

Driving Scenarios	S1	S2	S3	S4	S5	S6
Distance to Lane Lines (m)	0.45	0.49	0.07	0.63	0.44	0.59

Observation 1: OpenPilot fails to ensure safety in certain scenarios (e.g., S4) and exhibits aggressive braking when approaching a lead vehicle and inadequate lateral control for maintaining lane centering, highlighting the need for improved coordination between the ALC and ACC modules, particularly on curvy roads.

2) ADAS Resilience under Attack: We assess the resilience of ADAS against attacks (see Table III) without any safety intervention. We present the results in Table VI. We see in Table VI that accidents happen in all the simulations, indicating the insufficiency of OpenPilot in tolerating adversarial perception attacks and the need for safety mechanisms to improve safety. Additionally, 17.5% of out-of-the-lane (A2) accidents are caused by fault injection against ACC, resulting from overspeeding on curvy roads and lane changes to avoid forward collisions in cut-in scenarios. For mixed attacks, more A2 accidents occur than A1 accidents due to the shorter time needed to trigger accidents in the latter direction, highlighting the severe vulnerability of ALC to perception attacks.

We also observe that OpenPilot fails to recognize the lead vehicle when the relative distance (RD) is short. As shown in Fig. 6, the ego vehicle continues to approach the lead vehicle using the tampered input. However, once the ego vehicle gets within a certain range, such as 2 meters, OpenPilot is unable to detect the lead vehicle through the camera. This failure causes the ego vehicle to accelerate, resulting in a collision.

Observation 2: OpenPilot is unable to tolerate perception attacks against ACC and ALC and fails to recognize the front vehicle at a short distance.

3) Evaluation of Safety Interventions: Table VI also shows the results of fault injection experiments with various configurations of safety interventions. We see in Table VI that the designed safety mechanisms can prevent 19.17-100% accidents across different safety configurations and fault types, demonstrating safety interventions' key role in mitigating ADAS perception attacks.

Among the tested mechanisms, AEB utilizing an independent data source consistently outperforms other strategies in

TABLE VI: Fault Injection with or w/o Safety Interventions.

Fault	Safety Interventions					Accidents		Prevented	Avg. Mitigation Time(s)			Trigger Rate		
Type	Driver	Safety Check	AEB Comp.	AEB Indep.	ML Model	A1	A2	Accident	AEB	Driver Brake	Driver Steering	AEB	Driver Brake	Driver Steering
Relative Distance	-	-	-	-	-	82.50%	17.50%	0%	-	-	-	-	-	-
	✓	✓	-	-	-	55.00%	0%	45.00%	-	3.19	0.97	-	51.67%	6.70%
	✓	✓	✓	-	-	49.17%	0%	50.83%	3.55	2.65	5.50	20.00%	42.50%	3.33%
	✓	✓	-	✓	-	0%	0%	**100%**	3.30	2.01	0.93	83.33%	93.33%	4.17%
	-	-	✓	-	-	80.83%	0%	19.17%	3.46	-	-	19.17%	-	-
	-	-	-	✓	-	0%	0%	**100%**	3.26	-	-	100%	-	-
	✓	-	-	-	-	51.17%	0.83%	40.00%	-	2.72	3.00	-	66.00%	3.33%
	-	-	-	-	✓	1.67%	65.83%	32.5%	-	-	-	-	-	-
Desired Curvature	-	-	-	-	-	0%	100%	0%	-	-	-	-	-	-
	✓	✓	-	-	-	0%	54.17%	45.83%	-	4.26	0.92	-	30.00%	40.83%
	✓	✓	✓	-	-	0%	52.72%	47.27%	3.14	0.82	0.77	43.64%	32.73%	16.36%
	✓	✓	-	✓	-	0%	46.67%	**53.33%**	3.52	3.15	0.13	42.50%	39.17%	14.17%
	-	-	✓	-	-	0%	60%	40.00%	3.55	-	-	40.83%	-	-
	-	-	-	✓	-	0%	59.17%	40.83%	3.12	-	-	48.83%	-	-
	✓	-	-	-	-	0%	51.67%	48.33%	-	4.26	0.93	-	30.00%	41.67%
	-	-	-	-	✓	0%	60.00%	40.00%	-	-	-	-	-	-
Mixed	-	-	-	-	-	4.17%	95.83%	0%	-	-	-	-	-	-
	✓	✓	-	-	-	7.50%	54.17%	38.33%	-	3.21	3.16	-	59.17%	32.50%
	✓	✓	✓	-	-	8.33%	41.67%	50.00%	3.35	2.93	2.85	12.50%	80.83%	72.50%
	✓	✓	-	✓	-	0%	48.33%	51.67%	3.68	3.20	0.87	41.67%	36.67%	12.50%
	-	-	✓	-	-	6.67%	67.50%	25.83%	0.05	-	-	25.83%	-	-
	-	-	-	✓	-	0%	58.33%	41.67%	3.62	-	-	43.33%	-	-
	✓	-	-	-	-	8.33%	22.50%	**69.17%**	-	3.18	0.93	-	64.17%	32.50%
	-	-	-	-	✓	0%	76.92%	23.08%	-	-	-	-	-	-

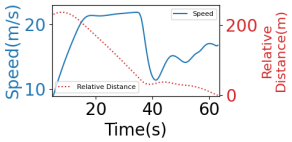

Fig. 6: Speed and Relative Distance under Fault Injection.

preventing forward collisions, achieving an accident prevention rate of up to 100%. This underscores the effectiveness of AEB in addressing immediate forward hazards. However, when AEB uses compromised data, the accident prevention rate drops significantly to 19.17%, highlighting the importance of designing an AEB with inputs from an independent sensor or a fault-resilient data stream. While AEB is not designed to address hazards or accidents in the lateral direction, we observe that it prevents up to 40.83% of A2 accidents caused by attacks on ALC. This occurs because the ego vehicle's aggressive acceleration toward the lead vehicle activates AEB, stopping the ego vehicle from driving out of the lane.

We also observe human driver intervention, including braking and steering, prevents a significant portion of accidents across different fault types (40-69.17%), indicating the important role of driver in ensuring safety in addition to advanced safety features (e.g., AEB, FCW) when driving Level-2 AVs [1]. However, the average driver reaction time (2.5 seconds) introduces a critical delay, which impacts the effectiveness of interventions in fast-evolving situations. For example, in relative distance attacks, where rapid deceleration is required to prevent collisions, the delayed response limits the accident prevention rate compared to AEB (40% vs. 100%). Conversely, for curvature attacks leading to lane deviations, driver steering interventions demonstrate higher efficacy.

Observation 3: AEB and driver intervention can prevent accidents in both driving and lateral directions and are critical to ensuring driving safety.

For the mixed fault type, human intervention achieves the highest hazard prevention rate at 69.17%. However, when combined with AEB, the prevention rate drops to 50–51.67%. This decline is primarily due to the prevalence of lateral-direction accidents (A2), where AEB underperforms compared to human intervention. Since AEB has the highest priority in the safety hierarchy, it overrides human inputs, leading to more unsuccessful mitigation cases. These findings underscore the challenges of prioritizing safety mechanisms to prevent conflicts across diverse driving and fault scenarios. While AEB is critical for immediate risk mitigation, human intervention remains vital for addressing complex or prolonged faults. Driver responses provide a complementary layer of safety, particularly when automated systems are disabled or compromised.

Observation 4: Better coordination of safety mechanisms at different levels is needed to resolve conflicts and ensure safety under complex attacks.

4) Evaluation with Different Driver Reaction Times: Although the designed safety mechanisms mitigate up to 53.33% of the accidents caused by attacks against ALC, they still fail to prevent nearly half of the accidents. This is due to the short mitigation time in the lateral direction and the critical delay in driver reaction. To account for the variability in driver's reaction time and further assess their impact on hazard mitigation outcomes, we rerun the simulations with reaction times varying between 1.0s and 3.5s [47] [48], using the same scenario and fault settings introduced in Sections IV-A and IV-B, while enabling only driver interventions. This results in

TABLE VII: Prevention Rate vs. Driver Reaction Time.

Fault Type	Driver Reaction Time(s)					
	1.0	1.5	2.0	2.5	3.0	3.5
Relative Distance	53.33%	55%	55%	40%	43.33%	41.67%
Desired Curvature	77.50%	55.83%	58.11%	48.33%	52.50%	40.00%
Mixed	70.83%	70.00%	68.33%	69.17%	60.83%	53.33%

360 simulations for each reaction time configuration. Table VII demonstrates that the success rate of driver intervention increases when reaction time is shorter than 2 seconds, emphasizing the importance of staying alert while driving.

Observation 5: Attacks against ALC cannot be easily mitigated while an alert driver may improve the success rate of accident prevention.

5) Evaluation with Various Environmental Conditions: Environmental conditions impact hazard mitigation in two ways: lighting affects visibility and perception, while weather conditions, such as rain, impact road friction. In MetaDrive, there is no direct interface to modify environmental conditions, and adjusting lighting using third-party renderers degrades OpenPilot's functionality under normal operation. Therefore, we can only simulate varying road friction by modifying friction parameters to represent wet or icy conditions. To assess the effect of different weather conditions on safety mechanisms for hazard mitigation, we rerun all simulations across all fault types and scenarios.

Table VIII shows that the hazard mitigation rate declines as road friction decreases. However, the safety mechanisms maintain a similar prevention rate at 50% road friction (e.g., heavy rain [49]), indicating a certain level of robustness in mitigating hazards under varying weather conditions. When road friction is reduced to 25% (e.g., icy road [50]), we observe a significant drop in the hazard mitigation rate against attacks targeting desired curvature, highlighting the increased vulnerability of ALC in severe weather conditions.

TABLE VIII: Hazard Prevention Rate vs. Road Friction.

Fault Type	Road Friction			
	Default	25% off	50% off	75% off
Relative Distance	50.83%	51.65%	47.50%	43.33%
Curvature/Lateral	47.27%	44.17%	45.83%	18.33%

[1]Enabled Safety Intervention: Driver, Safety Check, AEB Compromised

6) Comparison to ML-based Mitigation: From Table VI we see that the ML-based baseline achieves a hazard prevention rate of 23.08%–40.00% across different fault types, which, while reasonable, remains lower than driver intervention and AEB. While the ML model prevented nearly all A1 accidents caused by relative distance attacks, it introduced new A2 accidents, highlighting its inadequate performance in vehicle centering and the need for further efforts in designing and training a more advanced ML model.

Observation 6: Basic safety mechanisms (e.g., AEB) or human intervention may be more effective than certain automated ML-based mitigation methods in preventing accidents in ADAS in our simulated scenarios.

V. Threats to Validity

This study is conducted solely in simulation, leaving the real-world impact of safety mechanisms on perception attacks against ADAS in actual vehicles uncertain. Further, the driver reaction model relies on fixed rules and thresholds, which may not fully represent human behavior or distribution of reaction time during emergencies. We try to mitigate this weakness by developing a realistic testbed and examining various driver reaction times. Testing on actual vehicles and modeling more complex driver reactions are directions of future work.

VI. Related Work

Existing work has explored the vulnerability of autonomous driving and AVs against faults or attacks [51] targeting Lidar [52], GPS [53], radar [54], camera [10], [55], perception model [56]–[58], CAN bus [35], multi-sensor fusion [59], object tracking [7], or controller [60]. However, most works do not account for the existing safety mechanisms. Few studies on safety interventions have focused solely on a single ADAS feature, such as ACC [9], without accounting for the interactions between multiple safety mechanisms. Additionally, some studies have evaluated the resilience and robustness of autonomous systems [61]–[68]. However, these efforts often focus on individual components, such as AEB [66] or FCW [18], and lack a comprehensive, system-wide testing approach. A comparison of our work with other existing work is shown in Table IX.

TABLE IX: Comparison with Existing Work.

Work	Attack Vector	Target	AEB	Driver Intervention	Autonomy Level
[35]	Control Commands	ACC, ALC	N	Y	L2
[69]		ACC	Y	N	L2
[66]	Perception Input	LiDAR, Camera	Y	N	N/A
[68]		ALC	N	N	L2
[59]		LiDAR, Camera, Radar	N	N	N/A
[51]		ADAS	N	N	L2
Ours		ACC, ALC	**Y**	**Y**	L2

VII. Conclusion

This paper systematically evaluates the resilience of an open-source ADAS to adversarial patch attacks and the effectiveness of safety interventions in mitigating hazards and ensuring driving safety. Our findings show that OpenPilot is highly susceptible to these attacks targeting ACC and ALC, often failing to detect the front vehicle at close range and displaying unsafe, aggressive speed control even in benign conditions. Lateral attacks remain challenging to mitigate, though highly alert drivers achieve better prevention rates. Further, the results emphasize the potential of AEB to prevent lateral accidents, the importance of independent sensors for AEBS, and the urgent need to address conflicts among safety features to enhance overall system reliability and resilience.

Acknowledgment

This work was partially supported by the National Science Foundation (NSF) under Grants 2402940, 2402941, 2402942, 2410856, and CCI HC-3Q24-047.

REFERENCES

[1] "SAE Levels of Driving Automation™ Refined for Clarity and International Audience," https://www.sae.org/blog/sae-j3016-update, 2021.

[2] "ADAS Market Report 2030." [Online]. Available: https://www.marketsandmarkets.com/Market-Reports/driver-assistance-systems-market-1201.html

[3] "Global Advanced Driver Assistance Systems (ADAS) Market Size and Forecast." [Online]. Available: https://www.verifiedmarketresearch.com/product/global-advanced-driver-assistance-systems-adas-market-size-and-forecast/

[4] "The 23 Most Dangerous Cars on the Road." [Online]. Available: https://www.iseecars.com/most-dangerous-cars-study

[5] U.S. Department of Transportation National Highway Traffic Safety Administration, "Summary Report: Standing General Order on Crash Reporting for Level 2 Advanced Driver Assistance Systems," Tech. Rep., 2022. [Online]. Available: https://www.nhtsa.gov/sites/nhtsa.gov/files/2022-06/ADAS-L2-SGO-Report-June-2022.pdf

[6] S. Jha, S. Banerjee, T. Tsai, S. K. Hari, M. B. Sullivan, Z. T. Kalbarczyk, S. W. Keckler, and R. K. Iyer, "ML-Based Fault Injection for Autonomous Vehicles: A Case for Bayesian Fault Injection," in *2019 49th annual IEEE/IFIP international conference on dependable systems and networks (DSN)*. IEEE, 2019, pp. 112–124.

[7] S. Jha, S. Cui, S. Banerjee, J. Cyriac, T. Tsai, Z. Kalbarczyk, and R. K. Iyer, "ML-Driven Malware that Targets AV Safety," in *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2020, pp. 113–124.

[8] X. Zhou, M. Kouzel, and H. Alemzadeh, "Robustness testing of data and knowledge driven anomaly detection in cyber-physical systems," in *2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. IEEE, 2022, pp. 44–51.

[9] X. Zhou, A. Chen, M. Kouzel, H. Ren, M. McCarty, C. Nita-Rotaru, and H. Alemzadeh, "Runtime Stealthy Perception Attacks against DNN-Based Adaptive Cruise Control Systems," in *ACM Asia Conference on Computer and Communications Security (ASIA CCS)*, 2025.

[10] T. Sato, J. Shen, N. Wang, Y. Jia, X. Lin, and Q. A. Chen, "Dirty Road Can Attack: Security of Deep Learning Based Automated Lane Centering under {Physical-World} Attack," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 3309–3326.

[11] "Tesla Autopilot," https://www.tesla.com/autopilot.

[12] "Subaru EyeSight," https://www.subaru.com/eyesight.html.

[13] Y. Jia, Y. Lu, J. Shen, Q. A. Chen, Z. Zhong, and T. Wei, "Fooling Detection Alone is not Enough: First Adversarial Attack against Multiple Object Tracking," *arXiv:1905.11026*, 2019.

[14] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust Physical-World Attacks on Deep Learning Visual Classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1625–1634.

[15] Tencent, "Experimental Security Research of Tesla Autopilot," *Tencent Keen Security Lab*, 2019.

[16] J. Li, F. Schmidt, and Z. Kolter, "Adversarial Camera Stickers: A Physical Camera-Based Attack on Deep Learning Systems," in *International Conference on Machine Learning*, 2019, pp. 3896–3904.

[17] C. Ma, N. Wang, Q. A. Chen, and C. Shen, "WIP: Towards the Practicality of the Adversarial Attack on Object Tracking in Autonomous Driving," in *Inaugural International Symposium on Vehicle Security & Privacy*, 2023.

[18] Y. Ma, J. A. Sharp, R. Wang, E. Fernandes, and X. Zhu, "Sequential Attacks on Kalman Filter-Based forward Collision Warning Systems," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 10, 2021, pp. 8865–8873.

[19] H. Choi, S. Kate, Y. Aafer, X. Zhang, and D. Xu, "Software-Based Realtime Recovery from Sensor Attacks on Robotic Vehicles," in *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*, 2020, pp. 349–364.

[20] P. Dash, G. Li, Z. Chen, M. Karimibiuki, and K. Pattabiraman, "PID-Piper: Recovering Robotic Vehicles from Physical Attacks," in *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2021, pp. 26–38.

[21] P. Dash, E. Chan, and K. Pattabiraman, "Specguard: Specification Aware Recovery for Robotic Autonomous Vehicles from Physical Attacks," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 1849–1863.

[22] Comma.ai, "OpenPilot." [Online]. Available: https://github.com/commaai/openpilot

[23] "Supported Cars by OpenPilot," https://github.com/commaai/openpilot/blob/master/docs/CARS.md.

[24] Q. Li, Z. Peng, Z. Xue, Q. Zhang, and B. Zhou, "MetaDrive: Composing Diverse Driving Scenarios for Generalizable Reinforcement Learning," *arXiv preprint arXiv:2109.12674*, 2021.

[25] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An Open Urban Driving Simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[26] S. Hoory, T. Shapira, A. Shabtai, and Y. Elovici, "Dynamic Adversarial Patch for Evading Object Detection Models," *arXiv preprint arXiv:2010.13070*, 2020.

[27] A. Chahe, C. Wang, A. Jeyapratap, K. Xu, and L. Zhou, "Dynamic Adversarial Attacks on Autonomous Driving Systems," *arXiv preprint arXiv:2312.06701*, 2023.

[28] Y. Man, R. Muller, M. Li, Z. B. Celik, and R. Gerdes, "That Person Moves Like a Car: Misclassification Attack Detection for Autonomous Systems Using Spatiotemporal Consistency," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 6929–6946.

[29] Y. Lou, Y. Zhu, Q. Song, R. Tan, C. Qiao, W.-B. Lee, and J. Wang, "A First Physical-World Trajectory Prediction Attack via LiDAR-Induced Deceptions in Autonomous Driving," *arXiv preprint arXiv:2406.11707*, 2024.

[30] Y. Guo, T. Sato, Y. Cao, Q. A. Chen, and Y. Cheng, "Adversarial Attacks on Adaptive Cruise Control Systems," in *Proceedings of Cyber-Physical Systems and IoT Week 2023*, 2023, pp. 49–54.

[31] R. Schram, A. Williams, and M. van Ratingen, "Implementation of Autonomous Emergency Braking (AEB), the Next Step in Euro NCAP'S Safety Assessment," *ESV, Seoul*, 2013.

[32] "UN Regulation No 152 – Uniform Provisions Concerning the Approval of Motor Vehicles with Regard to the Advanced Emergency Braking System (AEBS) for M1 and N1 Vehicles [2020/1597]," http://data.europa.eu/eli/reg/2020/1597/oj, pp. 66–89, 2020.

[33] "GRVA-12-50r1e.pdf," https://unece.org/sites/default/files/2022-01/GRVA-12-50r1e.pdf.

[34] T. Alsuwian, R. B. Saeed, and A. A. Amin, "Autonomous Vehicle with Emergency Braking Algorithm Based on Multi-Sensor Fusion and Super Twisting Speed Controller," *Applied Sciences*, vol. 12, no. 17, p. 8458, Aug. 2022.

[35] X. Zhou, A. Schmedding, H. Ren, L. Yang, P. Schowitz, E. Smirni, and H. Alemzadeh, "Strategic Safety-Critical Attacks against an Advanced Driver Assistance System," in *2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2022, pp. 79–87.

[36] Comma.ai, "Panda." [Online]. Available: https://github.com/commaai/panda

[37] "OpenPilot - Safety Architecture," 2018. [Online]. Available: https://blog.comma.ai/how-to-write-a-car-port-for-openpilot/#background--safety-architecture

[38] *Intelligent transport systems – Full speed range adaptive cruise control (FSRA) systems – Performance requirements and test procedures*, International Organization for Standardization International Standard ISO 22 179:2009, 2009.

[39] Virginia DMV, "Safe Driving," https://www.dmv.virginia.gov/webdoc/pdf/dmv39d.pdf.

[40] J. G. Gaspar and D. V. McGehee, "Driver Brake Response to Sudden Unintended Acceleration while Parking," *Transportation research interdisciplinary pertives*, vol. 2, p. 100039, 2019.

[41] "California Commercial Driver Handbook," https://www.dmv.ca.gov/portal/uploads/2020/06/comlhdbk.pdf, 2019.

[42] N. S. Council, "Reference Guide for NSC DDC Instructor-led Training," https://www.nsc.org/getmedia/a46d07cb-faf1-4572-8317-661e7f77ef7a/instructor-admin-reference-guide.pdf?srsltid=AfmBOop5BGVGMZl63t_31waS-4mgVvsUl4of_64EOIWQELG30mgonCkW, 2014.

[43] M. International, "Automotive Technology: What to Know About Automatic Emergency Braking," April 2024, [Online; accessed 5-December-2024]. [Online]. Available: https://maycointernational.com/blog/automotive-technology-what-to-know-about-automatic-emergency-braking/

[44] V. T. News, "New Automatic Emergency Braking Safety Standard will Save Lives, Expert Explains," April 2024, [Online; accessed 5-

December-2024]. [Online]. Available: https://news.vt.edu/articles/2024/04/Automatic-emergency-braking-safety-expert.html

[45] W. G. Najm, J. D. Smith, M. Yanagisawa, and John A. Volpe National Transportation Systems Center (U.S.), "Pre-Crash Scenario Typology for Crash Avoidance Research," Tech. Rep. DOT-VNTSC-NHTSA-06-02, Apr. 2007.

[46] O. A. Grigg, V. Farewell, and D. Spiegelhalter, "Use of Risk-Adjusted CUSUM and RSPRT Charts for Monitoring in Medical Contexts," *Statistical methods in medical research*, vol. 12, no. 2, pp. 147–170, 2003.

[47] University of Idaho, "Brake reaction time," https://www.webpages.uidaho.edu/niatt_labmanual/chapters/geometricdesign/theoryandconcepts/BrakeReactionTime.htm, n.d., [Online; accessed 27-February-2025].

[48] H. Makishita and K. Matsunaga, "Differences of Drivers' Reaction Times According to Age and Mental Workload," *Accident Analysis & Prevention*, vol. 40, no. 2, pp. 567–575, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0001457507001418

[49] Supercars.net, "How Much Does Rain Reduce Your Traction?" https://www.supercars.net/blog/how-much-does-rain-reduce-your-traction/, n.d., [Online; accessed 27-February-2025].

[50] M. Rasol, F. Schmidt, and S. Ientile, "Weather condition effect on the road surface friction: A Preliminary assessment based on sensor data," in *Life-Cycle of Structures and Infrastructure Systems*. CRC Press, 2023, pp. 2187–2194.

[51] B. Nassi, Y. Mirsky, D. Nassi, R. Ben-Netanel, O. Drokin, and Y. Elovici, "Phantom of the ADAS: Securing Advanced Driver-Assistance Systems from Split-Second Phantom Attacks," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 293–308. [Online]. Available: https://doi.org/10.1145/3372297.3423359

[52] T. Sato, Y. Hayakawa, R. Suzuki, Y. Shiiki, K. Yoshioka, and Q. A. Chen, "WIP: Practical Removal Attacks on LiDAR-Based Object Detection in Autonomous Driving," in *Inaugural International Symposium on Vehicle Security & Privacy*, 2023.

[53] J. Shen, J. Y. Won, Z. Chen, and Q. A. Chen, "Drift with Devil: Security of Multi-Sensor Fusion Based Localization in High-Level Autonomous Driving under GPS Spoofing," in *Proceedings of the 29th USENIX Conference on Security Symposium*, 2020, pp. 931–948.

[54] R. Komissarov and A. Wool, "Spoofing Attacks against Vehicular FMCW Radar," in *Proceedings of the 5th Workshop on Attacks and Solutions in Hardware Security*, 2021, pp. 91–97.

[55] D. Nassi, R. Ben-Netanel, Y. Elovici, and B. Nassi, "MobilBye: Attacking ADAS with Camera Spoofing," 2019. [Online]. Available: https://arxiv.org/abs/1906.09765

[56] A. Schmedding, P. Schowitz, X. Zhou, Y. Lu, L. Yang, H. Alemzadeh, and E. Smirni, "Strategic resilience evaluation of neural networks within autonomous vehicle software," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2024, pp. 33–48.

[57] A. Schmedding, L. Yang, A. Jog, and E. Smirni, "Aspis: Lightweight neural network protection against soft errors," in *35th IEEE International Symposium on Software Reliability Engineering, ISSRE 2024, Tsukuba, Japan, October 28-31, 2024*. IEEE, 2024, pp. 248–259.

[58] L. Yang, B. Nie, A. Jog, and E. Smirni, "Practical resilience analysis of GPGPU applications in the presence of single- and multi-bit faults," *IEEE Trans. Computers*, vol. 70, no. 1, pp. 30–44, 2021.

[59] Y. Zhu, C. Miao, H. Xue, Y. Yu, L. Su, and C. Qiao, "Malicious Attacks against Multi-Sensor Fusion in Autonomous Driving," in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, ser. ACM MobiCom '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 436–451. [Online]. Available: https://doi.org/10.1145/3636534.3649372

[60] X. Zhou, B. Ahmed, J. H. Aylor, P. Asare, and H. Alemzadeh, "Hybrid knowledge and data driven synthesis of runtime monitors for cyber-physical systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 1, pp. 12–30, 2023.

[61] C. E. Tuncali, G. Fainekos, H. Ito, and J. Kapinski, "Simulation-Based Adversarial Test Generation for Autonomous Vehicles with Machine Learning Components," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1555–1562.

[62] 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.

[63] R. Ben Abdessalem, S. Nejati, L. C. Briand, and T. Stifter, "Testing Advanced Driver Assistance Systems using Multi-Objective Search and Neural Networks," in *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*, 2016, pp. 63–74.

[64] R. B. Abdessalem, S. Nejati, L. C. Briand, and T. Stifter, "Testing Vision-Based Control Systems using Learnable Evolutionary Algorithms," in *2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)*. IEEE, 2018, pp. 1016–1026.

[65] A. Gambi, M. Mueller, and G. Fraser, "Automatically Testing Self-Driving Cars with Search-Based Procedural Content Generation," in *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2019, pp. 318–328.

[66] T. Rahman, A. Liu, D. S. Cheema, V. Chirila, D. C. Transport, and C. Canada, "ADAS Reliability against Weather Conditions: Quantification of Performance Robustness." [Online]. Available: https://api.semanticscholar.org/CorpusID:268237053

[67] Z. Wan, K. Swaminathan, P.-Y. Chen, N. Chandramoorthy, and A. Raychowdhury, "Analyzing and Improving Resilience and Robustness of Autonomous Systems," in *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*, ser. ICCAD '22. New York, NY, USA: Association for Computing Machinery, 2022. [Online]. Available: https://doi.org/10.1145/3508352.3561111

[68] J. F. Rojas, P. Patil, A. M. Masterson, T. H. Bradley, A. R. Ekti, and Z. D. Asher, "Automated Vehicle Lane Centering System Requirements Informed by Resilience Engineering and a Solution Using Infrastructure-Based Sensors," *IEEE Access*, vol. 12, pp. 97 605–97 620, 2024.

[69] A. Berdich and B. Groza, "Cyberattacks on Adaptive Cruise Controls and Emergency Braking Systems: Adversary Models, Impact Assessment, and Countermeasures," *IEEE Transactions on Reliability*, vol. 73, no. 2, pp. 1216–1230, 2024.