UNIVERSITY OF CALIFORNIA,
IRVINE

Exploring Novel Security Vulnerabilities and Their Safety Implications in Sensors and Perception for Autonomous Systems

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Computer Science

by

Takami Sato

Dissertation Committee:
Assistant Professor Qi Alfred Chen, Chair
Professor Mohammad Al Faruque
Assistant Professor Habiba Farrukh

2024

© 2024 Takami Sato

DEDICATION

To my parents, for their unwavering support, and to Pashmak, my roommate's cat, whose presence brought me comfort and solace throughout my Ph.D. journey.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my advisor, Prof. Qi Alfred Chen. His unwavering guidance, continuous support, and boundless encouragement have been instrumental throughout my Ph.D. journey. I distinctly remember our initial conversation, where I was captivated by his visionary perspective on Autonomous Driving security research. That pivotal moment led to my decision to pivot my research direction and join his lab. Throughout the years, his open-mindedness, especially in accepting a Ph.D. student with little background in security and ill English proficiency, has been truly inspiring. This attribute has profoundly impacted both my research and my personal development. I feel incredibly fortunate to have had Prof. Chen as my advisor.

I also wish to extend my heartfelt thanks to the members of my candidacy advancement and dissertation committee. Prof. Gene Tsudik, Prof. Sameer Singh, Prof. Sangeetha Abdu Jyothi, and Prof. Zhou Li offered invaluable feedback and insights during my Ph.D. advancement to candidacy. For my final defense, I am deeply grateful to Prof. Mohammad Al Faruque and Prof. Habiba Farrukh for their constructive suggestions.

Special appreciation goes to my collaborators and mentors who played significant roles in my research projects. Prof. Kentaro Yoshioka at Keio University, along with his brilliant students Yuki Hayakawa, Ryo Suzuki, Kazuma Ikeda, Ozora Sako, Rokuto Nagata, and Ryo Yoshida, have been tremendous collaborators. I also extend my gratitude to Prof. Qi Zhu at Northeastern University and his dedicated students Ruochen Jiao, Xiangguo Liu, Hengyi Liang, and Xiaowei Yuan for their collaborative efforts. I am equally thankful to Prof. Tatsuya Mori at Waseda University and his students Go Tsuruoka, Kazuki Nomoto, Ryunosuke Kobayashi, and Yuna Tanaka for their invaluable contributions. Further acknowledgment goes to Prof. Sara Rampazzi at the University of Florida and her student Sri Hrushikesh Varma Bhupathiraju, Prof. Kaidi Xu at Drexel University, Prof. Takeshi Sugawara at The University of Electro-Communications, and Dr. Michael Clifford from Toyota who mentored me during my internship. Their insights and support were crucial to my research progress. I am also very grateful to Dr. Yueqiang Cheng and Dr. Yulong Cao from NIO, who not only collaborated with me but also provided mentorship during my internships. A significant thanks to Prof. Kazuhide Nakata, my prior advisor during my time at Tokyo Tech, and his students Ryota Ueta and Yukina Nakagawa. I also appreciate the collaboration with Prof. Ken Kobayashi from Tokyo Tech.

My labmates Ningfei Wang, Ziwen Wan, Junjie Shen, Fayzah Alshammari, Shaoyuan Xie, Trishna Chakraborty, Jong Ho Lee, Jun Yeon Won, Zeyuan Chen, and Yunpeng Luo have been invaluable companions throughout this journey. The cherished memories in research, life, and beyond are treasures I deeply value. I extend special thanks to my close friends Shusuke Okita, Ryozo Masukawa, Takashi Nagata, and Yoshimichi Nakatsuka, whose support and camaraderie have been steadfast throughout my Ph.D. journey. Finally, I am eternally grateful to my family for their unwavering support, sacrifices, and blessings that have enabled me to pursue my dreams. This dissertation is dedicated to all those who have been a part of this remarkable journey. Thank you to everyone who made this journey possible.

ABSTRACT OF THE DISSERTATION

Exploring Novel Security Vulnerabilities and Their Safety Implications in Sensors and Perception for Autonomous Systems

By

Takami Sato

Doctor of Philosophy in Computer Science

University of California, Irvine, 2024

Assistant Professor Qi Alfred Chen, Chair

Autonomous systems, particularly autonomous driving (AD), are rapidly developing in our society, driven by significant progress in perception systems with advanced deep neural networks (DNN) and sensors. Self-driving taxi services and robot delivery services are becoming increasingly common in major cities around the world. Despite their high perception capability in various challenging driving scenarios, it does not always mean high robustness in adversarial scenarios. DNN models, in particular, are known to be vulnerable to adversarial input perturbations, which can significantly affect the model output with only minor changes to the input data. Given the critical importance of road safety in AD vehicles, it is crucial to systematically understand the potential security vulnerabilities of DNN models and sensors in AD systems. My dissertation focuses on exploring these vulnerabilities and their safety implications in autonomous systems, specifically targeting two major AD sensors: cameras and LiDARs. Cameras are widely used in AD perception due to their affordability and the reliance on visual information in human driving, such as lane lines and traffic signs. However, camera inputs can be easily perturbed by various attack vectors, including physical stickers and laser or light reflections. LiDAR is one of the most innovative sensors in the past decade and plays a crucial role in higher automation levels such as Level-4 AD due to their ability to capture 3D surrounding information as point clouds. Nevertheless, LiDAR is

fundamentally vulnerable to malicious lasers emitted by adversaries. In this dissertation, I present new practical attacks for these two types of sensors, evaluate the impact of these attacks, and explore potential defenses. By systematically analyzing the security implications at the DNN model level and the closed-loop autonomous driving level, my research aims to contribute to the development of more secure and safer autonomous systems.

Chapter 1

Introduction

Our society is witnessing the rapid integration of autonomous systems into daily life such as Autonomous Driving (AD), which is nowadays a common sight in many cities around the world [36, 9]. For instance, Waymo's robotaxi services are available in several cities, such as Phoenix, San Francisco, and Los Angeles [69]. While the convenience of AD technologies, their rapid deployment highly motivates extensive research efforts to ensure their security due to the potentially fatal safety implications. If the AD system makes incorrect steering decisions, human passengers might not have sufficient reaction time or even means to prevent safety hazards such as driving off-road or colliding with vehicles in adjacent lanes.

Among the security research interests in AD systems, there are two major interests that substitute human roles: the security of perception sensors substituting for human sensory organs and the security of deep neural networks (DNNs) substituting for human brain. DNN has been playing an essential role in enabling the recent significant capabilities of AD systems under accurate environmental sensing from advanced sensors such as cameras or LiDARs. However, DNN models are also known to be vulnerable to subtle malicious input perturbations also called adversarial examples or attacks [316, 178], which can significantly

affect the model output with only subtle changes to the input data. Recent research also demonstrates that such attacks can be realizable in the physical world [228, 110, 122, 168, 373, 128]. If these attacks are actually effective against real-world AD systems, the current AD systems should not be operated considering their security and safety criticality.

Fortunately, none of these prior works have successfully demonstrated end-to-end attack impacts on real AD systems driving on public roads. These prior works mainly concentrate on DNN model-level vulnerability analysis such as image classification and object detection levels, not AD system level. For instance, effective attacks should be effective for a certain duration since compromising one decision frame, typically corresponding to less than 0.1 seconds, is generally too short to cause a meaningful impact; To inject malicious perturbations into sensors, the adversary should have stealthy and remote attack channels, e.g., showing malicious images on a roadside digital signage or emitting malicious lasers from long range. In this dissertation, I do not consider malware or backdoor threat models that can directly inject digital perturbation into DNN inputs since these threat models generally assume strong or even unrealistic attacker capabilities, and are always wondered why an attacker would not use their capabilities to directly compromise the entire system rather than just the DNN model inputs.

To bridge the research gap, my dissertation focuses on revealing the actual threat of these vulnerabilities in AD scenarios, particularly targeting cameras and LiDARs, which are the two major perception sensors of AD vehicles.

Camera-based AD Perception: Camera is one of the most commonly used sensors to enable AD, particularly for lower-level autonomy such as Level-2 AD [287], due to its affordability and the fact that human drivers can drive only with visual information. Automated Lane Centering (ALC) is one of the most common Level-2 AD technologies that can automatically steer a vehicle to keep it centered in the traffic lane [59]. The ALC is widely available on various vehicle models such as Tesla, GM Cadillac, Honda Accord, Toyota RAV4, Volvo

XC90, etc. For models without ALC systems yet, there are also solutions such as OpenPilot [29] that can retrofit them to support ALC after adding a dashcam. Camera also plays a large role in Adaptive Cruise Control (ACC) features in several models, such as Tesla and OpenPilot, which can recognize speed limit signs and stop signs on public roads and control their speed to keep the speed limit or stop at the stop signs.

Compared to LiDAR, cameras offer the advantage of providing high-resolution information about the environment. While LiDAR can understand the 3D shape of objects, it struggles to perceive patterns on objects such as traffic signs or lane markings, which provide critical information for driving on public roads. Meanwhile, the advantage of cameras also benefits the adversaries, i.e., they can easily manipulate camera inputs with various measures such as malicious stickers [167, 245, 208] and light projection [261, 249, 338]

LiDAR-based AD Perception: LiDAR (Light Detection And Ranging) is one of the most innovative sensors in the past decade. By shooting a laser pulse and measuring its reflection, LiDAR can provide a detailed 3D understanding of the surrounding environment as point clouds. After LiDAR showed its high effectiveness in the 2007 DARPA Urban Challenge[335], it has been widely recognized as an essential sensor for Level-4 AD [287] to enable accurate 3D object detection and centimeter-level localization with the HD map [60]. LiDAR has been adopted in almost all recent robotaxi services (Waymo One [69], Cruise [9]) and AD vehicles operating in the US [26, 28]. While manipulating LiDAR sensing is not as straightforward as it is for cameras, the adversaries can still inject or remove points into the sensed point clouds by shooting malicious lasers against LiDARs, which is known as LiDAR spoofing attacks [270, 306, 130, 312, 129, 185, 125].

Research Goals: Given the critical role of cameras and LiDARs as major sensors to enable AD, my research is devoted to exploring the vulnerabilities and their safety implications of cameras and LiDARs in AD contexts. Adversarial attacks and LiDAR spoofing attacks have been shown to be highly effective in compromising DNN models, but no prior research has

clearly demonstrated the end-to-end attack impact and safety implications of these attacks on actual driving AD vehicles because of the following research challenges:

Lack of Stealthy and Practical Attacks against Camera: Adversarial attacks [316, 178] were originally implemented by noise-level perturbations and thus not designed to be realizable in the physical world. However, recent studies [167, 373, 249, 208, 376, 164, 261] have demonstrated that adversarial attacks do not need to be a noise-level perturbation, but realizable with physical changes such as projecting shadows [376], projecting visible colored patterns [249, 164, 261, 367], and adding stickers to traffic signs [167, 373, 208]. Meanwhile, these prior attacks have a clear limitation in stealthiness. Stickers, strong light projections, or shadows inconsistent with the environment are visible to humans, who can detect and mitigate them. For example, in semi-autonomous vehicles, such as Tesla's [86], drivers are required to stay alert and ready to take manual control of the vehicle if needed. As long as visual changes are inevitable, humans may notice them, even if they are subtle. Furthermore, these attacks have generally targeted only traffic sign recognition (TSR), which can only alter the longitudinal control of AD such as accelerating and desecrating AD according to compromised speed limit signs and permanently or temporarily stopping at stop signs. No prior work has demonstrated attacks against lateral control such as attacking ALC systems by compromising its camera-based lane detection.

Lack of Evaluation on Production-Grade LiDARs: Velodyne VLP-16 [48] has been dominantly used in the prior works since it is viewed as a *de facto* choice for LiDAR spoofing evaluation after the first practical spoofing attack was proposed in 2017 [306]. The following works thus all evaluate their attacks only on VLP-16 [130, 312, 185, 125] or use the attack capability on VLP-16 to justify the validity of their threat model [190, 191]. Although these results are valid on VLP-16, there is no guarantee that these results are still valid in more recent LiDARs, known as next-generation or new-generation (or *new-gen*) LiDARs [363], as opposed to the first-generation (or *first-gen*) ones such as VLP-16. The new-gen LiDARs

have more advanced security-related features, such as laser timing randomization and pulse fingerprinting-based authentication. Prior works [130, 125, 306] partially discussed some of them as potential defenses, but none of them experimentally studied their impact and effectiveness against LiDAR spoofing.

Assuming Unvalidated Attack Capabilities in LiDAR Spoofing Attacks: So far, all prior works on fake object injection [130, 312, 185] assume a *Chosen Pattern Injection (CPI)* attack capability, i.e., an attacker can spoof a specifically chosen point cloud pattern that was carefully optimized/identified offline beforehand. For example, the Adv-LiDAR attack [130] is a white-box adversarial attack that needs a specific pattern to trigger a vulnerability in deep neural networks (DNNs). The black-box attacks in [312, 185] also require a specific pattern in the shape of a vehicle surface. However, none of them have systematically studied how achievable this is in practice.

Lack of Practical LiDAR Spoofing Devices Capable in Practical Attack Scenarios: None of the prior works have demonstrated their attacks in realistic environments. Most of them have only been evaluated in indoor lab-level or outdoor setups where AD vehicles are driving slowly (<5km/h), such as parking lots. To deploy LiDAR spoofing attacks on real roads, the hardware requirements will be significantly higher to enable high-speed and long-range attacks. For example, when spoofing at long distances such as 100m, even small distortions in optics, errors in detection results, and delays in motor control could lead to an error of a few meters in aiming. Furthermore, precise and long-range detection and tracking systems are essential to keep LiDAR spoof attacks effective on moving AD vehicles, especially for removal attacks, which need to be effective for at least several consecutive frames. The only previous attempt is camera-based detection and tracking with a generic pan-tilt system. However, this system has not been evaluated in practical scenarios at long distances.

My dissertation research is motivated to address the research challenges. In essence, My dissertation discovers that **sensors and AI models critical for autonomous systems**

(e.g., in both camera- and LiDAR-perception) can be fundamentally vulnerable to physical-world attack vectors and end-to-end exploitable in real-world autonomous system context. Furthermore, physical-world information can be extracted in a cost-effective manner (e.g., from public maps and GNSS-based localization) to defend against perception attacks in autonomous systems.

1.1 Dissertation Organization

This dissertation is structured as follows: Chapter 2 describes the background and related work for my dissertation research. Based on the introductory information, I first describe the security analysis of the camera-based perception. In Chapter 3, I explore the vulnerability in the production-level camera-based ALC system and its lane detection model. In Chapter 4, I introduce an attack detection method for camera-based ALC systems by cross-checking road shapes in publicly available maps and detected lane lines. In Chapter 5, I describe that the current de facto evaluation metrics of lane detection have not been adequately researched, and propose 2 new driving-oriented metrics that can address the limitation to reflect the usability of lane detection models in AD more correctly. In Chapter 6, we propose a new attack vector, infrared laser reflection (ILR), against camera-based TSR systems. In Chapter 7, I describe the intriguing properties of diffusion models and show that the properties can be used to fool DNN models by intentionally removing essential features of the human visual system (HVS), through simple text prompts. I then describe the security analysis of the LiDAR-based perception. In Chapter 8, we conduct the first large-scale measurement study on LiDAR spoofing attack capabilities against object detection to evaluate the actual threats of LiDAR spoofing attacks on the recent line of LiDARs. In Chapter 9, we are not only the first to demonstrate LiDAR spoofing attacks against practical AD scenarios where the victim vehicle is driving at high speeds (60 km/h) and the attack is launched from long

distances (110 meters) but also the first to perform LiDAR spoofing attacks against a vehicle actually operated by a popular AD stack. In Chapter 10, we conclude this dissertation and discuss potential future directions.

Chapter 2

Background and Related Work

2.1 Autonomous Driving System Designs

Fig. 2.1 shows the overview of AD system designs and the roles of AD components. The physical world is perceived by multiple sensors such as camera, LiDAR, radar, GPS, and IMU. Based on the sensor inputs, the typical AD systems make the final control decisions (e.g., steering and throttle) with the three main modules: perception, prediction, and planning [300]. While several end-to-end DNN approaches make the final control decisions directly from the sensor inputs, the end-to-end designs are only for demonstration purposes [368] and thus the major AD systems, such as Apollo [7], Autoware [219], Waymo [344], and Cruise [153], adopt the modular design. For each module, *Perception* is to perceive the surrounding environments and extract the semantic information for driving, such as road object (e.g. pedestrian, vehicles, obstacles) detection, object tracking, segmentation, lane/traffic light detection. Perception module usually takes diverse sensor data as the inputs, including cameras and LiDARs. *Prediction* aims to estimate the future status (position, heading, velocity, acceleration, etc.) of surrounding objects. *Planning* aims to

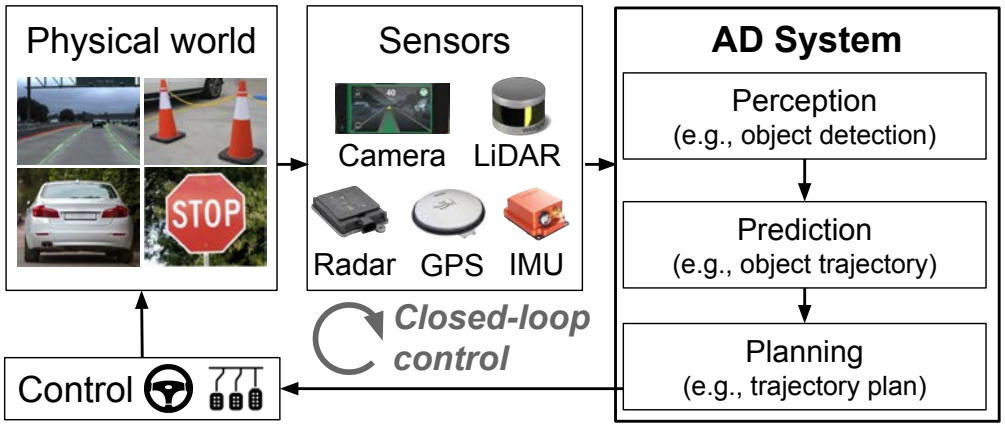


Figure 2.1: Overview of AD system designs and the roles of AD components.

make trajectory-level driving decisions (e.g., cruising, stopping, lane changing, etc.) that are safe and correct (e.g., conforming to traffic rules).

2.2 Physical-World Adversarial Attacks in Autonomous Driving Contexts

Deep Neural Network (DNN) models today are shown to be generally vulnerable to adversarial examples (or adversarial attacks) [316, 178], which can significantly affect the model output with only subtle perturbations to the input data. Prior research demonstrates that such malicious perturbations are realizable even in the physical world [228, 298, 110, 122, 141, 130] via a wide variety of sensors such as LiDAR [270, 306, 130], camera [270, 353, 373, 208], radar [353], ultrasonic [353], IMU [333, 332], etc. Although these prior attacks have been shown to be highly effective in their lab-level evaluation, no prior research has clearly demonstrated the end-to-end attack impact and safety implications of these attacks on actual driving AD vehicles. My dissertation research aims to fill the research gap.

2.3 LiDAR Basics and New-Gen LiDARs

LiDAR is an active sensor that can obtain high-resolution 3D point cloud data of the surrounding environment. The most common type of LiDAR today is direct time-of-flight (dToF) LiDAR, which fires laser pulses to the environment and uses their reflections to actively measure (or "scan") the surrounding objects. Specifically, for each laser pulse, it uses the time difference between the firing and reflection to obtain the position of a "point" on the object surface. By firing the laser at different horizontal angles (*azimuth*) and vertical angles (*altitude*), the point measurements thus form a "point cloud". Such LiDARs typically use laser peak power ($\sim$10 W) that is significantly higher than sunlight even after a long flight, allowing them to detect objects $\geq$200 meters away even under high ambient light.

New-Gen LiDARs: Early Velodyne LiDARs such as VLP-16 [48] and VLP-32c [46] are commonly known as the first-generation (*first-gen*) LiDARs [363], which naively integrate the classic point-wise laser ranging systems. The advent of first-gen LiDARs has significantly boosted critical real-world applications such as autonomous driving (AD), but its complex mechanical design increases costs and limits scalability. To overcome the limitations, the new-generation (*new-gen*) LiDARs [363] mount all components, such as the photodetector and the readout circuitry, on a single chip (called a system-on-chip (SoC) approach). This not only reduces the cost and improves the scalability of the system, but also allows new designs of laser firing and receiving. For example, microelectromechanical systems (MEMS) LiDAR [337] move a mirror to scan a wide azimuth range instead of mechanical rotation. Flash LiDARs [286, 33] fire a broad laser that covers the entire field of view (FOV) and calculates the distance by compensating its weak return laser by accumulation. Moreover, the SoC approach allows more complex signal processing, such as a large number of simultaneous laser firings [10], laser timing randomization [10, 37], and fingerprinting [49], to be robust against challenging environments, e.g., multiple LiDARs operating adjacent to each other.

In summary, the new-gen LiDARs substantially improved the electronics and the scanning mechanisms even though they still have the same basic design: laser firing and receiving. However, none of the prior works on LiDAR spoofing attacks have studied the security property of such new-gen LiDARs; *all* of them are performed on only the first-gen rotational ones, with a predominate focus on in fact *only 1 specific LiDAR model*: VLP-16 [306, 130, 312, 185, 125]. The major design differences between first- and new-gen LiDARs should cause significant differences in their security characteristics, which motivates this study.

2.4 LiDAR Spoofing Attacks

LiDAR spoofing attacks [270, 306, 130, 312, 185, 213, 290, 125, 129] compromise the distance measurements of LiDAR sensors by overwriting legitimate laser signals with higher-power malicious lasers. Fig. 2.2 illustrates how the LiDAR spoofing attacks with timing synchronization work against VLP-16. The attack mechanism is common on both the injection attacks [306, 130, 312, 185] and the removal attacks [190, 125] since their difference is whether they move points at target locations or move points into undetectable area. The attack procedure consists of 3 steps: ① PD first receive the legitimate laser from the target LiDAR to know when the LiDAR will scan the point that the attacker want to change; ② FG plans when to fire lasers based on the information from PD and the pre-defined scan pattern of the target LiDAR; ③ the laser is fired based on the plan from FG through the gate driver and LD. Therefore, to achieve the CPI attack capabilities, the attacker must know exactly where LiDAR is scanning and the scan schedule must be predefined or predictable. The timing randomization breaks the assumption by randomizing the scan schedule.

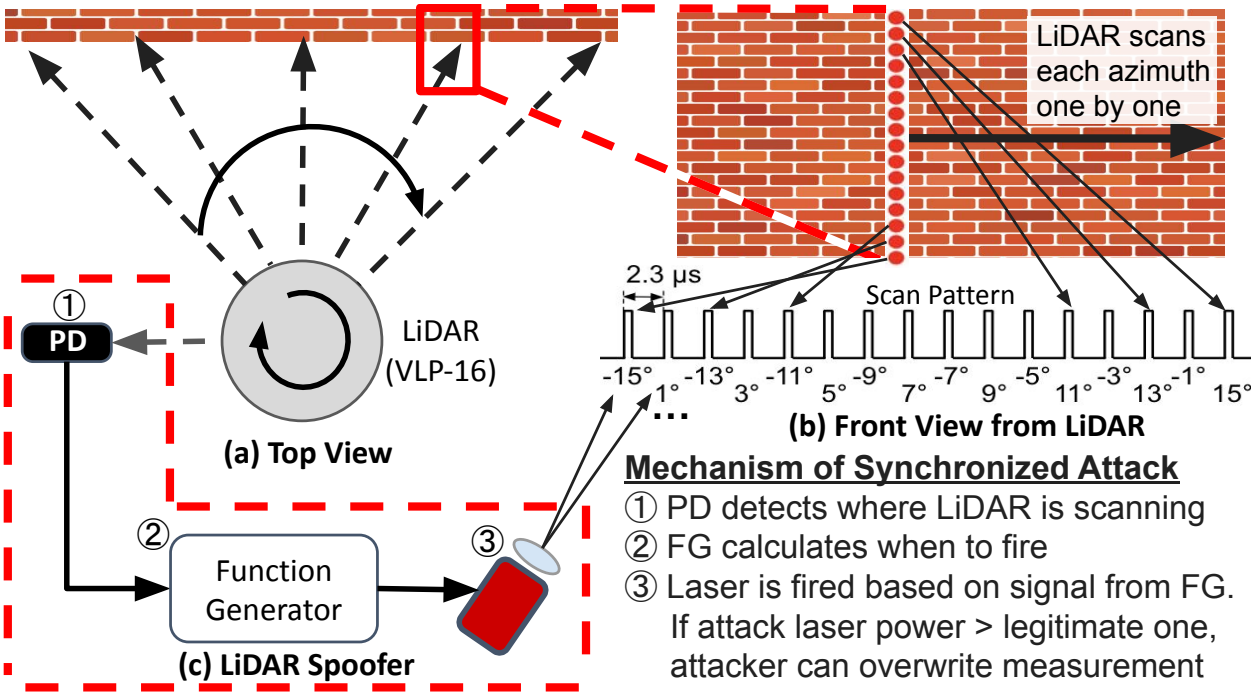


Figure 2.2: Illustration of synchronized attacks on VLP-16. VLP-16 scans each azimuth (every 0.1°) one by one. At an azimuth, it fires 16 lasers vertically based on the pre-defined scan pattern. Once attackers can identify its state by PD, they can know *when* to fire a malicious laser based on the pattern.

Chapter 3

Security of Deep Learning based Automated Lane Centering under Physical-World Attack

3.1 Introduction

Automated Lane Centering (ALC) is a Level-2 driving automation technology that automatically steers a vehicle to keep it centered in the traffic lane [59]. Due to its high convenience for human drivers, today it is widely available on various vehicle models such as Tesla, GM Cadillac, Honda Accord, Toyota RAV4, Volvo XC90, etc. While convenient, such system is highly security and safety critical: When the ALC system starts to make wrong steering decisions, the human driver may not have enough reaction time to prevent safety hazards such as driving off road or colliding into vehicles in adjacent lanes. Thus, it is imperative and urgent to understand the security property of ALC systems.

In an ALC system, the most critical step is *lane detection*, which is generally performed

using a front camera. So far, Deep Neural Network (DNN) based lane detection achieves the highest accuracy [62] and is adopted in the most performant production ALC systems today such as Tesla Autopilot [41]. Recent works show that DNNs are vulnerable to physical-world adversarial attacks such as malicious stickers on traffic signs [168, 373]. However, these methods cannot be directly applied to attack ALC systems due to two main design challenges. First, in ALC systems, the physical-world attack generation needs to handle *inter-dependencies* among camera frames due to attack-influenced vehicle actuation. For example, if the attack deviates the detected lane to the right in a frame, the ALC system will steer the vehicle to the right accordingly. This causes the following frames to capture road areas more to the right, and thus directly affect their attack generation. Second, the optimization objective function designs in prior works are mainly for image classification or object detection models and thus aim at changing class or bounding box probabilities [168, 373]. However, attacking lane detection requires to change the *shape* of the detected traffic lane, making it difficult to directly apply prior designs.

The only prior effort that studied adversarial attacks on a production ALC is from Tencent [74], which fooled Tesla Autopilot to follow fake lane lines created by white stickers on road regions *without lane lines*. However, it is neither attacking the designed operational domain for ALC, i.e., roads *with lane lines*, nor generating the perturbations systematically by addressing the design challenges above.

To fill this critical research gap, in this work we are the first to systematically study the security of DNN-based ALC systems in their designed operational domains (i.e., roads with lane lines) under physical-world adversarial attacks. Since ALC systems assume a fully-attentive human driver prepared to take over at any time [42, 59], we identify the attack goal as not only causing the victim to drive out of the current lane boundaries, but also achieving it shorter than the average driver reaction time to road hazard. This thus directly breaks the design goal of ALC systems and can cause various types of safety hazards such

as driving off road and vehicle collisions.

Targeting this attack goal, we design a novel physical-world adversarial attack method on ALC systems, called *DRP (Dirty Road Patch) attack*, which is the first to systematically address the design challenges above. First, we identify *dirty road patches* as a novel and domain-specific attack vector for physical-world adversarial attacks on ALC systems. This design has 2 unique advantages: (1) Road patches can appear to be legitimately deployed on traffic lanes in the physical world, e.g., for fixing road cracks; and (2) Since it is common for real-world roads to have dirt or white stains, using similar dirty patterns as the input permutations can allow the malicious road patch to appear more normal and thus stealthier.

With this attack vector, we then design systematic malicious road patch generation following an optimization-based approach. To efficiently and effectively address the first design challenge without heavyweight road testing or simulations, we design a novel method that combines vehicle motion model and perspective transformation to dynamically synthesize camera frame updates according to attack-influenced vehicle control. Next, to address the second design challenge, one direct solution is to design the objective function to directly change the steering angle decisions. However, we find that the lateral control step in ALC that calculates steering angle decisions are generally not differentiable, which makes it difficult to effectively optimize. To address this, we design a novel lane-bending objective function as a differentiable surrogate function. We also have domain-specific designs for attack robustness, stealthiness, and physical-world realizability.

We evaluate our attack method on a production ALC system in OpenPilot [29], which is reported to have close performance to Tesla Autopilot and GM Super Cruise, and better than many others [81, 87, 64]. We perform experiments on 80 attack scenarios from real-world driving traces, and find that our attack is highly effective with over 97.5% success rates for all scenarios, and less than 0.903 sec average success time, which is substantially lower than 2.5 sec, the average driver reaction time (§3.3.1). This means that even for a fully-attentive

driver who can take over as soon as the attack starts to take effect, the average reaction time is still not enough to prevent the damage. We further find this attack is (1) robust to real-world factors such as different lighting conditions, viewing angles, printer color fidelity, and camera sensing capability, (2) general to different lane detection model designs, and (3) stealthy from the driver's view based on a user study.

To understand the potential safety impacts, we further conduct experiments using (1) software-in-the-loop simulation in a production-grade simulator, and (2) attack trace injection in a real vehicle. The simulation results show that our attack can successfully cause a victim running a production ALC to hit the highway concrete barrier or a truck in the opposite direction with 100% success rates. The real-vehicle experiments show that it causes the vehicle to collide with (dummy) road obstacles in all 10 trials even with common safety features such as Automatic Emergency Braking (AEB) enabled. Demo videos are available at: **https://sites.google.com/view/cav-sec/drp-attack/**. We also explore and discuss possible defenses at DNN level and those based on sensor/data fusion.

In summary, this work makes the following contributions:

- We are the first to systematically study the security of DNN-based ALC systems in the designed operational domains under physical-world adversarial attacks. We formulate the problem with a safety-critical attack goal, and a novel and domain-specific attack vector, dirty road patches.
- To systematically generate attack patches, we adopt an optimization-based approach with 2 major novel and domain specific designs: motion model based input generation, and lane-bending objective function. We also have domain-specific designs for improving the attack robustness, stealthiness, and physical-world realizability.
- We perform evaluation on a production ALC using 80 attack scenarios from real-world driving traces. The results show that our attack is highly effective with $\geq$97.5% success

rates and $\leq$0.9 seconds average success time, which is substantially lower than the average driver reaction time. This attack is also (1) robust to various real-world factors, (2) general to different lane detection model designs, and (3) stealthy from the driver's view.

- To understand the safety impacts, we conduct experiments using (1) software-in-the-loop simulation, and (2) attack trace injection in a real vehicle. The results show that our attack can cause a 100% collision rate in different scenarios, including when tested with safety features such as AEB. We also evaluate and discuss possible defenses.

Code and data release. Our code and data for the attack and evaluations are available at our project website [11].

3.2 Background

3.2.1 Overview of DNN-based ALC Systems

Fig. 3.1 shows an overview of typical ALC system [82, 231, 29], which operates in 3 steps:

Lane Detection (LD). Lane detection (LD) is the most critical step in an ALC system, since the driving decisions later are mainly made based on its output. Today, production ALC systems predominately use front cameras for this step [40, 41]. On the camera frames, an LD model is used to detect lane lines. Recently, DNN-based LD models achieve the state-of-the-art accuracy [342, 267, 223] and thus are adopted in the most performant production ALC systems today such as Tesla Autopilot [41]. Since lane line shapes do not change much across consecutive frames, recurrent DNN structure (e.g., RNN) is widely adopted in LD models to achieve more stable prediction [236, 381, 29]. LD models typically first predict the lane line points, and then post-process them to lane line curves using curve fitting algorithms [342, 307, 267, 172].

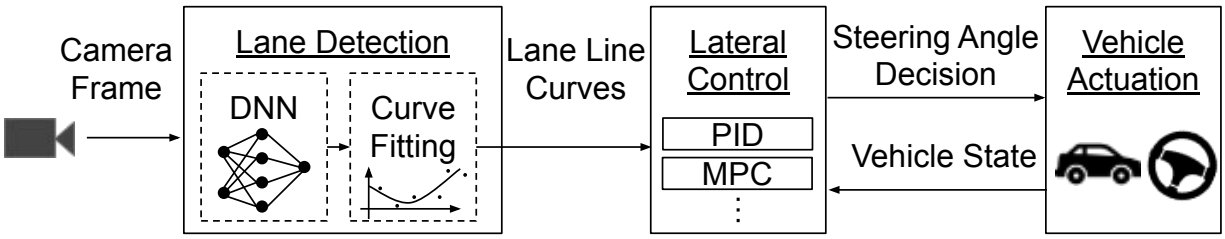


Figure 3.1: Overview of the typical ALC system design.

Before the LD model is applied, a Region of Interest (ROI) filtering is usually performed to the raw camera frame to crop the most important area out of it (i.e., the road surface with lane lines) as the model input. Such ROI area is typically around the center and much smaller than the original frame, to improve the model performance and accuracy [360].

Lateral control. This step calculates *steering angle decisions* to keep the vehicle driving at the center of the detected lane. It first computes a desired driving path, typically at the center of the detected left and right lane lines [115]. Next, a control loop mechanism, e.g., Proportional-Integral-Derivative (PID) [161] or Model Predictive Control (MPC) [284], is applied to calculate the optimal steering angle decisions that can follow the desired driving path as much as possible considering the vehicle state and physical constraints.

Vehicle actuation. This step interprets the steering angle decision into actuation commands in the form of *steering angle changes*. Here, such actuated changes are limited by a maximum value due to the physical constraints of the mechanical control units and also for driving stability and safety [115]. For example, in our experiments with a production ALC with 100 Hz control frequency, such limit is 0.25° per control step (every 10 ms) for vehicle models [43]. As detailed later in §3.3.3, such a steering limit prevents ALC systems from being affected too much from successful attack in one single LD frame, which introduces a unique challenge to our design.

3.2.2 Physical-World Adversarial Attacks

Recent works find that DNN models are generally vulnerable to adversarial examples, or adversarial attacks [316, 178]. Some works further explored such attacks in the physical world [228, 110, 122, 168, 373, 128]. While these prior works concentrate on DNN models for image classification and object detection tasks, we are the first to systematically study such attacks on production DNN-based ALC systems, which requires to address several new and unique design challenges as detailed later in §3.3.3.

3.3 Attack Formulation and Challenge

3.3.1 Attack Goal and Incentives

In this paper, we consider an attack goal that directly breaks the design goal of ALC systems: causing the victim vehicle a lateral deviation (i.e., deviating to the left or right) large enough to drive out of the current lane boundaries. Meanwhile, since ALC systems assume a fully-attentive human driver who is prepared to take over at any moment [42, 59], such deviation needs to be achieved fast enough so that the human driver cannot react in time to take over and steer back. Table 3.1 shows concrete values of these two requirements for successful attacks on highway and local roads respectively, which will be used as evaluation metrics later in §3.5. In the table, the required deviations are calculated based on representative vehicle and lane widths in the U.S., and the required success time is determined using commonly-used average driver reaction time to road hazards, which is detailed in Appendix 3.A.

Free-flow driving. Our study targets the most common driving scenario for using ALC systems: *free-flow* driving scenarios [372], in which a vehicle has at least 5–9 seconds clear headway [119] and thus can drive freely without considering the front vehicle [372].

Table 3.1: Required deviations and success time for successful attacks on ALC systems on highway and local roads. Detailed calculations and explanations are in Appendix 3.A.

Road Type	Required Lateral Deviation	Required Success Time
Highway	0.735 meters	<2.5 seconds (average driver reaction time to road hazard)
Local road	0.285 meters	

Safety implications. The attack goal above can directly cause various safety hazards in the real world: (1) *Driving off road*, which is a direct violation of traffic rules [51] and can cause various safety hazards such as hitting road curbs or falling down the highway cliff. (2) *Vehicle collisions*, e.g., with vehicles parked on the road side, or driving in adjacent or opposite traffic lanes on a local road or a two-lane undivided highway. Even with obstacle or collision avoidance, these collisions are still possible for two reasons. First, today's obstacle and collision avoidance systems are not perfect. For example, a recent study shows that the AEB (Automatic Emergency Braking) in popular vehicle models today fail to avoid crashes 60% of the time [73]. Second, even if they can successfully perform emergency stop, they cannot prevent the victim from being hit by other vehicles that fail to yield on time. Later in §3.7, we evaluate the safety impacts of our attack with a simulator and a real vehicle.

3.3.2 Threat Model

We assume that the attacker can obtain the same ALC system as the one used by the victim to get a full knowledge of its implementation details. This can be done through purchasing or renting the victim vehicle model and reverse engineering it, which has already been demonstrated possible on Tesla Autopilot [74]. Moreover, there exist production ALC systems that are open sourced [29]. We also assume that the attacker can obtain a motion model [277] of the victim vehicle, which will be used in our attack generation process (§3.4.2). This is a realistic assumption since the most widely-used motion model (used by us in §3.4.2) only needs vehicle parameters such as steering ratio and wheelbase as input [277], which can be directly found from vehicle model specifications. We assume the victim drives at the

speed limit of the target road, which is the most common case for free-flow driving. In the attack preparation time, the attacker can collect the ALC inputs (e.g., camera frames) of the target road by driving the victim vehicle model there with the ALC system on.

3.3.3 Design Challenges

Compared to prior works on physical-world adversarial attacks on DNNs, we face 3 unique design challenges:

C1. Lack of legitimately-deployable attack vector in the physical world. To affect the camera input of an ALC system, it is ideal if the malicious perturbations can appear legitimately around traffic lane regions in the physical world. To achieve high legitimacy, such perturbations also must not change the original human-perceived lane information. Prior works use small stickers or graffiti in physical-world adversarial attacks [168, 373, 74]. However, directly performing such activities to traffic lanes in public is illegal [50]. In our problem setting, the attacker needs to operate in the middle of the road when deploying the attack on traffic lanes. Thus, if the attack vector cannot be disguised as legitimate activities, it becomes highly difficult to deploy the attack in practice.

C2. Camera frame inter-dependency due to attack-influenced vehicle actuation. In real-world ALC systems, a successful attack on one single frame can barely cause any meaningful lateral deviations due to the steering angle change limit at the vehicle actuation step (§3.2.1). For example, for the vehicle models with 0.25° angle change limit per control loop (§3.2.1), even if a successful attack on a single frame causes a very large steering angle decision at MPC output (e.g., 90°), it can only cause at most 1.25° steering angle changes before the next frame comes, which can only cause up to *0.3-millimeter* lateral deviations at 45 mph (72 km/h). More detailed explanations are in our extended version [293].

Thus, to achieve our attack goal in §3.3.1, the attack must be *continuously effective on sequential camera frames* to increasingly reach larger actuated steering angles and thus larger lateral deviations per frame. In this process, due to the dynamic vehicle actuation applied by the ALC system, the attack effectiveness for later frames are directly dependent on that for earlier frames. For example, if the attack successfully deviates the detected lane to the right in a frame, the ALC system will steer the vehicle to the right accordingly. This causes the following frames to capture road areas more to the right, and thus directly affect their attack generation. There are prior works considering attack robustness across sequential frames, e.g., using EoT [110, 122] and universal perturbation [237], but none of them consider frame inter-dependencies due to attack-influenced vehicle actuation in our problem setting.

C3. Lack of differentiable objective function design for LD models. To systematically generate adversarial inputs, prior works predominately adopt optimization-based approaches, which have shown both high efficiency and effectiveness [316, 168]. However, the objective function designs in these prior works are mainly for image classification [168, 122] or object detection [168, 373] models, which thus aim at decreasing class or bounding box probabilities. However, as introduced in §3.2.1, LD models output detected lane line curves, and thus to achieve our attack goal the objective function needs to aim at changing the *shape* of such curves. This is substantially different from decreasing probability values, and thus none of these existing designs can directly apply.

Closer to our problem, prior works that attack end-to-end autonomous driving models [269, 328, 146, 378] directly design their objective function to change the final steering angle decisions. However, as described in §3.2.1, state-of-the-art LD models do not directly output steering angle decisions. Instead, they output lane line curves and rely on the lateral control step to compute the final steering angle decisions. However, many steps in the lateral control module, e.g., the desired driving patch calculation and the MPC framework, are generally not differentiable to the LD model input, which makes it difficult to effectively optimize.

3.4 Dirty Road Patch Attack Design

In this paper, we are the first to systematically address the design challenges above by a novel physical-world attack method on ALC, called *Dirty Road Patch (DRP) attack*.

3.4.1 Design Overview

To address the 3 design challenges in §3.3.3, our DRP attack method has the following novel design components:

Dirty road patch: Domain-specific & stealthy physical-world attack vector. To address challenge **C1**, we are the first to identify *dirty road patch* as an attack vector in physical-world adversarial attacks. This design has 2 unique advantages. First, road patches can appear to be legitimately deployed on traffic lanes in the physical world, e.g., for fixing road cracks. Today, deploying them is made easy with adhesive designs [70] as shown in Fig. 3.2. The attacker can thus take time to prepare the attack in house by carefully printing the malicious input perturbations on top of such adhesive road patches, and then pretend to be road workers like those in Fig. 3.2 to quickly deploy it when the target road is the most vacant, e.g., in late night, to avoid drawing too much attention.

Second, since it is common for real-world roads to have dirt or white stains such as those in Fig. 3.2, using similar dirty patterns as the input perturbations can allow the malicious road patch to appear more normal and thus stealthier. To mimic the normal dirty patterns, our design only allows color perturbations on the gray scale, i.e., black-and-white. To avoid changing the lane information as discussed in §3.3.3, in our design we (1) require the original lane lines to appear exactly the same way on the malicious patch, if covered by the patch, and (2) restrict the brightness of the perturbations to be strictly lower than that of the original lane lines. To further improve stealthiness, we also design parameters to adjust the

Figure 3.2: Illustration of our novel and domain-specific attack vector: Dirty Road Patch (DRP).

perturbation size and pattern, which are detailed in §3.4.3.

So far, none of the popular production ALC systems today such as Tesla, GM, etc. [42, 75] identify roads with such dirty road patches as driving scenarios that they do not handle, which can thus further benefit the attack stealthiness.

Motion model based input generation. To address the strong inter-dependencies among the camera frames (**C2**), we need to dynamically update the content of later camera frames according to the vehicle actuation decisions applied at earlier ones in the attack generation process. Since adversarial attack generation typically takes thousands of optimization iterations [134, 252], it is practically highly difficult, if not impossible, to drive real vehicles on the target road to obtain such dynamic frame update in every optimization iteration. Another idea is to use vehicle simulators [19, 162], but it requires the attacker to first create a high-definition 3D scene of the target road in the real world, which requires a significant amount of hardware resource and engineering efforts. Also, launching a vehicle simulator in each optimization iteration can greatly harm the attack generation speed.

To efficiently and effectively address this challenge, we combine *vehicle motion model* [277]

and *perspective transformation* [321] to dynamically synthesize camera frame updates according to a driving trajectory simulated in a lightweight way. This method is inspired by Google Street View that synthesizes 360° views from a limited number of photos utilizing perspective transformation. Our method only requires one trace of the ALC system inputs (i.e., camera frames) from the target road without attack, which can be easily obtained by the attacker (§3.3.2).

Optimization-based DRP generation. To systematically generate effective malicious patches, we adopt an optimization-based approach similar to prior works [316, 168]. To address challenge **C3**, we design a novel lane-bending objective function as a differentiable surrogate that aims at changing the derivatives of the desired driving path before the lateral control module, which is equivalent to change the steering angle decisions at the lateral control design level. Besides this, we also have other domain-specific designs in the optimization problem formulation, e.g., for a differentiable construction of the curve fitting process, malicious road patch robustness, stealthiness, and physical-world realizability.

Fig. 3.3 shows an overview of the malicious road patch generation process, which is detailed in the following sections.

3.4.2 Motion Model based Input Generation

In Fig. 3.3, step ①–⑦ belong to the motion model based input generation component. As described earlier in §3.4.1, the input to this component is a trace of ALC system inputs such as camera frames from driving on the target road without attack. In ①, we apply *perspective transformation*, a widely-used computer vision technique that can project an image view from a 3D coordinate system to a 2D plane [188, 321]. Specifically, we apply it to the original camera frames from the driver's view to obtain their Bird's Eye View (BEV) images. This transformation is highly beneficial since it makes our later patch placement

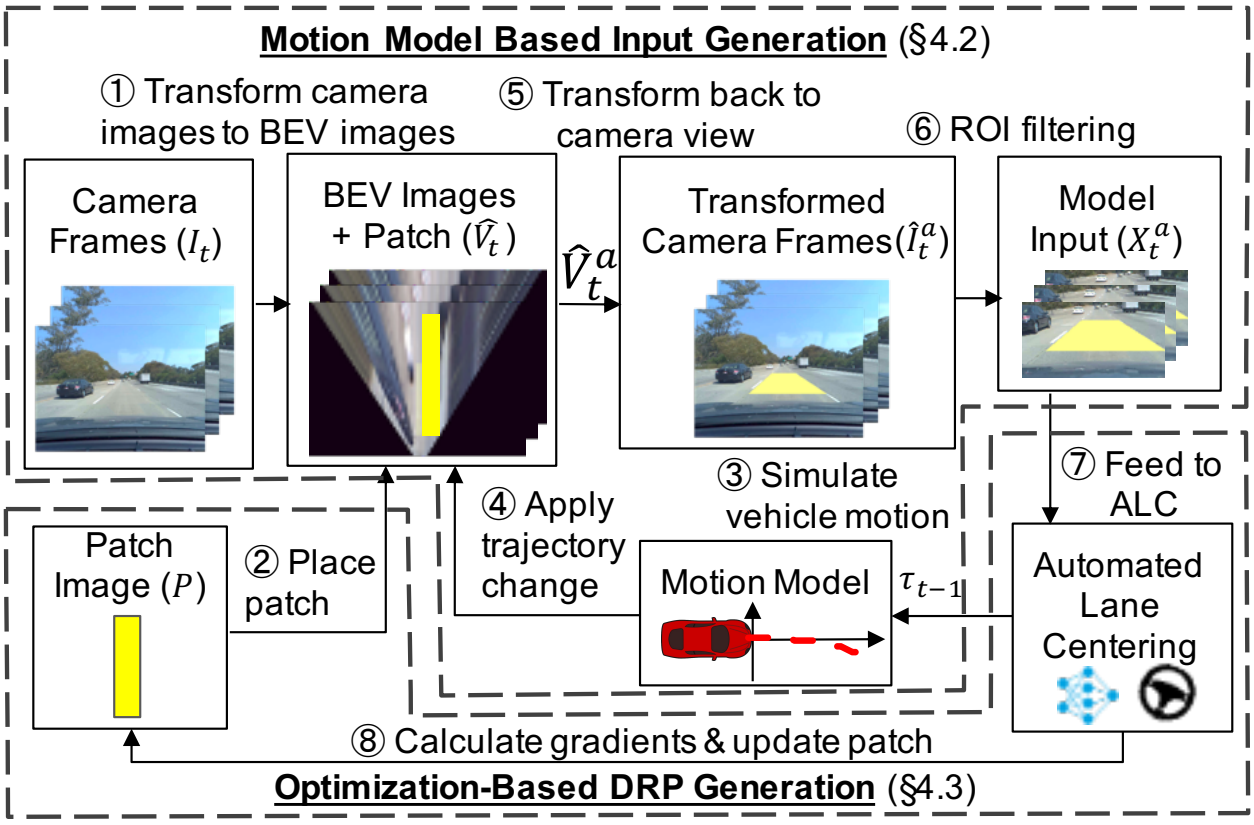


Figure 3.3: Overview of our DRP (Dirty Road Patch) attack method. ROI: Region of Interest; BEV: Bird's Eye View.

and attack-influenced camera frame updates much more natural and thus convenient. We denote this as $V_t := \text{BEV}(I_t)$, where I_t and V_t are the original camera input and its BEV view respectively at frame t. This process is inversible, i.e., we can also obtain I_t with $\text{BEV}^{-1}(V_t)$.

Next, in ②, we obtain the generated malicious road patch image P from the optimization-based DRP generation step (§3.4.3) and place it on V_t to obtain the BEV image with the patch, denoted as $\widehat{V}_t := \Lambda(V_t, P)$. To achieve consistent patch placements in the world coordinate across frames, we calculate the *pixel-meter relationship*, i.e., the number of pixels per meter, in BEV based on the driving trace of the target road. With this, we can place the patch in each frame precisely based on the driving trajectory changes across frames.

Next, we compute the vehicle moving trajectory changes caused by the placed malicious road patch, and reflect such changes in the camera frames. We represent the vehicle moving

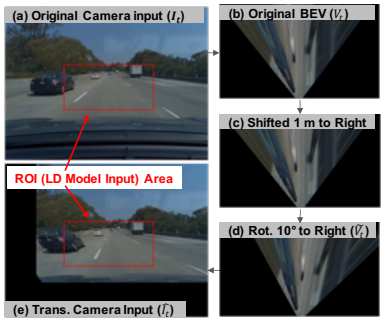


Figure 3.4: Motion model based input generation from original camera input.

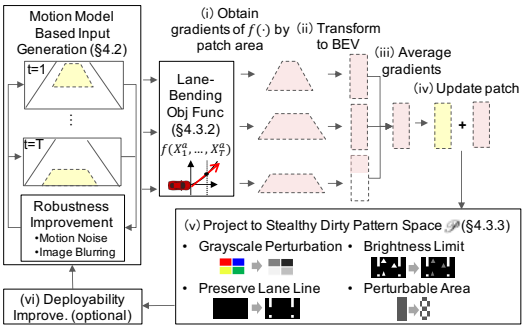


Figure 3.5: Iterative optimization process design for our optimization-based DRP generation.

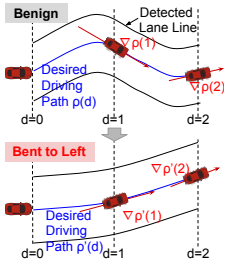


Figure 3.6: "Lane bending" effect of our objective function by maximizing $\nabla\rho(d)$ at each curve point.

trajectory as a sequence of vehicle states $S_t := [x_t, y_t, \beta_t, v_t], (t = 1, ..., T)$, where x_t, y_t, β_t, v_t are the vehicle's 2D position, heading angle, and speed at frame t, and T is the total number of frames in the driving trace. Thus, the trajectory change at frame t is $\delta_t := S_t^a - S_t^o$, where S_t^a and S_t^o are vehicle states with and without attack respectively.

To calculate δ_t caused by the attack effect at the frame $t-1$, we need to know the attack-influenced vehicle state S_t^a. To achieve that, we use a *vehicle motion model* to simulate the vehicle state S_t^a by feeding the steering angle decision τ_{t-1} from the lateral control step in the ALC system (§3.2.1) given the attacked frame at $t-1$ and the previous vehicle state S_{t-1}^a, denoted as $S_t^a := \text{MM}(S_{t-1}^a, \tau_{t-1})$. A vehicle motion model is a set of parameterized mathematical equations representing the vehicle dynamics and can be used to simulate its driving trajectory given the speed and actuation commands. In this process, we set the vehicle speed as the speed limit of the target road as described in our threat model (§3.3.2). In our design, we adopt the kinematic bicycle model [225], which is the most widely-used motion model for vehicles [225, 343].

With δ_t, in ④ we then apply affine transformations on the BEV image $\widehat{V}_t$ to obtain the attack-influenced one $\widehat{V}_t^a$, denoted as $\widehat{V}_t^a := T(\widehat{V}_t, \delta_t)$. Fig. 3.4 shows an example of the shifting and rotation $T(\cdot)$ in the BEV, which synthesizes a camera frame with the vehicle position

shifted by 1 meter and rotated by 10° to the right. Although it causes some distortion and missing areas on the edge, the ROI area (red rectangle), i.e., the LD model input, is still complete and thus sufficient for our purpose. Since the ROI area is typically focused on the center and much smaller than the raw camera frame (§3.2.1), our method can successfully synthesize multiple complete LD model inputs from only 1 ALC system input trace.

Next, in ⑤, we obtain the attack-influenced camera frame at the driver's view $\widehat{I}_t^a$, i.e., the direct input to ALC, by projecting $\widehat{V}_t^a$ back using $\widehat{I}_t^a := \text{BEV}^{-1}(\widehat{V}_t^a)$. Next, in ⑥, the ROI filtering is used to extract the model input $X_t^a := \text{ROI}(\widehat{I}_t^a)$. X_t^a and vehicle state S_t^a are then fed to ALC system in ⑦ to obtain the steering angle decision τ_t, denoted as $\tau_t := \text{ALC}(X_t^a, S_t^a)$. Step ③–⑦ are then iteratively applied to obtain $\widehat{I}_{t+1}^a, \widehat{I}_{t+2}^a, ...$ one after one until all the original frames are updated to reflect the moving trajectory changes caused by P. These updated attack-influenced inputs are then fed to the optimization-based DRP generation component, which is detailed next.

3.4.3 Optimization-Based DRP Generation

In Fig. 3.3, step ⑧ belongs to the optimization-based road path generation component. In this step, we design a domain-specific optimization process on the target ALC system to systematically generate the malicious dirty road patch P.

DRP attack optimization problem formulation. We formulate the attack as the fol-

lowing optimization problem:

$$\min \quad \mathscr{L} \tag{3.1}$$

$$\text{s.t.} \quad X_t^a = \text{ROI}(\text{BEV}^{-1}(T(\Lambda(V_t, P), S_t^a - S_t^o))) \quad (t = 1, ..., T) \tag{3.2}$$

$$\tau_t^a = \text{ALC}(X_t^a, S_t^a) \qquad (t = 1, ..., T) \tag{3.3}$$

$$S_{t+1}^a = \text{MM}(S_t^a, \tau_t^a) + \epsilon_t \qquad (t = 1, ..., T-1) \tag{3.4}$$

$$S_1^a = S_1^o \tag{3.5}$$

$$P = \text{BLUR}(\text{FILL}(B) + \Delta) \tag{3.6}$$

$$\Delta \in \mathscr{P} \tag{3.7}$$

where the $\mathscr{L}$ in Eq. 3.1 is an objective function that aims at deviating the victim out of the current lane boundaries as fast as possible (detailed in §3.4.3). Eq. 3.2–3.5 have been described in §3.4.2. In Eq. 3.6, the patch image $P \in \mathbb{R}^{H \times W \times C}$ consists of a base color $B \in \mathbb{R}^C$ and the perturbation $\Delta \in \mathbb{R}^{H \times W \times C}$, where W, H, and C are the patch image width, height, and the number of color channels respectively. We select an asphalt-like color as the base color B since the image is designed to mimic a road patch. Function FILL: $\mathbb{R}^C \to \mathbb{R}^{H \times W \times C}$ fills B to the entire patch image. Since we aim at generating perturbations that mimic the normal dirty patterns on roads, we restrict Δ to be within a stealthy road pattern space $\mathscr{P}$, which is detailed in §3.4.3. We also include a noise term ϵ_t in Eq. 3.4 and an image blurring function BLUR$(\cdot)$ in Eq. 3.6 to improve the patch robustness to vehicle motion model inaccuracies and camera image blurring, which are detailed in §3.4.3.

Optimization Process Overview

Fig. 3.5 shows an overview of our iterative optimization process design. Given an initial patch image P, we obtain the model input $X_1^a, ..., X_T^a$ from the motion model based input generation process. In step (i), we calculate the gradients of the objective function with respect to $X_1^a, ..., X_T^a$, and only keep the gradients corresponding to the patch areas. In step

(ii), these gradients are projected into the BEV space. In step (iii), we calculate the average BEV-space gradients weighted by their corresponding patch area sizes in the model inputs. This step involves an approximation of the gradient of $\text{BEV}^{-1}(\cdot)$, which are detailed in our extended version [293]. Next, in step (iv), we update the current patch with Adam [222] using the averaged gradient as the gradient of the patch image. In step (v), we then project the updated patch into the stealthy road pattern space $\mathscr{P}$. This updated patch image is then fed back to the motion model based input generation module, where we also add robustness improvement such as motion noises and image blurring. We terminate this process when the attack-introduced lateral deviations obtained from the motion model are large enough.

Lane-Bending Objective Function Design

As discussed in §3.4.1, directly using steering angle decisions as $\mathscr{L}$ makes the objective function non-differentiable to $X_1^a, ..., X_T^a$. To address this, we design a novel lane-bending objective function $f(\cdot)$ as a differentiable surrogate function. In this design, our key insight is that at the design level, the lateral control step aims at making steering angle decisions that follows a *desired driving path* in the middle of the detected left and right lane line curves from the lane detection step (§3.2.1). Thus, changing the steering angle decisions is equivalent to changing the derivatives of (or "bending") such desired driving path curve. This allows us to design $f(\cdot)$ as:

$$f(X_1^a, ..., X_T^a) = \sum_{t=1}^{T} \sum_{d \in D_t} \nabla \rho_t(d; \{X_j^a | j \leq t\}, \theta) + \lambda ||\Omega_t(X_t^a)||_p \tag{3.8}$$

where $\rho_t(d)$ is a parametric curve whose parameters are decided by (1) both the current and previous model inputs $\{X_j^a | j \leq t\}$ due to frame inter-dependencies (§3.3.3), and (2) the LD DNN parameters θ. D_t is a set of curve point index $d = 0, 1, 2, ...$ for the desired driving path

curve at frame t. λ is the weight of the p-norm regularization term, designed for stealthiness (§3.4.3). We then can define $\mathscr{L}$ in Eq. 3.1 as $f(\cdot)$ and $-f(\cdot)$ when attacking to the left and right. Fig. 3.6 illustrates this surrogate function when attacking to the left. As shown, by maximizing $\nabla\rho_t(d)$ at each curve point in Eq. 3.8, we can achieve a "lane bending" effect to the desired driving path curve. Since the direct LD output is lane line points (§3.2.1) but $\rho_t(\cdot)$ require lane line curves, we further perform a differentiable construction of curve fitting process (detailed in our extended version[293]).

Designs for Dirty Patch Stealthiness

To mimic real-world dirty patterns like in Fig. 3.2, we have 4 stealthiness designs in stealthy road pattern space $\mathscr{P}$ in Eq. 3.7:

Grayscale perturbation. Real-world dirty patterns on the road are usually created by dust or white stains (Fig. 3.2), and thus most commonly just appear white. Thus, we cannot allow perturbations with arbitrary colors like prior works [373]. Thus, our design restricts our perturbation Δ in the grayscale (i.e., black-and-white) by only allowing increase the Y channel in the YCbCr color space [186], denoted as $\Delta_Y \geq 0$.

Preserving original lane line information. We preserve the original lane line information by drawing the same lane lines as the original ones on the patch (if covered by the patch). Note that without this our attack can be easier to succeed, but as discussed in §3.3.3, it is much more preferred to preserve such information so that the attack deployment can more easily appear as legitimate road work activities and the deployed patch is less likely to be legitimately removed.

Brightness limits. While the dirty patterns are restricted to grayscale, they are still the darker, the stealthier. Also, to best preserve the original lane information, the brightness of the dirty patterns should not be more than the original lane lines. Thus, we (1) add

the p-norm regularization term in Eq. 3.8 to suppress the amount of Δ_Y, and (2) restrict $B_Y + \Delta_Y < \text{LaneLine}_Y$, where B_Y and LaneLine_Y are Y channel values for the base color and original lane line color respectively.

Perturbation area restriction. Besides brightness, also the fewer patch areas are perturbed, the stealthier. Thus, we define Perturbable Area Ratio (PAR) as the percentage of pixels on P that can be perturbed. Thus, when PAR=30%, 70% pixels on P will only have the base color B.

Designs for Improving Attack Robustness, Deployability, and Physical-World Realizability

We also have domain-specific designs for improving (1) **attack robustness**, which addresses the driving trajectory/angle deviations and camera sensing inaccuracies in real-world attacks; (2) **attack deployability**, which designs an optional *multi-piece patch attack* mode that allows deploying DRP attack with multiple small and quickly-deployable road patch pieces; and (3) **physical-world realizability**, which addresses the color and pattern distortions due to physical-world factors such as lighting condition, printer color accuracy, and camera color sensing capability. More details are in our extended version [293].

3.5 Attack Methodology Evaluation

In this section, we evaluate the effectiveness, robustness, generality, and realizability of our DRP attack methodology.

Targeted ALC system. In our evaluation, we perform experiments on the production ALC system in OpenPilot [29], which follows the state-of-the-art DNN-based ALC system

Figure 3.7: Driver's view at 2.5 sec (average driver reaction time to road hazards [264]) before our attack succeeds under different stealthiness levels in local road scenarios. Inset figures are the zoomed-in views of the malicious road patches. Larger images are in our extended version [293].

Figure 3.8: Real-world dirty road patterns.

Figure 3.9: Stop sign hiding and appearing attacks [373].

design (§3.2.1). OpenPilot is an open-source production Level-2 driving automation system that can be easily installed in over 80 popular vehicle models (e.g., Toyota, Cadillac, etc.) by mounting a dashcam. We select OpenPilot due to its (1) *representativeness*, since it is reported to have close performance to Tesla Autopilot and GM Super Cruise and better than many others [81, 87, 64], (2) *practicality*, from the large quantity and diversity of vehicle models it can support [29], and (3) *ease to experiment with*, since it is the only production ALC system that is open sourced. In this paper, we mainly evaluate on the lane detection model in OpenPilot v0.7.0, which is released in Dec. 2019. More details of the OpenPilot ALC system are in Appendix 3.B.1.

Evaluation dataset. We perform experiments using the comma2k19 dataset [294], which contains over 33 hours driving traces between California's San Jose and San Francisco in a Toyota RAV4 2017 driven by human drivers. These traces are collected using the official OpenPilot dashcam device, called EON. From this dataset, we manually look for short free-flow driving periods to make road patch placement convenient. In total, we obtain 40 eligible short driving clips, 10 seconds each, with half of them on the highway, and half on local roads. For each driving clip, we consider two attack scenarios: attack to the left, and to the right. Thus, in total we evaluate 80 different attack scenarios.

3.5.1 Attack Effectiveness

Evaluation methodology and metrics. We evaluate the attack effectiveness using the evaluation dataset described above. For each attack scenario, we generate an attack road patch, and use the motion model based input generation method in §3.4.2 to simulate the vehicle driving trajectory influenced by the malicious road patch. To judge the attack success, we use the attack goal defined in §3.3.1 and concrete metrics listed in Table 3.1, i.e., achieving over 0.735 meters and 0.285 meters lateral deviations on highway and local road scenarios respectively within the average driver reaction, 2.5 seconds. We measure the achieved deviation by calculating the lateral distances at each time point between the vehicle trajectories with and without the attack, and use the earliest time point to reach the required deviation to calculate the success time.

Since ALC systems assume a human driver who is prepared to take over, it is better if the malicious road patch can also look stealthy enough at 2.5 sec (driver reaction time) before the attack succeeds so that the driver will not be alerted by its looking and decide to take over. Thus, in this section, we also study the stealthiness of the generated road patches. Specifically, we quantify their perturbation degrees using the average pixel value changes from the original road surface in $\mathcal{L}_1, \mathcal{L}_2$ and $\mathcal{L}_{\text{inf}}$ distances [244, 109] and also a user study.

Experimental setup. For each scenario in the evaluation dataset, we manually mark the road patch placement area in the BEV view of each camera frame based on the lane width and shape. To achieve consistent road patch placements in the world coordinate across a sequence of frames, we calculate the number of pixels per meter in the BEV images and adjust the patch position in each frame precisely based on the driving trajectory changes across consecutive frames. The road patch sizes we use are 5.4 m wide, and 24–36 m long to ensure at least a few seconds of visible time at high speed. The patches are placed 7 m far from the victim at the starting frame. For stealthiness levels, we evaluate the $\mathcal{L}_2$

Table 3.2: Attack success rate and time under different stealthiness levels. Larger λ means stealthier. Average success time is calculated only among the successful cases. Pixel $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_{inf}$ are the average pixel value changes from the original road surface in the RGB space and normalized to $[0, 1]$.

Stealth. Level λ	Succ. Rate	Succ. Time (s)	Pixel $\mathcal{L}_1$	Pixel $\mathcal{L}_2$	Pixel $\mathcal{L}_{inf}$
10^{-2}	97.5%	0.903	0.018	0.045	0.201
10^{-3}	100%	0.887	0.033	0.066	0.200
10^{-4}	100%	0.886	0.071	0.109	0.200

regularisation coefficient $\lambda = 10^{-2}, 10^{-3}$, and 10^{-4}, with PAR set to 50%. According to Eq. 3.8, larger λ value means more suppression of the perturbation, and thus should lead to a higher stealthiness level. For the motion model, we use the vehicle parameters (e.g., wheelbase) of Toyota RAV4 2017, the vehicle model that collects the traces in our dataset.

Results. As shown in Table 3.2, our attack has high effectiveness ($\geq$97.5%) under all the 3 stealthiness levels. Fig. 3.7 shows the malicious road patch appearances at different stealthiness levels from the driver's view at 2.5 seconds before our attack succeeds. As shown, even for the lowest stealthiness level ($\lambda = 10^{-4}$) in our experiment, the perturbations are still smaller than some real-world dirty patterns such as the left one in Fig. 3.8. In addition, the perturbations for all these 3 stealthiness levels are a lot less intrusive than those in previous physical-world adversarial attacks in the image space [373], e.g., in Fig. 3.9. Among the successful cases, the average success time is all under 0.91 sec, which is substantially lower than 2.5 sec, the required success time. This means that even for a fully attentive human driver who is always able to take over as soon as the attack starts to take effect, the average reaction time is still far from enough to prevent the damage. A more detailed result discussion is in our extended version [293].

Stealthiness user study. To more rigorously evaluate the attack stealthiness, we conduct a user study with 100 participants, and find that (1) even for the lowest stealthiness level at $\lambda = 10^{-4}$, only *less than 25%* of the participants decide to take over the driving before the attack starts to take effect. This suggests that the majority of human drivers today do not

treat dirty road patches as road conditions where ALC systems cannot handle; and (2) at 2.5 seconds before the attack succeeds, the attack patches with $\lambda = 10^{-2}$ and 10^{-3} appear to be *as innocent as normal clean road patches to human drivers*, with only less than 15% participants deciding to take over. More detailed results and discussion are in Appendix 3.B.

From these results, the stealthiness level with $\lambda = 10^{-3}$ strikes an ideal balance between attack effectiveness and stealthiness: it does not increase driver suspicion compared to even a benign clean road patch at 2.5 seconds before our attack succeeds, while having no sacrifice of attack effectiveness as shown in Table 3.2. We thus use it as the default stealthiness configuration in our following experiments.

3.5.2 Comparison with Baseline Attacks

Evaluation methodology. To understand the benefits of our current design choices over possible alternatives, we evaluate against 2 baseline attack methods: (1) *single-frame EoT attack*, which still uses our lane-bending objective function but optimizes for the EoT (Expectation over Transformation) of the patch view (e.g., different positions/angles) in a single camera frame, and (2) *drawing-lane-line attack*, which directly draws straight solid white lane line instead of placing dirty road patches. EoT is a popular design in prior works to improve attack robustness across sequential frames [110, 122]. Thus, comparing with such a baseline attack can evaluate the benefit of our motion model based input generation design (§3.4.2) in addressing the challenge of frame inter-dependencies due to attack-influenced vehicle actuation (C2 in §3.3.3).

The drawing-lane-line attack is designed to evaluate the type of ALC attack vector identified in the prior work [74], which uses straightly-aligned white stickers to fool Tesla Autopilot on road regions *without lane lines*. In our case, we perform evaluations in road regions *with lane lines*, and use a more powerful form of it (directly drawing solid lane lines) to understand

the upper-bound attack capability of this style of perturbation for ALC systems.

Experimental setup. For single-frame EoT attack, we apply random transformations of the patch in BEV via (1) lateral and longitudinal position shifting. We apply up to $\pm$0.735m and $\pm$0.285m for highway and local respectively, which are their maximum in-lane lateral shifting from the lane center; and (2) viewing angle changes. we apply up to $\pm 5.8^{\circ}$ changes, the largest average angle deviations under possible real-world trajectory variations based on our experiments (detailed in our extended version [293]). For each scenario, we repeat the experiments for each frame with a complete patch view (usually the first 4 frames), and take the most successful one to obtain the upper-bound effectiveness. Other settings are the same as the DRP attack, e.g., $\lambda = 10^{-3}$.

For the drawing-lane-line attack, we use the same perturbation area (i.e., the patch area) as the others for a fair comparison. Specifically, we sample points every 20cm at the top and bottom patch edges respectively, and form possible attacking lane lines by connecting a point at the top with one at the bottom. We exhaustively try all possible top and bottom point combinations and take the most successful one. The attacking lane lines are 10cm wide (a typical lane marking width [83]) with the same white color as the original lane lines.

Results. Table 3.3 shows the results under different patch area lengths. As shown, the DRP attack always has the highest attack success rate than these two baselines (with a $\geq$46% margin). When the patch area length is shorter and thus the perturbation capability is more limited, such advantage becomes larger; when the length is 12m, the success rates of single-frame EoT attack and the drawing-lane-line attack drops to 0% and 2.5%, while that for DRP is still 66%. This shows that our motion model based input generation can indeed benefit attack effectiveness, as it can more accurately synthesize subsequent frame content based on attack-influenced vehicle actuation, instead of the blind synthesis in EoT. Also note that the single-frame EoT attack still uses our domain-specific lane-bending objective function design. The drawing-lane-line attack only has 2.5% success rate when the length

Table 3.3: Attack success rates of the DRP attack and 2 baseline attacks under different patch area lengths.

Attack	Patch Area Length			
	12m	18m	24m	36m
DRP	66.25%	82.50%	90.75%	100%
Single-frame EoT	0.00%	8.75%	21.25%	50.00%
Drawing-lane-line	2.50%	13.75%	31.25%	53.75%

is 12m; the length used in the Tencent work is actually even shorter (<5m) [74]. This shows that in the road regions *with lane lines*, simply adding lane-line-style perturbations, especially a short one, can barely affect production ALC systems. Instead, an attack vector with larger perturbation area, e.g., in DRP attack, may be necessary.

3.5.3 Attack Robustness, Generality, and Deployability Evaluations

Robustness to run-time driving trajectory and angle deviations. As described in §3.4.3, the run-time victim driving trajectories and angles will be different from the motion model predicted ones in attack generation time due to run-time driving dynamics. To evaluate attack robustness against such deviations, we use (1) 4 levels of vehicle position shifting at each vehicle control step in attack evaluation time, and (2) 27 vehicle starting positions to create a wide range of approaching angles and distances to the patch, e.g., from (almost) the leftmost to the rightmost position in the lane. Our attack is shown to maintains a high effectiveness ($\geq 95\%$ success rate) even when the vehicle positions at the attack evaluation time has 1m shifting on average from those at the attack generation time at each control step. Details are in our extended version [293].

Attack generality evaluation. To evaluate the generality of our attack against LD models of different designs, ideally we hope to evaluate on LD models from other production ALC besides OpenPilot, e.g., from Tesla Autopilot. However, OpenPilot is the only one that is

currently open sourced. Fortunately, we find that the LD models in some older versions of OpenPilot actually have different DNN designs, which thus can also serve for our purpose. We evaluate on 3 versions of LD models with large DNN architecture differences, and find that our attack is able to achieve $\geq$90% success rates against all 3 LD models, with an average attack transferability of 63%. More details are in our extended version [293].

Attack deployability evaluation. We evaluate the attack deployability by estimating the required efforts to deploy the attack road patch. We perform experiments using our multi-piece patch attack mode design (§3.4.3), and find that the attack success rate can be *93.8% with only 8 pieces of quickly-deployable road patches*, each requiring only 5-10 sec for 2 people to deploy based on videos of adhesive patch deployment [71]. More details are in our extended version [293].

3.5.4 Physical-World Realizability Evaluation

While we have shown high attack effectiveness, robustness, and generality on real-world driving traces, the experiments are performed by synthesizing the patch appearances digitally, which is thus still different from the patch appearances in the physical world. As discussed in §3.4.3, there are 3 main practical factors that can affect the attack effectiveness in physical world: (1) the lighting condition, (2) printer color accuracy, and (3) camera sensing capability. Thus, in this section we perform experiments to understand the physical-world attack realizability against these 3 main practical factors.

Evaluation methodology: miniature-scale experiments. To perform the DRP attack, a real-world attacker can pretend to be road workers and place the malicious road patch on public roads. However, due to the access limit to private testing facilities, we cannot do so ethically and legally on public roads with a real vehicle. Thus, we try our best to perform such evaluation by designing a *miniature-scale experiment*, where the road and the malicious road

patch are first physically printed out on papers and placed according to the physical-world attack settings but in miniature scale. Then the real ALC system camera device is used to get camera inputs from such a miniature-scale physical-world setting. Such miniature-scale evaluation methodology can capture all the 3 main practical factors in the physical-world attack setting, and thus can sufficiently serve for the purpose of this evaluation.

Experimental setup. As shown in Fig. 3.10, we create a miniature-scale road by printing a real-world high-resolution BEV road texture on multiple ledger-size papers and concatenating them together to form a long straight road. In the attack evaluation time, we create the miniature-scale malicious road patch using the same method, and place it on top of the miniature-scale road following our DRP attack design. The patch is printed with a commodity printer: RICOH MP C6004ex Color Laser Printer. We mount EON, the official OpenPilot dashcam device, on a tripod and face it to the miniature-scale road. The road size, road patch size, and the EON mounting position are carefully calculated to represent OpenPilot installed on a Toyota RAV4 driving on a standard 3.6-meter wide highway road at 1:12 scale. We also create different lighting conditions with two studio lights. The patch size is set to represent a 4.8me wide and 12m long one in the real world scale. The other settings are the same as in §3.5.3.

Evaluation metric. Since the camera is mounted in a static position, we evaluate the attack effectiveness directly using the steering angle decision at the frame level instead of the lateral deviation used in previous sections. This is equivalent from the attack effectiveness point of view since the large lateral deviation is essentially created by a sequence of large steering angle decisions at the frame level. Specifically, we first find the camera frame that has the same relative position between the camera and the patch as that in the miniature-scale experimental setup. Then we compare its *designed* steering angle at the attack generation time and its *observed* steering angle that the ALC system in OpenPilot intends to apply to the vehicle in the miniature-scale experiment. Thus, the more similar these two steering

angles are, the higher realizability our attack has in the physical world.

Results. Fig. 3.11 shows a visualization of the lane detection results of the benign and attacked scenarios in the miniature-scale experiment using the OpenPilot's official visualization tool. As shown, in the benign scenario, both detected lane lines align accurately with the actual lane lines, and the desired driving path is straight as expected. However, when the malicious road patch is placed, it bends the detected lane lines significantly to the left and causes the desired driving path to be curving to the left, which is exactly the designed attack effect of our lane-bending objective function (§3.4.3). In this case, the designed steering angle is 23.4° to the left at the digital attack generation time, and the observed one in the physical miniature-scale experiment is 24.5° to the left, which only differs by 4.7%. In contrast, in the benign scenario the observed steering angle for the same frame is 0.9° to the right.

Robustness under different lighting conditions. We repeat this experiment under 12 lighting conditions ranging from 15 lux (corresponding to sunset/sunrise) to 1210 lux (corresponding to midday of overcast days). The results show that the same attack patch above is able to *maintain a desired steering angle of 20-24° to the left under all 12 lighting conditions*, which are all significantly different from the benign case (0.9° to right). Details are in our extended version [293].

Robustness to different viewing angles. We evaluate the robustness from 45 different viewing angles created by different distances to the patch and lateral offsets to the lane center. Our results show that our attack always achieves over 23.4° to the left from all viewing angles. We record videos in which we dynamically change viewing angles in a wide range while showing real-time lane detection results under attack, available at **https://sites.google.com/view/cav-sec/drp-attack/**.

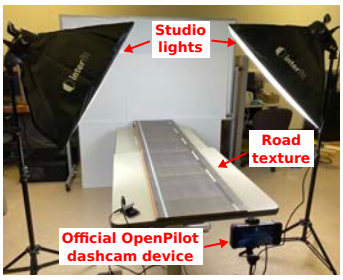


Figure 3.10: Miniature-scale experiment setup. Road texture/patch are printed on ledger-size papers.

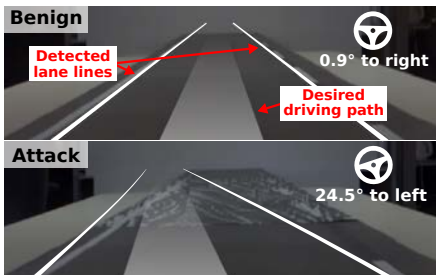


Figure 3.11: Lane detection and steering angle decisions in benign and attacked scenarios in the miniature-scale experiment.

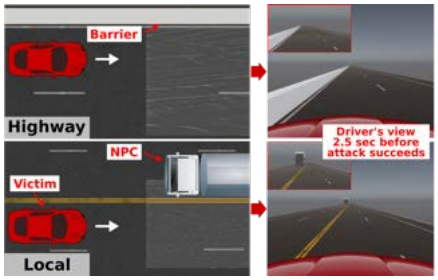


Figure 3.12: Software-in-the-loop simulation scenarios and driver's view 2.5 sec before attack succeeds. Larger images are in [293]

3.6 Software-in-the-Loop Simulation

To understand the safety impact, we perform software-in-the-loop evaluation on LGSVL, a production-grade autonomous driving simulator [19]. We overcame several engineering challenges in enabling this setup, which are detailed in our extended version [293] and open-sourced via our website [11].

Evaluation scenarios. We construct 2 attack scenarios for highway and local road settings respectively, as shown in Fig. 3.12. For the former, we place a concrete barrier on the left, and for the latter, we place a truck driving on an opposite direction lane. The attack goals are to hit the concrete barrier or the truck. Detailed setup are in Table 3.4.

Experimental setup and evaluation metrics. We perform evaluation on OpenPilot v0.6.6 with the Toyota RAV4 parameters. We follow the methodology in §3.4.3 to obtain and apply the color mapping in our simulation environment. The patch size is 5.4m wide and 70m long, and we place it in the simulation environment by importing the generated patch image into Unity. The other parameters are the same as §3.5.3. To evaluate the attack effectiveness from different victim approaching angles, for each scenario we evaluate the same patch from 18 different starting positions, created from the combinations of 2 longitudinal distances to the patch (50 and 100 m) and 9 lateral offsets (from -95% to 95%) as shown in

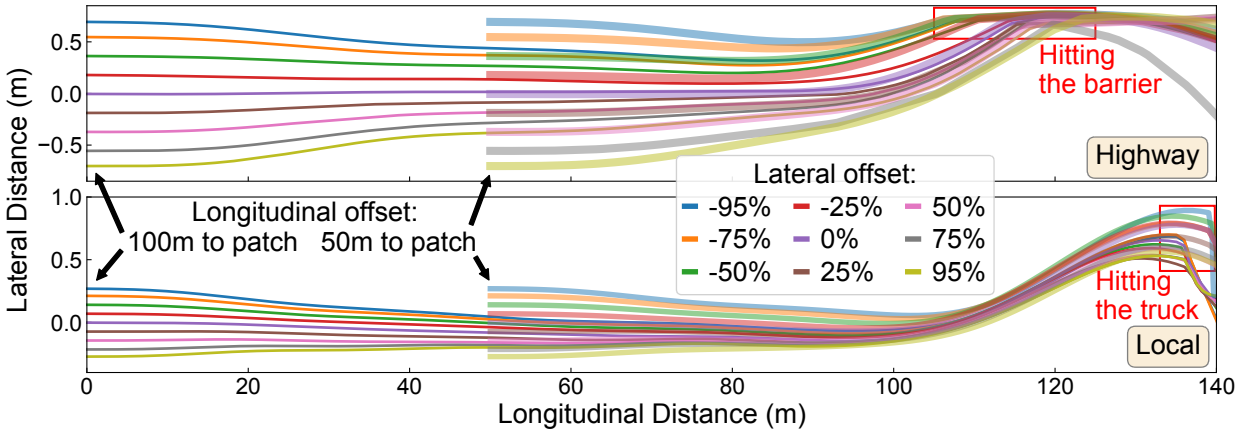


Figure 3.13: Victim driving trajectories in the software-in-the-loop evaluation from 18 different starting positions for highway and local road scenarios. Lateral offset values are percentages of the maximum in-lane lateral shifting from lane center; negative and positive signs mean left and right shifting.

Fig. 3.13. The patch is visible at all these starting positions. We repeat 10 times for each starting position in each scenario.

Results and video demos. Our attack achieves 100% success rates from all 18 starting positions in both highway and local road scenarios as shown in Table 3.4. Fig. 3.13 shows the averaged vehicle trajectories from each starting positions. As shown, the vehicle always first drives toward the lane center since the ALC system tries to correct the initial lateral deviations. After that, the patch starts to take effect, and causes the vehicle to deviate to the left significantly and hit the barrier or truck. We record demo videos at **https://sites.google.com/view/cav-sec/drp-attack/**. In the highway scenario, after the victim hits the concrete barrier, it bounces away quickly due to the abrupt collision. For local road, the victim crashes to the front of the truck, causing both the victim and truck to stop. This suggests that the safety impacts of our attack can be severe.

Table 3.4: Simulation scenario configurations and evaluation results. Lane widths and vehicle speeds are based on standard ones in the U.S. [265]. Simulation results without attack are confirmed to have 0% success rates with ≤0.018 m (std: ≤9e-4) average maximum deviations.

Sim. Scenario	Lane Width	Veh. Speed	Attack Goal	Ave. Max Dev. (std)	Succ. Rate	Succ. Time
Highway	3.6 m	65 mph (29 m/s)	Hit barrier on the left	0.76 m (5e-3)	100% (100/100)	0.97 s
Local	2.7 m	45 mph (20 m/s)	Hit truck in the opposite lane	0.55 m (7e-2)	100% (100/100)	1.36 s

3.7 Safety Impact on Real Vehicle

While the simulation-based evaluation above has shown severe safety impacts, it does not simulate other driver assistance features that are commonly used with ALC at the same time in real-world driving, for example Lane Departure Warning (LDW), Adaptive Cruise Control (ACC), Forward Collision Warning (FCW), and Automatic Emergency Braking (AEB). This makes it unclear whether the safety damages shown in §3.6 are still possible when these features are used, especially the safety-protection ones such as AEB. In this section, we thus use a real vehicle to more directly understand this.

Evaluation methodology. We install OpenPilot on a Toyota 2019 Camry, in which case OpenPilot provides ALC, LDW, and ACC, and the Camry's stock features provide AEB and FCW [29]. We then use this real-world driving setup to perform experiments on a rarely-used dead-end road, which has a double-yellow line in the middle and can only be used for U-turn. The driver's view of this road is shown on the left of Fig. 3.14. In our miniature-scale experiment in §3.5.4, the attack realizability from the physically-printed patch to the LD model output has already been validated under 12 different lighting conditions. Thus, in this experiment we evaluate the safety impact by directly injecting an attack trace at the LD model output level (detailed in Appendix 3.B.1). This can also avoid blocking the road for sticking patches to the ground and cleaning them up, which may affect other vehicles.

To create safety-critical driving scenarios, we place cardboard boxes adjacent to but outside

of the current lane as shown in Fig. 3.14, which can mimic road barriers and obstacles in opposite direction as in §3.6 while not causing damages to the vehicle and driver safety. Similar setup is also used in today's vehicle crash tests [61, 54, 55, 79]. To ensure that we do not affect other vehicles, we place the cardboard boxes only when the entry point of this dead-end road has no other driving vehicles in sight, and quickly remove them right after our vehicle passes them as required by the road code of conduct [52].

Experiment setup. We perform experiments in day time with and without attack, each 10 times. The driving speed is kept at ∼28 mph (∼45 km/h), the min speed for engaging OpenPilot on our Camry. The injected attack trace is from our simulation environment (§3.6) at the same driving speed.

Results. Our experiment results show that our attack causes the vehicle to hit the cardboard boxes in all the 10 attack trials (100% collision rate), including 5 front and 5 side collisions. The collision variations are caused by randomness in the dynamic vehicle control and the timing differences in OpenPilot engaging and attack launching. In contrast, in the trials without attack, OpenPilot can always drive correctly and does not hit or even touch the objects in any of the 10 trials.

These results thus show that driver assistance features such as LDW, ACC, FCW, and AEB are not able to effectively prevent the safety damages caused by our attack on ALC. We examine the attack process and find that LDW is not triggered since it relies on the same lane detection module as ALC and thus are affected simultaneously by our attack. ACC does not take any action since it does not detect a front vehicle to follow and adjust speed in these experiments. FCW is triggered 5 times out of the 10 collisions, but it is only a warning and thus cannot prevent the collision by itself. Moreover, in our experiments FCW is triggered only *0.46 sec* before the collision on average, which is far too short to allow human drivers to react considering the 2.5-second average driver reaction time to road hazard (§3.3).

Figure 3.14: Safety impact evaluation for our attack on a Toyota 2019 Camry with OpenPilot engaged. Even with other driver assistance features such as Automatic Emergency Braking (AEB), our attack still causes collisions in all the 10 trials.

In our Camry model, FCW and AEB are turned on together as a bundled safety feature [76]. However, while we have observed some triggering of FCW, we were not able to observe any triggering of AEB among the 10 attack trials, leading to a *100% false negative rate*. We check the vehicle manual [76] and find that this may be because the AEB feature (called *pre-collision braking* for Toyota) is used very conservatively: it is triggered only when the possibility of a collision is *extremely high*. This observation is also consistent with the previously-reported high failure rate (60%) for AEB features on popular car models today [73]. Such conservative use of AEB can reduce false alarms and thus avoid mistaken sudden emergency brakes in normal driving, but also makes it difficult to effectively preventing the safety damages caused by our attack — in our experiments, it was not able to prevent any of the 10 collisions. The video recordings for these real-vehicle experiments are available at **https://sites.google.com/view/cav-sec/drp-attack/**.

3.8 Limitations and Defense Discussion

3.8.1 Limitations of Our Study

Attack deployability. As shown in §3.5.3, our attack can achieve a high success rate (93.8%) with only 8 pieces of quickly-deployable road patches, each requiring only 5-10 sec to deploy for 2 people. To further increase stealthiness, the attacker can pretend to be road workers like in Fig. 3.2 to avoid suspicion, and pick a deployment time when the target road is the most vacant, e.g., at late night. Nevertheless, lower deployment efforts is always more preferred for attackers to reduce risks. One potential improvement is to explore other common road surface patterns besides dirty patterns, which we leave as future work.

Generality evaluation. Although we have shown high attack generality against LD models with different designs (§3.5.3), all our evaluations are performed on only one production ALC in OpenPilot. Thus, it is still unclear whether other popular ALC, e.g., Tesla Autopilot and GM Cruise, are vulnerable to our attack. Unfortunately, to the best of our knowledge, the OpenPilot ALC is the only production one that is open sourced. Due to the same reason, we are also unable to evaluate the transfer attacks from OpenPilot to these other popular ALC systems. Nevertheless, since the OpenPilot ALC is representative at both design and implementation levels (§3.5), we think our current discovery and results can still generally benefit the understanding of the security of production ALC today. Also, since DNNs are generally vulnerable to adversarial attacks [316, 178, 134, 110, 122, 168, 373, 128], if these other ALC systems also adopt the state-of-the-art DNN-based design, at least at design level they are also vulnerable to our attack.

End-to-end evaluation in real world. In this work, we evaluate our attack against various possible real-world factors such as lighting conditions, patch viewing angles, victim approaching angles/distances, printer color accuracy, and camera sensing capability (§3.5.3

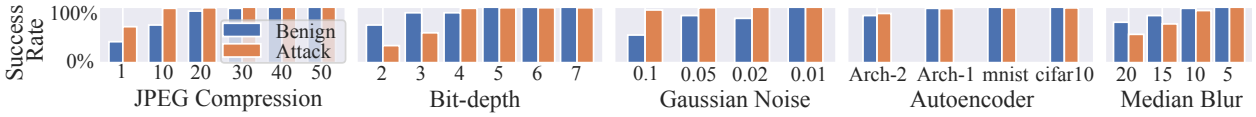


Figure 3.15: Evaluation results for 5 directly-applicable DNN model level defense methods. *Attack*: Attack success rate. *Benign*: Percentage of scenarios where the ALC can still behave correctly (i.e., not driving out of current lane) with defense applied.

and §3.5.4), and also evaluate the safety impact using software-in-the-loop simulation (§3.6) and attack trace injection in a real vehicle (§3.7). However, these setups still have a gap to real-world attacks as we did not perform direct end-to-end attack evaluation with real vehicles in the physical world. Such a limitation is caused by safety issues (vehicle-enforced minimum OpenPilot engagement speed at 28 mph, or 45 km/h) and access limits to private testing facilities (for patch placement). In the future, we hope to overcome this by finding ways to lower the minimum engagement speed and obtain access to private testing facilities.

3.8.2 Defense Discussion

Machine Learning Model Level Defenses

In the recent arms race between adversarial machine learning attacks and defenses, numerous defense/mitigation techniques have been proposed [241, 349, 352, 149]. However, so far none of them studied LD models. As a best effort to understand the effectiveness of existing defenses on our attack, we perform evaluation on 5 popular defense methods that only require model input transformation without re-training: JPEG compression [165], bit-depth reduction [352], adding Gaussian noise [371], median blurring [352], and autoencoder reformation [255], since they are directly applicable to LD models. Descriptions and our configurations of these methods are in our extended version [293]. Our experiments use the same dataset and success metrics as in §3.5. Meanwhile, we also evaluate a *benign-case success rate*, defined as the percentage of scenarios where the ALC can behave correctly (i.e.,

not driving out of lane) when the defense method is applied.

Fig. 3.15 shows the evaluation results. As shown, for each defense method we also vary the parameters to explore the trade-off between attack success rate and benign-case success rate. As shown, while all methods can effectively decrease the attack success rate with certain parameter configurations, the benign-case success rates are also decreased at the same time. In particular, when the benign-case success rates are still kept at 100%, the attack success rates are still 99 to 100% for all methods. This shows that *none of these methods can effectively defend against our attack without harming ALC performance in normal driving scenarios*. This might be because these defenses are mainly for disrupting digital-space human-imperceptible perturbations, and thus are less effective for physical-world realizable attacks with human-perceptible (but seemingly-benign) perturbations.

These results show that directly-applicable defense methods cannot easily defeat our attack. Thus, it is necessary to explore (1) novel adaptions of more advanced defenses such as adversarial training to LD, or (2) new defenses specific to LD and our problem setting, which we leave as future work.

Sensor/Data Fusion Based Defenses

Besides securing LD models, another direction is to fuse camera-based lane detection with other independent sensor/data sources such as LiDAR and High Definition (HD) map [60]. For example, LiDAR can capture the tiny laser reflection differences for lane line markings, and thus is possible to perform lane detection [111]. However, while LiDARs are commonly used in high-level (e.g., Level-4) AD systems such as Google Waymo [69] that provide self-driving taxi/truck, so far they are not generally used in production low-level (e.g., Level-2) AD such as ALC, e.g., Tesla, GM Cadillac, Toyota RAV4, etc. [41, 75, 45]. This is mainly because LiDAR is quite costly for vehicle models sold to individuals (typically $\geq$\$4,000 each

for AD [68]). For example, Elon Musk, the co-founder of Tesla, claims that LiDARs are "*expensive sensors that are unnecessary (for autonomous vehicles)*" [72].

Another possible fusion source is lane information from a pre-built HD map of the targeted road, which can be used to cross-check with the run-time detected lane lines to detect our attack. However, this requires ALC providers to collect and maintain accurate lane line information for each road, which can be time consuming, costly, and also hard to scale. To the best of our knowledge, ALC systems in production Level-2 AD systems today do not use HD maps in general. For instance, Tesla explicitly claims that it does not use HD map for Autopilot driving since it is a *"non-scalable approach"* [85].

Nevertheless, considering that Level-4 AD systems today are able to build and heavily utilize HD maps [58, 63], we think leveraging HD maps is still a more feasible solution than requiring production Level-2 vehicle models to install LiDARs. If such a map can be available, a follow-up research question is how to effectively detect our attack without raising too many false alarms, since mismatched lane information can also occur in benign cases due to (1) vehicle position and heading angle inaccuracies when localized on the HD map, e.g., due to sensor noises in GPS and IMU, and (2) normal-case LD model inaccuracies.

3.9 Related Work

Autonomous Driving (AD) system security. For AD systems, there are mainly two types of security research: *sensor security* and *autonomy software security*. For *sensor security*, prior works studied spoofing/jamming on camera [270, 353, 261], LiDAR [270, 128, 130], RADAR [353], ultrasonic [353], and IMU [333]. For *autonomy software security*, prior works have studied the security of object detection [168, 373, 130], tracking [209], localization [301], traffic light detection [323], and end-to-end AD models [328, 378]. Our

work studies autonomous software security in production ALC. The only prior effort [74] neither attacks the designed operational domain for ALC (i.e., roads with lane lines), nor generates perturbations systematically by addressing the design challenges in §3.3.3.

Physical-world adversarial attacks. Multiple prior works have explored image-space adversarial attacks in the physical world [228, 110, 122, 168, 373]. In particular, various techniques have been designed to improve the physical-world robustness, e.g., non-printability score [298, 141, 168, 377], low-saturation colors [373], and EoT [110, 122, 168, 373]. In comparison, prior efforts concentrate on image classification and object detection, while we are the first to systematically design physical-world adversarial attacks on ALC, which require to address various new and unique design challenges (§3.3.3).

3.10 Conclusion

In this work, we are the first to systematically study the security of DNN-based ALC in its designed operational domains under physical-world adversarial attacks. With a novel attack vector, dirty road patch, we perform optimization-based attack generation with novel input generation and objective function designs. Evaluation on a production ALC using real-world traces shows that our attack has over 95% success rates with success time substantially lower than average driver reaction time, and also has high robustness, generality, physical-world realizability, and stealthiness. We further conduct experiments using both simulation and a real vehicle, and find that our attack can cause a 100% collision rate in different scenarios. We also evaluate and discuss possible defenses. Considering the popularity of ALC and the safety impacts shown in this paper, we hope that our findings and insights can bring community attention and inspire follow-up research.

Appendix

3.A Required Deviations and Success Time

Required deviations. The required deviations for the highway and local roads are calculated based on Toyota RAV4 width (including mirrors) and standard lane widths in the U.S. [265] as shown in Fig. 3.A.1. We use Toyota RAV4 since it is the reference vehicle used by the OpenPilot team when collecting the comma2k19 data set [294]. For the lane widths, we refer to the design guidelines [265] published by the U.S. Department of Transportation Federal Highway Administration. The required deviations to touch the lane line are calculated using $\frac{L-C}{2} = 0.735m$ (highway) and $0.285m$ (local), where L is the lane width and C is the vehicle width.

Required success time. Since ALC systems assume a fully attentive human driver who is prepared to take over at any moment [42, 59], the required deviation above needs to be achieved fast enough so that the human driver cannot react in time to take over and steer back. Thus, when we define the attack goal, we require not only the required deviation above, but also an attack success time that is smaller than the average driver reaction time to road hazards. We select the average driver reaction time based on different government-issued transportation policy guidelines [264, 152]. In particular, in the California Department of Motor Vehicles Commercial Driver Handbook Section 2.6.1 [264], it describes (1) a 1.75 seconds *average perception time*, i.e., the time from the time the driver's eyes see a hazard until the driver's brain recognizes it, and (2) a 0.75 to 1 seconds *average reaction time*, i.e., the time from the driver's brain recognizing the hazard to physically take actions. Thus, in total it's **2.5 to 2.75 seconds** from the driver's eyes seeing a hazard to physically take actions. The UK "Highway Code Book" and "Code of Practice for Operational Use of Road Policing Enforcement Technology" use 3 seconds for driver reaction time [334, 169]. National

Safety Council also adopts a 3-second driver reaction time to calculate the minimum spacing between vehicles [152]. Among them, we select the **smallest** one, i.e., 2.5 seconds from the California Department of Motor Vehicles [264], as the required success time in this paper to avoid possible overestimation of the attack effectiveness in our evaluation.

Note that the driver reaction time above is commonly referring to the reaction time to apply the brake, instead of steering. In our paper, we use such reaction time to apply the brake as the reaction time to take over the steering wheel when the ALC systems are in control of the steering wheel. This is because in traditional driving, the driver is *actively* steering the vehicle but *passively* applying the brake. However, when the ALC system is controlling the steering, the human driver is *passively* steering the vehicle, i.e., her hands are not actively controlling the steering wheel. Thus, the reaction time to take over the steering wheel during passive steering is analogous to that to apply the brake during passive braking.

In fact, the actual average driver reaction time when the ALC system is taking control is likely to be much higher than the 2.5 seconds measured in traditional driving, due to the reliance of human drivers on such convenient driving automation technology today. A recent study performed a simulation-based user study on Tesla Autopilot, and found that 40% drivers fail to react in time to avoid a crash happening *6.2 seconds* after the Autopilot fails to operate [248]. In the real world, it is found multiple times that Tesla drivers fall asleep with Autopilot controlling the vehicle in high speed [77]. Thus, the required success time of 2.5 seconds used in this paper is a relatively conservative estimation, and thus the attack effectiveness reported in our evaluations is likely only a **lower bound** of the actual effectiveness of our attack in the real world.

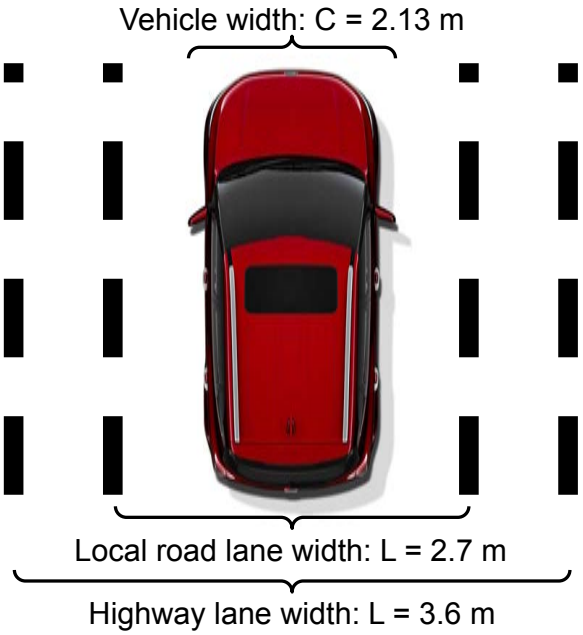


Figure 3.A.1: Vehicle and lane widths used in this paper.

3.B Attack Stealthiness User Study

In this section, we conduct a user study to more directly evaluate the stealthiness of the DRP attack. We have gone through the IRB process and our study is determined as in the IRB Exempt category since it does not involve the collection of any Personally Identifiable Information (PII) or target any sensitive population.

Evaluation methodology. We use the generated attacks on real-world driving traces in §3.5.1 to perform the user study. For an attack scenario, we ask the participants to imagine that they are driving with the ALC system taking control, and then show a sequence of image frames with the malicious road patch from the driver's view at 3, 2.5, 2, 1.5, and 1 second(s) before the attack succeeds. Here, 1 second before the attack succeeds is right before the attack starts to take effect. For each image frame, we ask whether they will decide to take over the driving to avoid danger or potential safety risks. These questions are also asked for the image frames with a benign road patch that only has the base color without the malicious dirty patterns as a control group.

Since our attack is designed for drivers who are in favor of using ALC system in normal cases, the same set of questions are asked at the beginning for the original image frames without attack, and we only accept a participant if she does not choose to take over the driving for these cases. This process also helps filter out ill-behaved participants who just provide random answers. Since DRP is a new form of attack vectors on the road, we do not tell the participants that the study is related to security attacks. Instead, we only tell them that our focus is on surveying driver's decisions under ALC systems for different road surface patterns such as road patches and scratches. At the beginning of the study, we also provide an introduction of ALC systems with demo videos to ensure that the participants fully understand what driving technology we are surveying. To understand the distribution of the participant background, we also ask demographic information and background information related to driving and ALC usage. None of the questions in our study involve PII or target any sensitive population; our study is thus determined as in the IRB Exempt category.

Evaluation setup. We use Amazon Mechanical Turk [5] to perform this study, and in total collected 100 participants. All of them have driving experience, which is confirmed by asking them the age when first licensed and the weekly driving mileage. A local-road driving trace is used in this study, and for the scenarios with attack, we evaluate 3 stealthiness levels as in §3.5.1 (i.e., $\lambda = 10^{-2}, 10^{-3}, 10^{-4}$). The survey is available at [80]. Among the 100 participants, 56% are male and 44% are female. The average age is 32.3 years old. 79% have experienced at least one ALC system, among which Tesla Autopilot has the largest share (28%). Statistics of ALC experiment and demographic information are shown in Fig. 3.B.2.

Results. Fig. 3.B.1 shows the study results. As shown, the closer it is to the attack success time, the more participants choose to take over the driving in the attacked scenarios since the dirty patterns become increasingly larger and clearer. Among the 3 stealthiness levels, the driver decisions are consistent with our design: the lowest stealthiness level ($\lambda = 10^{-4}$) has the highest take-over rate, while the highest level ($\lambda = 10^{-2}$) has the lowest. In particular,

we find that even for the lowest stealthiness level ($\lambda = 10^{-4}$), only *less than 25%* of the participants decide to take over before the attack starts to take effect. As shown in Fig. 3.7, at this stealthiness level the white dirty patterns are quite dense and prominent. Thus, these results suggest that the majority of human drivers today do not treat dirty road patches as road conditions where ALC systems cannot handle.

As introduced in §3.3.1, 2.5 seconds is commonly used as the average driver reaction time to road hazards. Thus, at 2.5 seconds or more before the attack succeeds, the human driver still has a chance to take over the driving to prevent the damage in common cases, as long as she can realize that it is a road hazard. However, our results show that only less than 20% of the participants decide to take over at 2.5 and 3 seconds before our attack succeeds even for the lowest stealthiness level. In particular, when the stealthiness levels are $\lambda = 10^{-2}$ and $\lambda = 10^{-3}$, the take-over rates at these 2 time points are similar to the rates for the benign road patch with only the base color. This suggests that at the time when there is still a chance to prevent the damage in common cases, our attack patches at $\lambda = 10^{-2}$ and 10^{-3} *appear to be as innocent as normal clean road patches to human drivers*. In these cases, the take-over rates are only *less than 15%*, which are from participants who will take over even for normal clean road patches. Note that the take-over rates in practice are likely to be lower than this since (1) this study is performed for a local road scenario, while the road patches in highway scenarios are much farther and thus much less noticeable as shown in Fig. 3.7, and (2) the road patches in this study are digitally synthesized into the image frames, which may appear less natural and thus may more easily alert the participants.

Stealthiness from pedestrian view. In local road scenarios, the stealthiness from the pedestrian's view is also an aspect worth considering, as pedestrians may report anomalies if our attack patch looks too suspicious. Our user study includes the driver's view at 1 second before the attack succeeds, which is 7 meters to the driver's eyes so similar to the distance from the pedestrian on local roads. However, only ¡25% of the participants choose

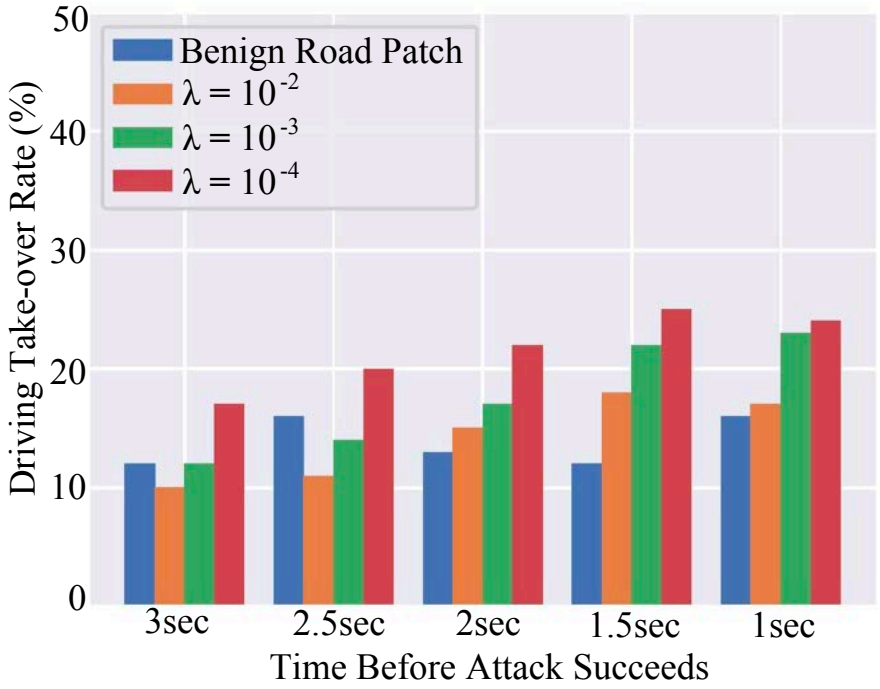


Figure 3.B.1: Results of the attack stealthiness user study. Driving take-over rate is the percentage of participants who choose to take over the driving at a particular time point before the attack succeeds.

to take over driving, meaning that ¿75% do not think our attack patch at this distance looks suspicious enough to affect driving. This may be because the general public today does not know that dirty road patches can be a road hazard. We hope that our paper can expose this and thus help raise such awareness.

3.B.1 Details of OpenPilot ALC system

In this section, we describe the implementation details of the OpenPilot ALC system, which follows the typical modular ALC system design introduced in §3.1:

Lane Detection (LD). The LD model used in OpenPilot uses recurrent DNN structures (e.g., RNN and GRU), which are more detailed in our extended version [293] for 3 specific versions of it. In each frame, the recurrent model receives a front-camera input of 512 pixels wide by 256 pixels high and 512-dimensional recurrent features from the previous frame.

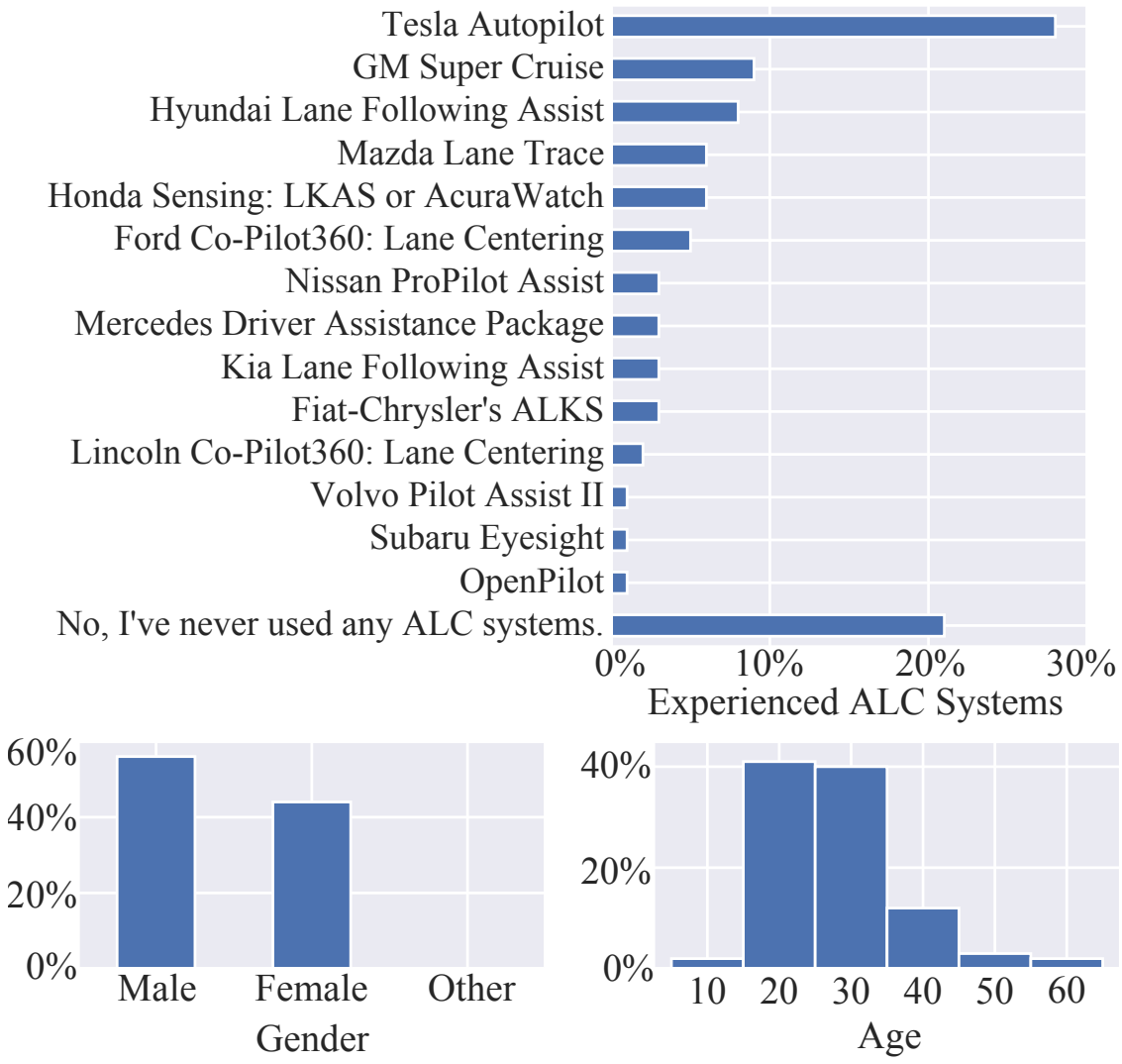


Figure 3.B.2: Statistics of the ALC system experience and demographic information in the attack stealthiness user study.

The recurrent features are the output of a middle layer. The final output for ALC consists of information of 3 lines (left and right lines and driving path). Each line has coordinates of 192 points (1 m interval from the vehicle to driving direction), uncertainty scores of each coordinate, and a confidence score of its lane. Thus, there are $(192 \times 2 + 1) \times 3 = 1,155$ output values in total. The desired driving path is calculated by the weighted average of the driving path and the center line of the left and right lines weighted by the uncertainty and confidence scores. See OpenPilot code [29] for more details.

Such recurrent structure is stateful: it allows leveraging the previous detection results to enhance the current frame detection since lane line shapes are typically not changed largely across consecutive frame. OpenPilot LD models output the detected lane line points of the left line, right line, and predicted driving path. Each line is fitted to the 3-degree polynomial, and the desired driving path is then calculated as the weighted average of the three lines with their confidence levels. OpenPilot LD operates at 20 Hz (every 50 ms). In §3.7, we inject the attack traces at the end of this step by modifying the ALC source code to replace the real-time LD model outputs with a sequence of attacked ones obtained from the software-in-the-loop simulation at the same driving speed (simulation environment described in §3.6).

Lateral control. OpenPilot adopts Model Predictive Control (MPC) [284] to decide the desired steering angle, which will then be sent to the vehicle actuation step. The input of the MPC is the desired driving path, the current speed, and the current steering angle. This step works at the same frequency as LD, i.e., the desired steering angle is decided every 50 ms. The MPC is stateful: it reuses the solution of the previous frame as the initial solution for the current frame.

Vehicle actuation. Based on the obtained desired steering angle from MPC, OpenPilot decides the steering angle *change* to actuate in the control step and sends actuation messages through CAN (Controller Area Network) bus. This thus makes the absolute value of the actuated steering angle stateful: the new actuated steering angle is built upon the previous one, by applying the angle *change* actuations. OpenPilot actuation works at 100 Hz control frequency. The actuated steering angle change is *up to 0.25° per control step* (every 10 ms). As in §3.2.1, such limit is typically imposed in production ALC systems due to the physical constraints of the mechanical control units and also for driving stability and safety [115]. OpenPilot is integrated into a vehicle by overriding the stock cruise control system. It thus is engaged to control the steering and throttle when the driver turns on the cruise control mode and can work with stock safety features such as AEB and FCW [29].

Chapter 4

A Cross-Verification Approach with Publicly Available Map for Detecting Off-Road Attacks against Lane Detection Systems

4.1 Introduction

Automated Lane Centering (ALC) [59], classified as Level-2 driving automation, is one of the most popular autonomous driving (AD) technologies widely available in many commercial vehicles such as Tesla, GM Cadillac, Honda Accord, Toyota RAV4, and Volvo XC90. ALC can automatically steer a vehicle and maintain it within its current driving lane, which is detected by windshield cameras. Despite its convenience for drivers, the ALC system carries significant security and safety implications. Despite its convenience for human drivers, ALC systems carry significant security and safety implications. DRP attack [291] demonstrated

that the commercial ALC system can be vulnerable to maliciously created dirty road patches. RAP-ALC attack [126] shows that a malicious patch on the back of a leading vehicle can mislead the victim vehicle controlled by an ALC to out of the driving lane. Due to the nature of ALC, which primarily controls steering, the attack effect on ALC is realized by lateral movement that causes the victim vehicle to deviate from its driving lane. We call this attack *off-road* attack by following the prior work [302].

To fundamentally defend against off-road attacks, two major approaches are actively researched: sensor fusion and map fusion. Sensor fusion utilizes sensing information from multiple sensors other than the windshield cameras to cross-check the detected lanes. However, sensor fusion has critical limitations to fully defend against off-road attacks: (1) The lane marking is not easy to detect other than windshield cameras. While prior research [189] demonstrates lane detection with LiDAR, it is fundamentally hard to sense the pattern on the road for other sensors than cameras; (2) The use of other sensors may open another attack vectors as there is no guarantee that the newly added sensors are more secure than the sensing with the windshield cameras.

Map fusion utilizes off-line map information to cross-check the detected lanes or to merely use the lane information in the map data. This approach is commonly used in Level-4 AD vehicles [59]. For example, Baidu Apollo [7] and Autoware [219] merely use the lane line information in map data without detecting lane lines on the fly. Since map data is typically managed by companies privately, it is not easy to compromise the data by the attacker. However, there are two major challenges in the current map fusion used in the Level-4 AD: (1) the Level-4-AD grade map, so-called High Definition (HD) map [60], needs tremendous cost to create and maintain and thus it is hard to scale to cover major roads. For example, Waymo [69] can only operate within the geo-fences such as the central area of San Francisco or Phoenix; (2) the map fusion generally requires very high-accurate vehicle localization, which is typically enabled by LiDAR localizer in the Level-4 AD. However,

AD-grade LiDARs are still too expensive to install into commodity vehicles. Due to the two challenges, map fusion is not adopted in the current popular ALC systems. The map data is only referenced for navigation as in Tesla [86] and OpenPilot [29].

However, map fusion may still have meaningful information to defend against off-road attacks even with low-quality localization (e.g., GNSS) and publicly available maps (e.g., Google Map [14] and OpenStreetMap [31]). This question motivates us to design a cross-verification approach, LaneGuard, to detect off-road attacks by cross-checking the lane line detection with the lane information in publicly available maps. In §4.3, we explain the detailed design of our cross-verification approach, LaneGuard. In §4.3.4, we evaluate the attack detection capability of LaneGuard and perform false positive analysis in benign scenarios. We also conduct a feasibility study during online real-world driving. Finally, we discuss the insights and limitations of our study in §4.4.

In summary, our study makes the following contributions:

- We are the first to design a map-fusion-based off-road attack detection method under low-quality localization and publicly available maps.
- We conduct a large-scale evaluation with multiple publicly available maps and real-world driving traces consisting of 80 attack scenarios and 11,558 benign scenarios. We find that LaneGuard can detect 89% of off-road attacks with a 12% false positive rate.
- We perform a feasibility study of LaneGuard on a real-world highway. LaneGuard shows 0% false positive rate with almost 0% false negative rate against simulated attack traces.

4.2 Background

4.2.1 Off-Road Attacks against ALC Systems

Recent studies demonstrate that ALC systems are vulnerable to off-road attacks enabled by adversarial attacks [291, 126]. These attacks compromise the DNN-based lane detection in the ALC systems to lead the victim out of their driving lane. DRP attack [291] demonstrates that it can lead an ALC-controlled vehicle out of the driving lane by fooling the DNN-based lane detection with a maliciously designed road patch pretending a benign but dirty road patch. Another study [214] shows that they can mislead Tesla Model S to the adjacent lane by putting several small stickers on the road without the original lane line. Phantom attack [261] also demonstrates that they can mislead Tesla Model S by projecting fake lane lines from a drone in the nighttime. The motivation of this study is to design an effective attack-defection methodology to defend against off-road attacks including not only these existing attacks but also any attacks compromising lane detection.

4.2.2 Prior Countermeasures against Off-Road Attacks

To defend against off-road attacks, there are 3 potential approaches: sensor-fusion, map-fusion, and software-based defenses. As discussed in §4.1, sensor-fusion-based defense requires more cost for additional sensors (e.g., LiDAR [189]) and these sensors could be new attack channels. For map-fusion-based defense, no prior work evaluates the capability in lower-level AD setups than Level-4 AD. This is one of the motivations for our research. For software-based defense, Sato et al. [291] evaluate possible defenses but none of them can be effective without harming the performance in benign cases. So far, none of the defenses against adversarial attacks is successful and the newly-proposed defenses are constantly being defeated over time [109, 331]. This also motivates us to seek a map-fusion-based defense.

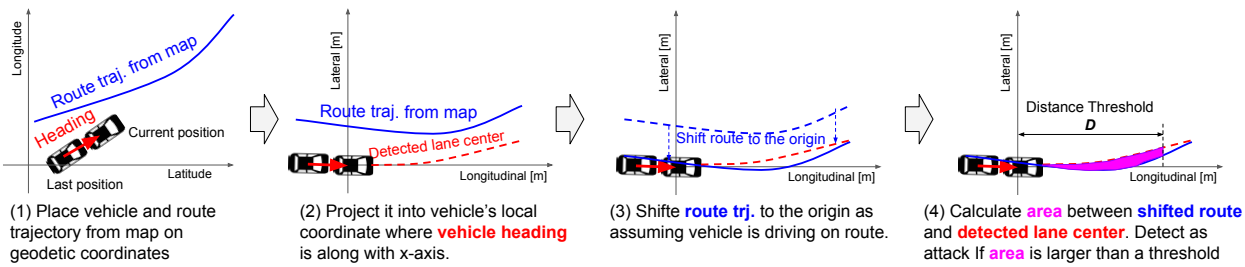


Figure 4.2.1: Overview of our LaneGuard. Its effective but simple idea is to detect off-road attacks by cross-checking the difference between the route trajectory shape and the detected road center line. To handle the low map and localization qualities, we assume that the vehicle heading matches the road direction as it is driving the road if off-road attacks have not been effective yet.

4.2.3 Publicly Available Maps

Many recent web platformers host their routing and navigation services as a part of their services, such as Google Maps [14] and Bing Maps [8]. These map services allow users to download the navigation routes upon their query via apps or APIs. OpenStreetMap [31] (OSM) is a free and open geographic database, which is maintained by volunteers around the world. While OSM itself is just indexed map data, many routing and navigation services are developed on the OSM such as OSRM [151] and OpenRouteService [30]. We can also import the OSM data into PostGIS [66] and query a route with pgRouting [34].

4.3 Methodology

In this section, We illustrate the design details of LaneGuard, which is a cross-verification approach to detect off-road attacks by comparing the lane line detection with the lane information in publicly available maps.

4.3.1 Threat Model

We assume that the attacker can launch off-road attacks by compromising the detected lane line information in an ALC system installed in the victim vehicle by some attack vectors, e.g., physical-world adversarial attacks discussed in §4.2.1 or malware. The victim's ALC can access publicly available map data and its routing services like those discussed in §4.2.3. With GPS and the IMU data, we assume that the victim's ALC has meter-level vehicle localization information that can know which road the victim is driving and its rough location, but the victim vehicle's accurate postural is not available.

4.3.2 Design Overview of LaneGuard

Fig. 4.2.1 illustrates the detection procedure of our LaneGuard. As described, (1) we first map the current driving route and the vehicle positions at the current and previous frames. The route is represented by a parameterized curve, e.g., spline and polynomial curves. We calculate the heading angle based on the position difference between the current and previous frames since we cannot directly obtain the vehicle heading due to its ill-quality localizations discussed in §4.3.1. (2) We project the route trajectory and the vehicle positions into the vehicle's local coordinate system where the vehicle heading is along with the x-axis. In this coordinate system, we can also plot the detected lane center obtained from the ALC system. The lane center is also represented by a parameterized curve. (3) We translate the route trajectory along with the y-axis to go through the origin where is the current vehicle position. Major routing services generally do not provide lane-level information, but road-level information, i.e., the route trajectory is drawn to pass through the center of the road. This is our approach to handling the coarse-grained route information from the map. We assume that the victim is not currently under attack effects and successfully driving the road. (4) Finally, we calculate the area between the shifted route trajectory and the

detected lane center and use this area as a detection metric δ to judge if the vehicle is under off-road attack, i.e., LaneGuard considers that an off-road attack is ongoing if δ is larger than a predefined threshold. For the calculation of δ, we set a distance threshold D and only consider the area until the distance because the lane line detection at far points is generally inaccurate. In summary, the attack detection metric δ can be calculated as follows:

$$\delta := \int_0^D |\text{ShiftedRoute}(x) - \text{LaneCenter(x)}|dx. \tag{4.1}$$

4.3.3 Route Path Smoothing

As described in §4.3.2, LaneGuard requires the driving route trajectory represented by a parameterized curve. However, the major routing services discussed in §4.2.3 return the route information represented by a sequence of points in a geodetic coordinate system. The simplest method is to connect them with a wire, but we find that this significantly degrades the detection accuracy of LaneGuard. The frequency of points in a route varies widely between routing services. Particularly, Google Maps [14] generates a route with very high frequency but perturbed points, which should be automatically generated by Google users' prob data. If we directly connect them with a line or interpolate them with a curve (e.g., spline), the calculated route trajectory will be highly zig-zagged. Fig. 4.3.1 shows To handle this, we first apply a Gaussian filtering-based path smoothing for the route points, and then apply the cubic spline interpolation for the smoothed points to get the parameterized curve.

4.3.4 Evaluation

We evaluate the performance of the LaneGuard with respect to its detection capability against off-road attacks and its sensitivity against benign scenarios.

(a) OpenStreetMap (b) Google Maps

Figure 4.3.1: Comparison of OpenStreetMap and Google Maps on a highway. The markers represent waypoints of the driver-defined route. The blue and red lines are the driver-defined route and the GNSS trajectory of our driving, respectively.

Evaluation on Attack Detection Capability

Experimental Setup. We simulate the attack effect of the off-road attacks by using the attack traces used in the DRP attack [291], which is one of the most effective off-road attacks so far and the only prior work that shared their evaluation code and scenarios with real-world driving traces. The scenarios are extracted from the comma2k19 dataset [294] and cover 40 eligible 10-second driving clips. Half of them are on highways, the other half are on local roads. For the vehicle location, we use the GNSS positions in the comma2k19 dataset [294] corrected with a smartphone-grade UBlox's GNSS receiver. For each clip, we generate attacks to lead the victim to the left and right, respectively. In total, there are 80 different attack

scenarios. To simulate the attack-influenced driving, we use the input transformation with the perspective transformation as used in [291]. We use the lane detection model used in the OpenPilot v0.7.0 [29]. With these scenarios, we generate the DRP attacks and evaluate the attack detection rates of the LaneGuard. We place the DRP attack patch 7 m away from an attack generation point.

We start the driving simulation 1 second before the attack generation point, i.e., the distance from the simulation start point to the patch varies in each scenario based on the vehicle speed. From the simulation start point, we simulate the driving for 2 seconds considering the average attack success time of the DRP attack is around 1 second. For publicly available maps, we evaluate 2 different maps: OpenStreetMap [31] and Google Maps [14]. To evaluate the quality of the publicly available maps, we also evaluate 2 other route trajectories that can be seen as similar to ground truth. One is the human driving trajectory of the comma2k19 dataset. Another one is a simulated driving trajectory with benign scenarios. The simulated driving trajectory is generated by calculating the vehicle position by using the bicycle vehicle motion model [277] based on the lane detection results.

Results. Table 4.3.1 lists the detection rates, the best thresholds δ to achieve them, and the average δ in attacked and benign scenarios. As shown, the publicly available maps, OpenStreetMap and Google Maps, have the highest detection rate as 89% of the off-road attacks are correctly detected. Meanwhile, the detection rates with the human driving and simulated route trajectory are lower than these publicly available maps. We consider that these approaches suffer from inaccuracies in their GNSS and/or bicycle models, and this observation highlights the necessity of utilizing offline map information for defense since online sensing always contains a certain level of error.

For the results with OpenStreetMap and Google Maps, the average benign δ is around 30 m^2. This means that the route trajectory and the detected lane center have around 1.5 m deviation average since we use $D = 20$ m (§4.3.2). As typical smartphone-grade GNSS

Table 4.3.1: Attack detection rates of LaneGuard on different publicly available maps and baselines.

Map/Baseline	Detection Rate	Threshold θ [m^2]	Avg. Attacked δ [m^2]	Avg. Benign δ [m^2]
OpenStreetMap	89%	89.8	141.1	28.6
Google Maps	89%	92.3	143.1	33.1
Human Driving	85%	74.2	141.0	28.8
Simulated Route	76%	125.7	142.9	43.3

is accurate to within a 4.9 m [336], the 1.5 m deviation in benign scenarios is reasonable. Considering the typical lane width is 2-3 m, this deviation is not so accurate, but it still has the potential to detect off-road attacks that try to largely deviate the victim vehicles out of the lane. On the other hand, The average attacked δ is around 140 m^2, i.e., 7 m average deviation between the detected lane center and the route trajectory. The best threshold θ for the attack detection thus can be around 90 m^2 (4.5 m average deviation). Since these deviations are close to the accuracy of the smartphone-grade GNSS, it may be challenging for this naive single-frame δ-based approach to detect all attacks as 11% of attacks are not correctly detected even with OpenStreetMap and Google Maps. To further explore this, we will evaluate more advanced detection designs leveraging the knowledge in multi frames.

False Positive Analysis

For a defense mechanism to be effective, it must not do much harm to the usability of the application it defends. As the LaneGuard is designed to detect off-road attacks, we evaluate the sensitivity of LaneGuard detection in large-scale benign driving traces.

Experimental Setup. To evaluate the false positive rate on benign driving, we extracted further scenarios from the comma2k19 dataset [294]. We split all driving traces in the comma2k19 dataset into every 5 seconds and obtain 30,565 5-second driving traces. Among the driving traces, we select the traces for the evaluation with the following criteria: (1) its

minimum speed is larger than 45 km/h, which is a typical operational speed of OpenPilot [29]; (2) the lateral deviation between the human driving in the trace and simulated trajectory based on the lane detection result = is less than 1 m because such scenarios should not be in the operational domain of ALC (e.g., lane changing and turning at an intersection). After the filtering, we eventually find 11,558 valid driving traces and calculate the detection metric δ for the first 20 frames (1 second) of the traces, i.e., we evaluate the false positive with 231,160 frames. Note that each frame has a 0.05-second duration. For the map, we use OpenStreetMap and the route trajectory is calculated by pgRouting [34].

Results. Fig. 4.3.2 shows the histogram of the detection metric δ in the 57,790 frames. The red vertical line represents the detection threshold $\theta = 89.8$ m^2 which is the best threshold for the attack detection as discussed in §4.3.4. With this threshold θ, the false positive rate is 12%. As shown, the LaneGuard does not cause false positives for the majority of the benign frames. There could be multiple potential causes such as inaccurate matching with map and location, inaccurate vehicle heading calculation, and inaccuracies in the GNSS localization. To diagnose 12% of long-tail false positive cases, we will perform further real-world online detection evaluation on a highway, which is the main operational domain of ALC.

Failure Case Analysis

To diagnose the root causes of the false positives, we diagnose them and find 2 common causes for false positives. Fig. 4.3.3 shows the 2 representative scenarios. In (a), the driver-defined route (dotted yellow line) starts curving to the left even though the road is still straight and the detected lane center is also detected like so. The road soon starts curving after around 100 m away. This false positive is caused by associating the GNSS position with the wrong point of the road. In (b), the false positive is caused by breaking the assumption that the vehicle heading and road curvature are the same. As shown, the vehicle is slightly more heading to the left. Furthermore, we think that the detected lane center represents

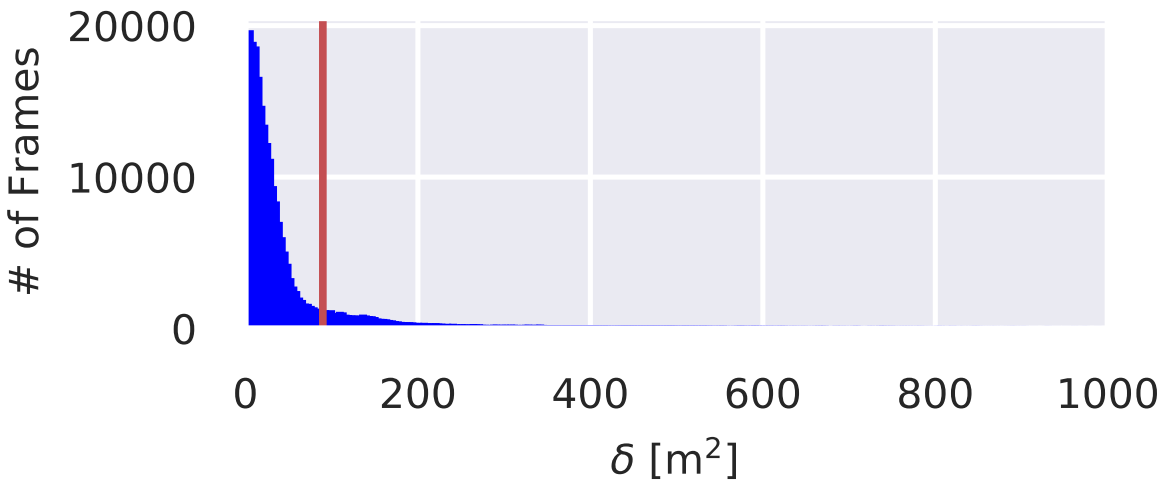
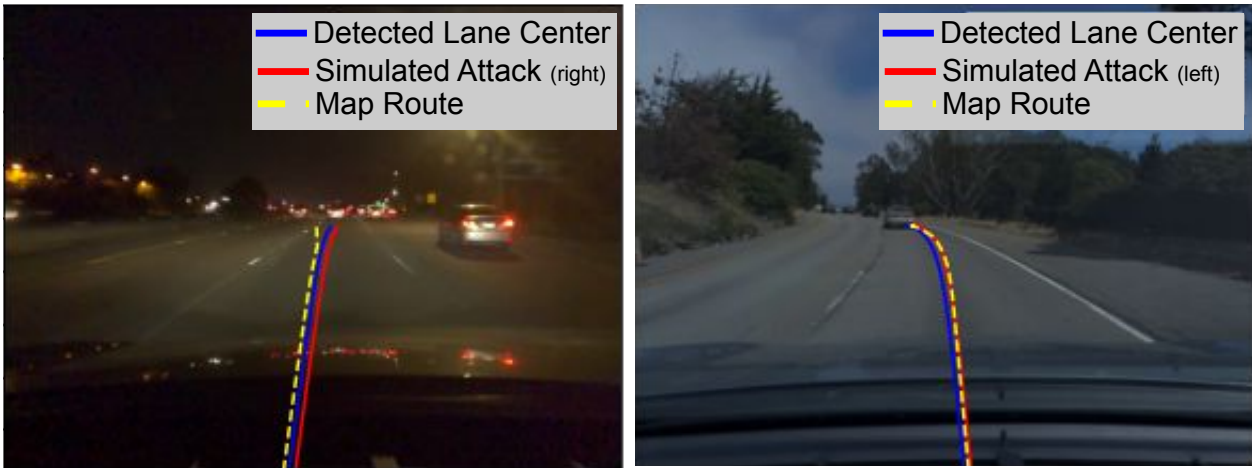


Figure 4.3.2: Histogram of the detection metric δ in the 57,790 frames. The red vertical line is the detection threshold $\theta = 89.8$ m^2 with a 12% false positive rate.



(a) Failure on map matching (b) Mismatch on heading and road curvature

Figure 4.3.3: Case studies on false positives: (a) The map is associated with the wrong point as the road curve should start at a further point; (b) the vehicle heading does not match with the road curvature while the detected lane center looks accurate.

the actual road more accurately than the map route. While the LaneGuard can handle the majority of benign scenarios, the false positives derived from the ill-quality of localization and maps are inevitable.

Attack Detection with Multi-Frame Metrics

As a simple extension, LaneGuard can leverage the information in the past multiple frames instead of the single-frame δ. LaneGuard depends on low-quality GNSS localization and publicly available maps, multi-frame aggregation of the detection metric δ (e.g., averaging) may improve the detection accuracy and reduce false positive rates.

Experimental Setup. We use the same setups used in §4.3.4 (for attack detection rate analysis) and §4.3.4 (for false positive analysis). We denote the multi-frame attack detection metrics as δ^{f}_{w}, where f means the aggregation method and w means the number of frames to aggregate δ from the current frame to the past with f. In this work, we explore four aggregation functions: mean, median, min, and max. The attack detection threshold θ is decided to maximize the detection rate and used for the false positive analysis for each aggregation method and window size.

Results. Fig. 4.3.4 and 4.3.5 show the attack detection rates and false positive rates on different multi-frame windows and their aggregation methods. As shown, the multi-frame detection metrics do not show any improvements in both detection and false positive rates. While longer aggregation frames with minimum aggregation slightly reduce the false positive rates, minimum aggregation largely harms the attack detection rates. We consider that these counterintuitive results should be due to the side-effect of the LaneGuard design, which is designed to be robust against low-quality localization and publicly available maps. As in §4.3.2, LaneGuard assumes that the current vehicle heading is the same as the curvature of the current driving road; path smoothing is also applied for the road shape. These operations may dismiss the difference between δ of the close frames, and produce similar δ values. The aggregations thus do not make meaningful differences from the single-frame δ.

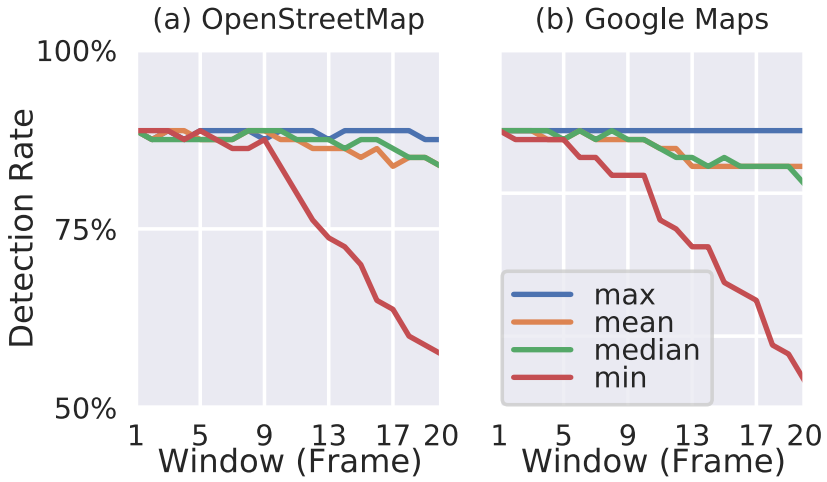


Figure 4.3.4: Attack detection rates on different frame windows and aggregation methods.

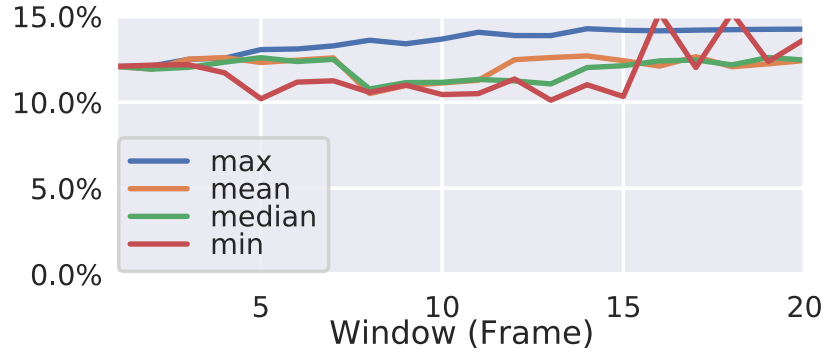


Figure 4.3.5: False positive rates on different frame windows and aggregation methods.

Ablation Study on Path Smoothing

As discussed in §4.3.3, several map services such as Google Maps generate routes with highly frequent points that should be automatically generated by their users' prob data. Table 4.3.2 lists the detection rates, the best thresholds δ to achieve them, and the average δ in attacked and benign scenarios, when we disable the path smoothing in the driver-defied route trajectory. As expected, the detection rates are significantly dropped from 89% to 65% on Google Maps compared to the case when the path smoothing is enabled as in Table 4.3.1. Meanwhile, the detection rate of Open Steet Map is slightly improved even without the smoothing. These results indicate that the benefit of path smoothing highly depends on the map source. For Google Maps, path smoothing is quite effective since the path waypoints

Table 4.3.2: Attack detection rates of LaneGuard *without the path smoothing* on different publicly available maps.

Map	Detection Rate	Threshold θ [m^2]	Avg. Attacked δ [m^2]	Avg. Benign δ [m^2]
OpenStreetMap	91%	106.8	161.6	30.6
Google Maps	65%	109.4	198.0	66.2

are frequent but noisy. For OpenStreetMap, the waypoints are sparse and could be already smoothed. Smoothing can reduce the noise effects, but it also causes information losses. For LaneGuard implementation, we need a pre-assessment of the publicly available maps we plan to use, particularly about the characteristics of the route points.

Feasibility Study on Real-World Highway

We evaluated the detection capability of the LaneGurad in §4.3.4 and its false positive analysis in §4.3.4 with the driving traces in the comma2k19 dataset [294].

Experimental Setup. As shown in Fig. 4.3.6, we installed an EON Devkit, the official dashcam device of OpenPilot [29], onto the windshield of a sedan vehicle. We drove the 2 km highway route without changing lanes to be consistent with the driving controlled by ALC. The vehicle speed was maintained around 120 km/h. We used a laptop to connect the dashcam and obtained real-time driving logs via SSH. LaneGuard operated with the logs and showed online detection results on the laptop. To simulate the attack scenario, we added the offset of the average DRP attack trace generated in §4.3.4 into the online detected lane center. For the map data, we use OpenStreetMap [31] and pgRouting [34].

Results. For the attack detection accuracy, we did not observe any false positives and only observed four times false negatives for the simulated attack. All false negatives only lasted 1 frame (0.05 sec) and thus the driver can get an attack detection alert at least within 0.1 sec, which is an ignorable delay considering human reaction time. For the latency of LaneGuard,

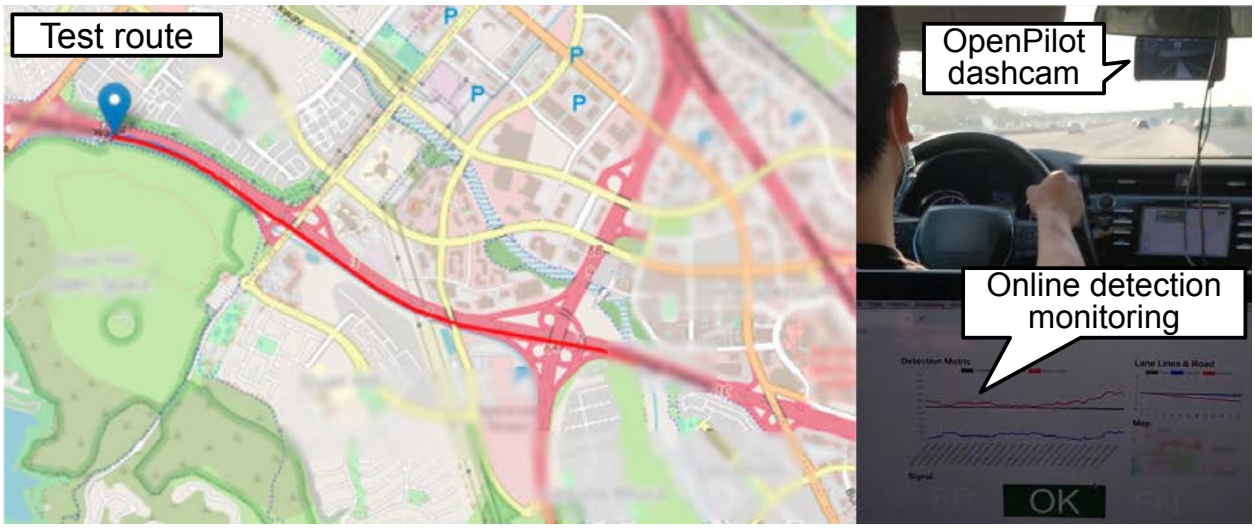


Figure 4.3.6: Overview of our feasibility study on a real-world highway: The highway route for the real-world evaluation (left) and the setup for the online LaneGuard detection on a real vehicle (right). The LaneGuard is operated on the laptop connecting to the OpenPilot dashcam via SSH.

we did not see particular delays. We find that LaneGuard is efficient enough to handle real-time driving logs sent from OpenPilot at every 0.05 sec (20 Hz).

4.4 Discussions and Limitations

Localization Quality. In this work, we only use GNSS localization with a smartphone-grade receiver. While other expensive localizers such as LiDAR localizers are not available for typical ALC systems, we may still use other information sources to improve the localization quality. For example, IMU data is widely used for improving localization quality [7, 219]. In our setup, we may also use the driver-define route under some assumptions, e.g., the vehicle should be on the route.

Further Improvement on Detection Algorithm. This study is motivated by our question about the potential of low-quality localization and publicly available maps to defend against off-road attacks. To simply answer this question, we focus more on exploring the usefulness

of information sources (low-quality localization and publicly available maps) rather than designing complex but more effective detection algorithms as we are the first to design map-fusion-based defense against off-road attacks. As a straightforward way to improve detection accuracy, we can adopt a machine learning (ML)-based detection approach. In addition to our detection metric δ, we can integrate any information as a feature such as vehicle speed, steering angle, previous locations, and route information (e.g., highway and local road). However, for safety-critical applications, accountability is as important as detection accuracy. ML-based approaches typically need large theoretical or empirical efforts to ensure their accountability. We thus leave the ML-based detection for future work.

Latency and Energy Consumption of LaneGuard. In §4.3.4, we find that LaneGuard can handle online driving logs sent from OpenPilot at every 0.05 sec. However, we run the LaneGuard on our laptop, which has much higher computational power than the smartphone-like OpenPilot's EON DevKit. To install the LaneGuard along with ALC systems, more detailed and quantitative evaluation of its latency and energy consumption analysis may help ALC developers estimate the cost of integrating the LaneGuard.

Detection against Adaptive Attacks. As LaneGuard detects off-road attacks based on the difference between road shape and the detected lane center, the attacker may design more stealthy attacks that gradually lead victims out of the lane while always keeping the difference below the threshold. However, this type of attack should require more time since the attacker cannot largely compromise the detected lane center. For the attack against ALC, the attack duration needs to be below the driver's reaction time otherwise the driver can take over the driving and apply countersteering. We thus leave this for future study as this needs more advanced attack design and careful evaluation design for user study.

4.5 Conclusion

In this work, we explore the potential defense capability of low-quality localization and publicly available maps against off-road attacks. We design the first map-fusion-based off-road attack detection, named LaneGuard. To handle the ill-quality localization and map data, we introduce a key assumption that vehicle heading and the curvature of driver-defined route trajectory should match if off-road attacks have not been effective yet. To evaluate the performance of the LaneGuard, we perform the attack detection capability analysis on 80 attack scenarios and evaluate the false positive analysis on 11,558 benign scenarios. Through the evaluation, we find that LaneGuard can detect 89% of off-road attacks with 12% false positive rates. To further evaluate the usability of LaneGuard, we conduct real-world driving experiments on the highway, which is the main operational domain of ALC. In the experiments, we do not observe any false positives, i.e., 0% false positive rate while keeping almost 0% false negative rate against simulated attacks. Recently, map information has become publicly available on the internet, but its majority application is navigation. While publicly available maps are not as high quality as merely supporting AD, we hope that our study facilitates further utilization of map information to secure AD vehicles.

Chapter 5

Towards Driving-Oriented Metric for Lane Detection Models

5.1 Introduction

Lane detection is one of the key technologies for realizing autonomous driving today. For lane detection, camera is the most commonly used sensor because it is a natural choice since lane lines are visual patterns [196]. Like most other areas of computer vision areas, lane detection has been significantly benefited from the recent advances of deep neural networks (DNNs). In the 2017 TuSimple Lane Detection Challenge [62], DNN-based lane detection shows substantial performance as all top three teams chose DNN-based lane detection. After this competition, its dataset and evaluation method based on accuracy and F1 score became the *de facto* standard in lane detection evaluation. These metrics are inherited by the subsequent datasets [267, 116].

However, the validity of this evaluation method in practical contexts, i.e., whether this is representative of practicality in real-world downstream applications, has not been ade-

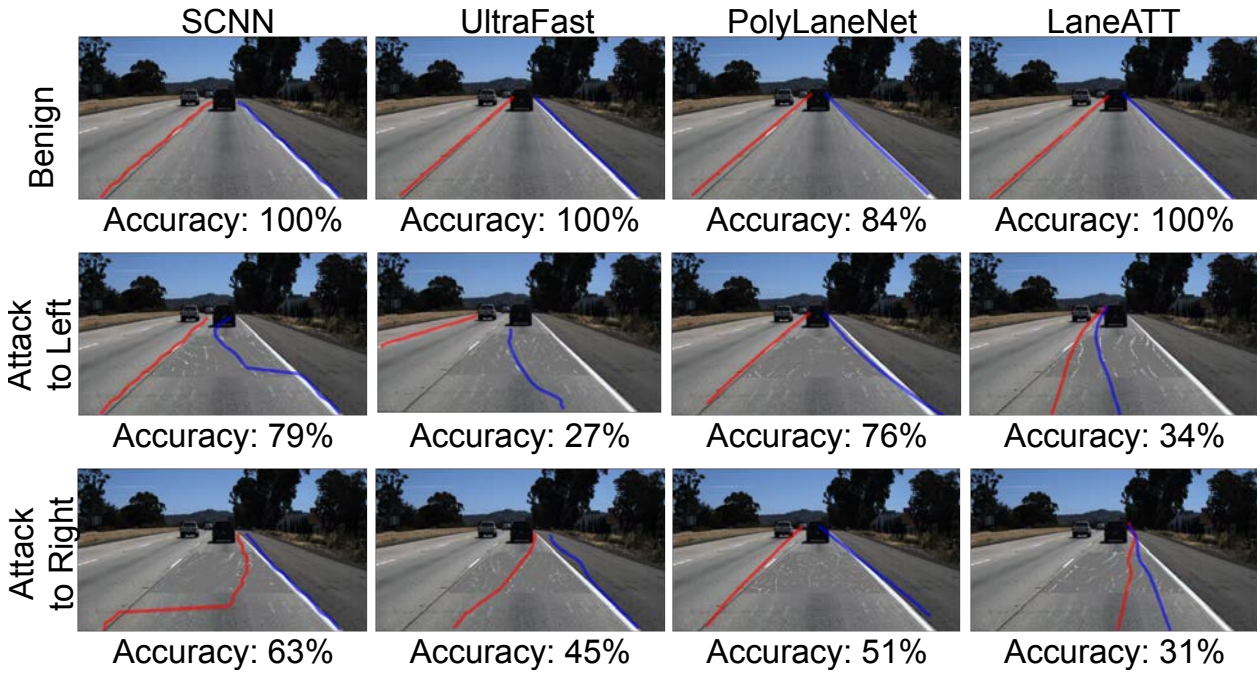


Figure 5.1.1: Examples of lane detection results and the accuracy metric in benign and adversarial attack scenarios on TuSimple Challenge dataset [62]. As shown, the conventional accuracy metric does not necessarily indicate drivability if used in autonomous driving, the core downsteam task. For example, SCNN always has higher accuracy than PolyLaneNet, but its detection results are making it much harder to achieve lane centering (detailed in §5.4.2).

quately researched. Specifically, the main real-world applications of lane detection are for autonomous driving (AD), e.g., online detection for automated lane centering (for lower-level AD such as in Telsa AutoPilot [41]), and offline detection for high-definition map creation (for both low-level [40] and high-level AD [322]). With such an application domain as its main target, the robustness of lane detection is highly critical as errors from it could be fatal. Unfortunately, we find that the conventional evaluation metrics (i.e., accuracy and F1 score) have limitations in correctly reflecting the performance of lane detection models in such main downstream application domain, especially in more challenging scenarios (e.g., when under adversarial attacks). Fig. 5.1.1 shows a few such examples that motivate this study. In the adversarial attack settings, the lane lines detected by SCNN [267] are largely disrupted, but their performance measured by the conventional accuracy metric is always higher than the one of PolyLaneNet [318], which are generally aligned with actual lane lines (and indeed

lead to less lane center deviation than SCNN when used with driving models as quantified later in §5.4.2). In the benign settings, PolyLaneNet has the lowest accuracy and is underestimated, despite its seemingly perfect detection for humans. As lane detection has been evaluated using mainly relatively clean and homogeneous driver's view images, it is not easy to identify such a great discrepancy at the metric level. Considering the criticality of robust lane detection to correct and safe AD, it is important to address such a metric-level limitation since (1) the cornerstone of real-world deployment and commercialization of AD today is exactly on the handling of those more challenging driving scenarios [365, 162, 207]; and (2) with increasingly more discoveries of physical-world adversarial attack on lane detection in AD context [291, 214], it is desired to have a more downstream task-aware performance metric when judging the model robustness (and its enhancement).

Motivated by such critical needs, we design 2 new driving-oriented metrics, *End-to-End Lateral Deviation metric* (E2E-LD) and *Per-frame Simulated Lateral Deviation metric* (PSLD), to measure the performance of lane detection models in AD, especially in Automated Lane Centring (ALC), a Level-2 driving automation that automatically steers a vehicle to keep it centered in the traffic lane [59]. E2E-LD is designed directly based on the requirements of driving automation by ALC. PSLD is a lightweight surrogate metric of E2E-LD that estimates the impact of lane detection results on driving from a single frame. This per-frame lightweight design allows the metric to be usable during upstream lane detection model training. To evaluate the validity of the metrics, we conduct a large-scale empirical study of the 4 major types of lane detection approaches on the TuSimple dataset and our newly constructed dataset, Comma2k19-LD, which contains both lane line annotation and driving information. To simulate corner-case but physically-realizable scenarios as in Fig. 5.1.1 for lane detection, we utilize and extend physical-world adversarial attacks on ALC [291]. We formulate attack objective functions to fairly generate adversarial attacks against the 4 major types of lane detection approaches. Throughout this study, we find that the conventional metrics have strongly negative correlations ($r \leq$-0.55) with E2E-LD in the benign scenarios,

meaning that some recent improvements purely targeting the conventional metrics may not have led to meaningful improvements in AD, but rather may actually have made it worse by overfitting to the conventional metrics. In the attack scenarios, while we observe a slight positive correlation ($r \leq 0.08$), it is not statistically significant. Consequently, we find that the conventional metrics tend to overestimate less robust models. On the contrary, our newly-designed PSLD metric is always strongly positively correlated with E2E-LD ($r \geq 0.38$), and all correlations are statistically significant ($p \leq 0.001$).

While the TuSimple Challenge dataset and its evaluation metrics have played a substantial role in developing performant lane detection methods, the recent improvement on the conventional metrics does not lead to the improvement on the core downstream task AD. We thus want to inform the community of such limitations of the conventional evaluation and facilitate research to conduct more downstream task-aware evaluation for lane detection, as the gap between upstream evaluation metrics and downstream application performance may hinder the sound development of lane detection methods for real-world application scenarios.

In summary, our contributions are as follows:

- We design 2 new driving-oriented metrics, E2E-LD and PSLD, that can more effectively measure the performance of lane detection models when used for AD, their core downstream task.
- We design a methodology to fairly generate physical-world adversarial attacks against the 4 major types of lane detection models.
- We build a new dataset Comma2k19-LD that contains both lane annotations and driving information which are necessary for the E2E-LD and PSLD.
- We are the first to conduct a large-scale empirical study to measure the capability of 4 major types of lane detection models in supporting AD.
- We highlight and discuss the critical limitations of the conventional evaluation and demon-

strate the validity of our new downstream task-aware metrics.

Code and data release. All our codes and datasets are available on our project websites [1].

5.2 Related Work

5.2.1 DNN-based Lane Detection

We taxonomize state-of-the-art DNN-based lane detection methods into 4 approaches. Similar taxonomy is also adopted in prior works [317, 246].

Segmentation approach. Segmentation approach handles lane detection as a segmentation task, which classifies whether each pixel is on a lane line or not. Since this approach achieved the state-of-the-art performance in the 2017 TuSimple Lane Detection Challenge [62] (all top-3 winners adopt the segmentation approach [267, 202, 263]), it has been applied in many recent lane detection methods [374, 200, 375]. This segmentation approach is also used in the industry. A reverse-engineering study reveals that Tesla Model S adopts this segmentation-based approach [214]. The major drawback of this approach is its higher computational and memory cost than the other approaches. Due to the nature of the segmentation approach, it needs to predict the classification results for every pixel, the majority of which is just background. Additionally, this approach requires a postprocessing step to extract the lane line curves from the pixel-wise classification result.

Row-wise classification approach. This approach [274, 361, 199, 246] leverages the domain-specific knowledge that the lane lines should locate the longitudinal direction of driving vehicles and should not be so curved to have more than 2 intersections in each row

[1] https://github.com/ASGuard-UCI/ld-metric
https://sites.google.com/view/cav-sec/ld-metric

of the input image. Based on the assumption, this approach formulates the lane detection task as multiple row-wise classification tasks, i.e., only one pixel per row should have a lane line. Although it still needs to output classification results for every pixel similar to the segmentation approach, this divide-and-conquer strategy enables to reduce the model size and computation while keeping high accuracy. For example, UltraFast[274] reports that their method can work at more than 300 FPS with a comparable accuracy 95.87% on the TuSimple Challenge dataset [62]. On the other hand, SAD [200], a segmentation approach, works at 75 frames per second with 96.64% accuracy. This approach also requires a postprocessing step to extract the lane lines similar to the segmentation approach.

Curve-fitting approach. The curve-fitting approach [318, 271] fits the lane lines into parametric curves (e.g., polynomials and splines). This approach is applied in an open-source production driver assistance system, OpenPilot [29]. The main advantage of this approach is lightweight computation, allowing OpenPilot to run on a smartphone-like device without GPU. To achieve high efficiency, the accuracy is generally not high as other approaches. Additionally, prior work mentions that this approach is biased toward straight lines because the majority of lane lines in the training data are straight [318].

Anchor-based approach. Anchor-based approach [317, 238, 275] is inspired by region-based object detectors such as Faster R-CNN [283]. Each lane line is represented as a straight proposal line (anchor) and lateral offsets from the proposal line. Similar to the row-wise classification approach, this approach takes advantage of the domain-specific knowledge that the lane lines are generally straight. This design enables to achieve state-of-the-art latency and performance. LaneATT [317] reports that it achieves a higher F1 score (96.77%) than the segmentation approaches (95.97%) [200, 267] on the TuSimple dataset.

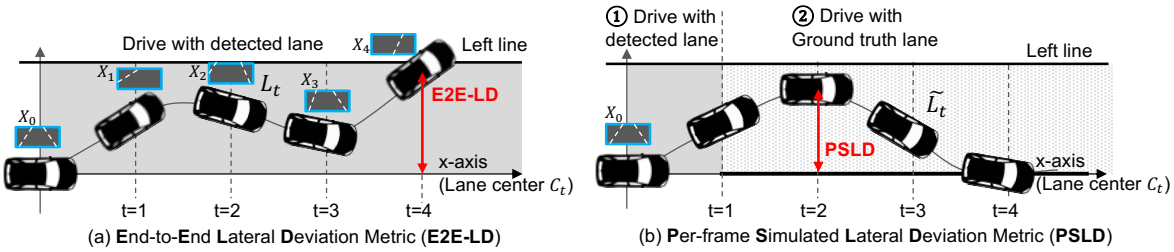


Figure 5.2.1: Overview of our driving-oriented metrics for lane detection models: E2E-LD and PSLD. X_t are camera frames from driver's view (lane detection model inputs). E2E-LD requires multiple (consecutive) camera frames, while PSLD only uses the current frame X_0.

5.2.2 Evaluation Metrics for Lane Detection

All lane detection methods we discuss in §5.2.1 evaluate their performance on the *accuracy* and *F1 score* metrics used in the 2017 TuSimple Challenge [62]. The *accuracy* is calculated by $\sum_{i \in H} \frac{\text{tp}_i}{|H|}$, where H is a set of sampled y-axis points in the driver's view image and tp_i is 1 if the difference of a predicted lane line point and the ground truth point at $y = i$ is within α pixels; otherwise is 0. α is set to 20 pixels in the TuSimple Challenge. The detected lane line is associated with a ground truth line with the highest accuracy. In other datasets [267, 116], IoU (Intersection over Union) is also used instead of accuracy. However, the ground-truth area is only defined as a 30-pixel wide line based on lane points, and this metric is almost equivalent to accuracy. The *F1 score* is a common metric to measure the performance of binary classification tasks. This is the harmonic mean of precision and recall: $\frac{2}{\text{recall}^{-1}+\text{precision}^{-1}}$. In the TuSimple Challenge, the precision and recall are calculated at the lane line level: The precision is the true positive ratio of detected lane lines and the recall is the true positive ratio of ground truth lines. The true positive is defined if the accuracy of a pair of the ground truth line and detected line is $\geq \beta$. β is set to 0.85 in the TuSimple Challenge. Although the *accuracy* and *F1 score* can measure the capability of lane detection at a certain level, these metrics do not fully represent the performance in the main real-world downstream application, AD [41, 40, 322], as concretely shown later in §5.4.2.

Specifically, to reflect its performance if used in AD, or *drivability*, accuracy and F1 score metrics have 2 major limitations: **(1)** There is no justification of $\alpha = 20$ pixels and $\beta = 0.85$ accuracy thresholds. For example, the ALC system can keep at the lane center even if the detection error is more than 20 pixels, as long as the detected lane lines are *parallel* with actual lane lines. Furthermore, the importance of detected lane line points should not be equal, i.e., closer points to the vehicle should be more important than the distanced points to control a vehicle. **(2)** The current metrics treat all lane lines in the driver's view equally, e.g., detection errors for the ego lane's left line are treated the same as the detection errors for the left lane's left line. However, the former is much more important to ALC systems than the latter, as the former can directly impact the downstream calculation of the lane center. For example, if a model cannot detect the left lane's left line but can still detect the ego lane's left line, it will not affect its use for the ALC. However, if it cannot detect the latter but can detect the former, the accuracy metric remains the same but the downstream modules in the ALC may consider the left lane's left line as ego lane's left line and thus mistakenly deviate to the left.

5.2.3 Automated Lane Centering

Automated Lane Centering (ALC) is a Level-2 driving automation technology that automatically steers a vehicle to keep it centered in the traffic lane [59]. Recently, ALC is widely adopted in various vehicle models such as Tesla [41] and thus one of the most popular downstream applications of lane detection. Typical ALC systems [82, 231, 29] operate in 3 modules: lane detection, lateral control, and vehicle actuation. More details of ALC are in the supplementary materials (Appendix G). While there is a line of research that designs end-to-end DNNs for ALC or higher driving automation [135, 118, 113], the current industry-standard solutions adopt such a modular design to ensure accountability and safety. In the lateral control, ALC plans to follow the lane center as waypoints with Proportional-Integral-

Derivative (PID) [161] or Model Predictive Control (MPC) [284].

Adversarial Attack on ALC. After researchers found DNN models generally vulnerable to adversarial attacks [316, 178], the following work further explored such attacks in the physical world [122, 167]. A recent study demonstrates that ALC systems are also vulnerable to physical-world adversarial attacks [291]. Their attack, dubbed Dirty Road Patch (DRP) attack, targets industry-grade DNN-based ALC systems, and is designed to be robust to the vehicle position and heading changes caused by the attack in the earlier frames. We use the DRP attack to simulate challenging but realizable scenarios in our evaluations.

5.3 Methodology

In this section, we motivate the design of 2 new downstream task-aware metrics to measure the performance of lane detection models in ALC. To evaluate the validity of the metrics even in challenging scenarios, we formulate attack objective functions to fairly generate adversarial attacks against the 4 major types of lane detection methods.

5.3.1 End-to-End Lateral Deviation Metric

As the name of ALC indicates, the performance of ALC should be evaluated by how accurately it can drive in the lane center, i.e., the *lateral* (left or right) deviation from the lane center. In particular, the maximum lateral deviation from the lane center in continuous closed-loop perception and control is the ultimate downstream-task performance metric for lane detection. Such deviation is directly safety-critical as large lateral deviations can cause a fatal collision with other driving vehicles or roadside objects. We call it *End-to-End Lateral Deviation metric* **(E2E-LD)**, shown in Fig. 5.2.1 (a). The E2E-LD at $t = 0$ is obtained as

follows.

$$\max_{t \leq T_E}(|L_t - C_t|) \tag{5.1}$$

, where L_t is the lateral (y-axis) coordinate of the vehicle at t. C_t is the lane center lateral (y-axis) coordinate corresponding to the vehicle position at t. We use the vehicle coordinate system at $t = 0$. T_E is a hyperparameter to decide the time duration. If $T_E = 1$ second, the E2E-LD is the largest deviation within one second. To obtain L_t, it requires a closed-loop mechanism to simulate driving by ALC, such as AD simulators [163, 19]. Starting from $t = 0$, the vehicle position and heading at $t = 1$ is calculated based on the camera frame at $t = 0$ (X_0): The lane detection detects lane lines from the frame, the lateral control interprets them by a steering angle, and the vehicle actuation operates the steering wheel. This process is repeated until $t = T_e$. Hence, the consecutive camera frames $X_0, ..., X_{T_E}$ are required and they are dynamically changed based on the lane detection results in the earlier frames.

However, such AD simulations are too computationally expensive for large-scale evaluations. Thus, we simulate vehicle trajectories by following prior work [291], which combines vehicle motion model [277] and perspective transformation [188, 321] to dynamically synthesize camera frames from existing frames according to a driving trajectory.

5.3.2 Per-Frame Simulated Lateral Deviation Metric

The E2E-LD metric is defined as the desired metric based on the requirements of downstream task ALC. However, it is still too computationally intensive to be monitored during training of the upstream lane detection model. This overhead is mainly due to the camera frame inter-dependency that the camera frames are dynamically changed based on the lane detection results in the earlier frames. To address this limitation, we design the *Per-Frame Simulated Lateral Deviation metric* **(PSLD)**, which simulates E2E-LD only with a *single* camera input

at the current frame (X_0) and the geometry of the lane center.

The overview of PSLD is shown in Fig. 5.2.1 (b). The calculation consists of two stages: ① update the vehicle position with the current camera frame at $t = 0$ (X_0) and its lane detection result, and ② apply the closed-loop simulation using the ground-truth lane center as waypoints from $t = 1$ to $t = T_p$. Note that we do not need camera frames in ② as the vehicle just tries to follow the ground-truth waypoints with lateral control, i.e., we bypass the lane detection assuming we know the ground-truth in $t \geq 1$. We then take the maximum lateral deviation from the lane center as a metric as with E2E-LD. For convenience, we normalize the maximum lateral deviation by T_p to make it a *per-frame* metric. The definition of PSLD is as follows:

$$\frac{1}{T_p} \max_{1 \leq t \leq T_p} (|\widetilde{L}_t - C_t|) \tag{5.2}$$

, where the $\widetilde{L}_t$ is the simulated lateral (y-axis) coordinate of the vehicle at t. For example, for $T_p = 1$, it is just a single-step simulation with the current lane detection result. The longer T_p can simulate the tailing effect of the current frame result in the later frames, but it may suffer from accumulated errors. In §5.4.3, we explore which T_p achieves the best correlation between PSLD and E2E-LD. More details are in the supplementary material (Appendix A).

5.3.3 Attack Generation

In this study, we utilize and extend physical-world adversarial attacks to evaluate the robustness of the lane detection system against challenging but realizable scenarios. To fairly generate adversarial attacks for all 4 major types of lane detection methods, we design an attack objective that can be commonly applicable to them. We name it the *expected road center*, which averages all detected lane lines weighted with their probabilities. Intuitively, the average of all lane lines is expected to represent the road center. If the expected center

locates at the center of the input image, its value will be 0.5 in the normalized image width. We maximize the expected road center to attack to the right and minimize it to attack to the left. Detailed calculation of the expected road center for each method is as follows.

Segmentation & row-wise classification approaches:

$$\frac{1}{L \cdot H} \sum_{l=1}^{L} \sum_{i=1}^{W} \sum_{j=1}^{H} i \cdot P_{ij}^{l} \tag{5.3}$$

, where H and W are the height and width of probability map, L is the number of probability maps (channels), and P_{ij}^{l} is the lane line existence probability of the pixel in the (i, j) element of the probability map.

Curve-fitting approach:

$$\frac{1}{L \cdot |\mathcal{H}|} \sum_{l=1}^{L} \sum_{j \in \mathcal{H}} [j^{d}, j^{d-1}, \cdots, j, 1] p_{l} \tag{5.4}$$

, where L is the number of detected lane lines, d is the degrees of polynomial ($d = 3$ used in PolyLaneNet [318]), $\mathcal{H}$ is a set of sampled y-axis values, and $p_l \in \mathbb{R}^{d+1}$ is the coefficient of detected lane line l.

Anchor-based approach:

$$\sum_{l \in \mathcal{A}} \left[\frac{1}{|\Delta^{l}|} \sum_{j \in \Delta^{l}} (a_{j}^{l} + \delta_{j}^{l}) \right] \cdot \pi^{l} \tag{5.5}$$

, where $\mathcal{A}$ is a set of the anchor proposals, Δ^{l} is an index set of y-axis value for anchor proposal l, π^{l} is the probability of anchor proposal l, and a_{j}^{l} and δ_{j}^{l} are the x-axis value and its offset of anchor proposal l at y-axis index j respectively.

We incorporate this expected road center functions into DRP attack [291] procedure to

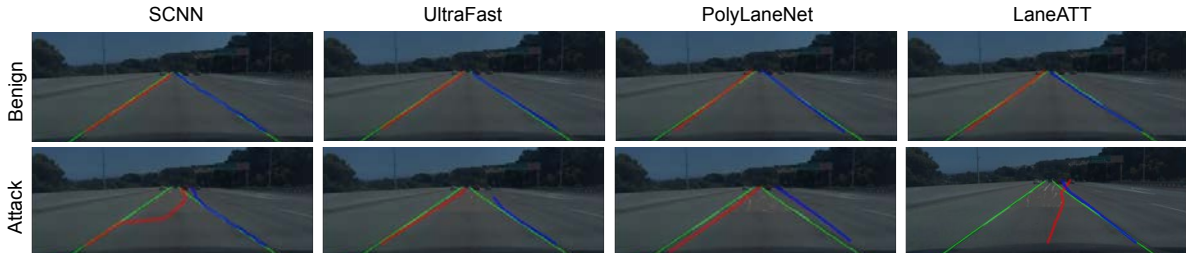


Figure 5.4.1: Examples of the benign and attack-to-the-right scenarios on the Comma2k19-LD dataset. The red, blue, and green lines are the detected left and right lines and the ground-truth lines respectively.

Table 5.4.1: Target lane detection methods. *Acc.* is the accuracy of the TuSimple Challenge dataset [62] in the reference papers.

Approach	Selected Method	Acc.
Segmentation	SCNN [267]	96.53%
Row-wise classification	UltraFast (ResNet18) [274]	95.87%
Curve-fitting	PolyLaneNet (b0) [318]	88.62%
Anchor-based	LaneATT (ResNet34) [317]	95.63%

generate adversarial attacks that are effective for multiple frames.

5.4 Experiments

We conduct a large-scale empirical study to evaluate the validity of the conventional metrics and our PLSD by comparing them with the ultimate downstream-task performance metric E2E-LD. We evaluate the 4 major types of lane detection approaches. We select a representative model for each approach as shown in Table 5.4.1. The pretrained weights of all models are obtained from the authors' or publicly available websites[2]. All pretrained weights are trained on the TuSimple Challenge training dataset [62].

[2] LaneATT https://github.com/lucastabelini/LaneATT
SCNN https://github.com/harryhan618/SCNN_Pytorch
UltraFast https://github.com/cfzd/Ultra-Fast-Lane-Detection
PolyLaneNet https://github.com/lucastabelini/PolyLaneNet

5.4.1 Conventional Evaluation on TuSimple Dataset

Evaluation Setup. We first evaluate the lane detection models with the conventional accuracy and F1 score metrics on the TuSimple dataset[62], which has 2,782 one-second-long video clips as test data. Each clip consists of 20 frames, and only the last frame is annotated and used for evaluation. We randomly select 30 clips from the test data. For each clip, we consider two attack scenarios: attack to the left, and to the right. Thus, in total, we evaluate 60 different attack scenarios. In each scenario, we place 3.6 m x 36 m patches 7 m away from the vehicle as shown in Fig. 5.1.1. To know the world coordinate, we manually calibrate the camera matrix based on the size of lane width and lane marking. To deal with the limitation (2) discussed in §5.2.2, we remove lane lines other than the ego-left and ego-right lane lines to evaluate the applicability to ALC systems more correctly. More details of each attack implementation and parameters are in Appendix B.

Results. Table 5.4.2 shows the accuracy and F1 score metrics in the benign and attacks scenarios. In the benign scenarios, LaneATT has the best accuracy (94%) and F1 score (88%). SCNN and UltraFast show also high accuracy and F1 score while UltraFast has the lowest F1 score (8%) in the attack scenarios. PolyLaneNet has lower accuracy and F1 score than the others in both benign and attack scenarios. These results are generally consistent with the reported performance as in Table 5.4.1. However, when we visually look into the detected lane lines under attack, we find quite some cases suggesting vastly different conclusions if used in AD as the downstream task. For example, as shown in Fig. 5.1.1, even though SCNN has the highest accuracy in all three scenarios, its detected lane lines are heavily curved by the attack. In contrast, the detection of PolyLaneNet looks like the most robust among the 4 models, as the detected lane lines are generally parallel to the actual lane lines. However, its accuracy (63%) is smaller than the one of SCNN (51%) in the attack to the right scenario. In the benign scenario, PolyLaneNet has a lower accuracy (16% margin) than the others, but it is hard to find meaningful differences for humans as

Table 5.4.2: Accuracy and F1 scores for attack and benign cases on the TuSimple Challenge dataset. The metrics are calculated only with ego left and right lanes. The **bold** and underlined letters mean the highest and lowest scores, respectively, among the 4 lane detection methods. The higher score means the higher performance.

	Accuracy		F1 Score	
	Benign	Attack	Benign	Attack
SCNN [267]	89%	**58%**	75%	28%
UltraFast [274]	87%	36%	77%	8%
PolyLaneNet [318]	72%	53%	50%	19%
LaneATT [317]	**94%**	51%	**88%**	**29%**

Table 5.4.3: Evaluation results of the E2E-LD and the conventional metrics, accuracy and F1 in the benign and attack scenarios. For each metric, the corresponding Pearson correlation coefficient with E2E-LD in the bottom rows. The original parameters are the ones used in the TuSimple challenge. The best parameters are those that have the highest correlation between E2E-LD with respect to F1 score. The **bold** and underlined letters indicate the highest and lowest performance or correlation, respectively.

		Benign					Attack				
			Original Parameters ($\alpha = 20, \beta = 0.85$)		Best Parameters ($\alpha = 5, \beta = 0.9$)			Original Parameters ($\alpha = 20, \beta = 0.85$)		Best Parameters ($\alpha = 50, \beta = 0.65$)	
		E2E-LD [m]	Accuracy	F1	Accuracy	F1	E2E-LD [m]	Accuracy	F1	Accuracy	F1
Metric	SCNN [267]	0.21	**0.93**	**0.84**	**0.59**	0.03	0.48	**0.68**	**0.31**	**0.83**	0.76
	UltraFast [274]	**0.18**	0.92	0.81	0.55	**0.10**	0.58	0.60	0.21	0.82	**0.77**
	PolyLaneNet [318]	0.20	0.78	0.50	0.44	0.01	**0.38**	0.59	0.13	0.81	0.76
	LaneATT [317]	0.21	0.89	0.75	0.54	0.06	0.72	0.51	0.14	0.66	0.48
Corr.	SCNN [267]	-	-0.65^{***}	-0.60^{***}	$\mathbf{-0.33}^{***}$	-0.13^{ns}	-	-0.13^{ns}	$\mathbf{-0.06}^{ns}$	-0.14^{ns}	-0.06^{ns}
	UltraFast [274]	-	-0.58^{***}	-0.59^{***}	-0.38^{***}	-0.24^{*}	-	-0.24^{*}	-0.14^{ns}	-0.20^{*}	-0.13^{ns}
	PolyLaneNet [318]	-	-0.60^{***}	$\mathbf{-0.55}^{***}$	-0.46^{***}	$\mathbf{0.10}^{ns}$	-	-0.27^{**}	-0.28^{**}	-0.06^{ns}	$\mathbf{0.01}^{ns}$
	LaneATT [317]	-	$\mathbf{-0.57}^{***}$	-0.58^{***}	-0.34^{***}	-0.14^{ns}	-	$\mathbf{0.08}^{ns}$	-0.09^{ns}	$\mathbf{0.11}^{ns}$	0.12^{ns}

^{ns} Not Significant ($p > 0.05$),* $p \leq 0.05$,** $p \leq 0.01$,*** $p \leq 0.001$

the detected lines are well-aligned with actual lane lines. We provide more examples in the supplementary material (Appendix G). Hence, the conventional accuracy and F1 score-based evaluation may not be well suitable to judge the performance of the lane detection model in representative downstream tasks such as AD.

5.4.2 Consistency of TuSimple Metrics with E2E-LD

To more systematically evaluate the consistency of the conventional accuracy and F1 score with the performance in AD as the downstream tasks, we conduct a large-scale empirical study on our newly-constructed dataset.

New Dataset: Comma2k19-LD. To evaluate both the conventional metrics and the

downstream task-centric metrics E2E-LD and PSLD on the same dataset, we need both lane line annotations and driving information (e.g., position, steering angle, and velocity). Unfortunately, there is no existing dataset that satisfies the requirements to our best knowledge. Thus, we create a new dataset, coined *Comma2k19-LD*, in which we manually annotate the left and right lane lines for 2,000 frames (100 scenarios of 1-second clips at 20 Hz). The selected scenarios are randomly selected from the scenarios with more than 30 mph ($\approx$ 48 km/h) in the original Comma2k19 dataset [294]. Fig 5.4.1 shows the example frames of the Comma2k19-LD dataset. These frames are the first frames of the scenario. The following 20 frames are also annotated and the same patch is used for each attack. More details are in Appendix C. The Comma2k19-LD dataset is published on our website[91].

Evaluation Setup. We conduct the evaluation on the Comma2k19-LD dataset. For the attack generation, we attack to the left in randomly selected 50 scenarios and attack to the right in the other 50 scenarios. For the lateral control, we use the implementation of MPC [284] in OpenPilot v0.6.6, which is an open-source production ALC system. For the longitudinal control, we used the velocity in the original driving trace. For the motion model, we adopt the kinematic bicycle model [225], which is the most widely-used motion model for vehicles [16, 225, 343]. The vehicle parameters are from Toyota RAV4 2017 (e.g., wheelbase), which is used to collect the traces of the comma2k19 dataset. To make the model trained on the TuSimple dataset work on the Comma2k19-LD dataset, we manually adjust the input image size and field-of-view to be consistent with the TuSimple dataset. We place a 3.6 m x 36 m patch at 7 m away from the vehicle at the first frame. For the E2E-LD metric, we use $T_E = 20$ frames (1 second). It follows the result that the average attack success time of the DRP attack is nearly 1 sec [291]. More setup details are in Appendix B, D, and G.

Results. Table 5.4.3 shows the evaluation results of conventional accuracy and F1 score and E2E-LD. We calculate the Pearson correlation coefficient r and its p value. As shown, there are *substantial inconsistencies between the downstream-task performance (from the heavy-*

weight E2E-LD metric) and the conventional metrics. In the benign scenarios, SCNN has the highest accuracy (0.59) and F1 score (0.84) under the original parameters ($\alpha = 20, \beta = 0.85$). However, SCNN is one of the methods with the *lowest* E2E-LD (0.21), and instead UltraFast has the highest E2E-LD (0.18). In the attack scenarios, the inconsistency is more obvious: PolyLaneNet has the highest E2E-LD (0.38), but PolyLaneNet achieves the 2nd lowest accuracy (0.59) and the highest F1 score (0.13) with the original parameters. Hence, the E2E-LD draws quite different conclusions from the conventional metrics. If we adopt the conventional metrics, SCNN should be preferred as the best performant model. This is consistent with the results in Table 5.4.1 and §5.4.1 since SCNN, UltraFast, and LaneATT show close performance in the conventional metrics (SCNN may have slight advantages in Comma2k19-LD). On the other hand, if we adopt E2E-LD, PolyLaneNet should be preferred since there is only a slight difference between the 4 lane detection methods in the benign scenarios and PolyLaneNet clearly outperforms the other methods in the attack scenarios.

The inconsistency between the E2E-LD and the conventional metrics can be more systematically quantified using Pearson correlation coefficient r. Generally, the E2E-LD and the conventional metrics have strongly *negative* correlations ($r \leq$-0.55) with high statistical significance ($p \leq 0.001$), meaning that some recent improvements in the conventional metrics may not have led to improvements in AD, but rather may have made it worse by overfitting to the metrics. SCNN, the segmentation approach, is the only one that does not use domain knowledge, e.g., lane lines are smooth lines (§5.2.1). This high degree of freedom in the model may lead to overfitting of the human annotations with noise.

Finally, we evaluate the parameters in the conventional metrics: α for the accuracy and β for F1 score. For α, we explore every 5 pixels from 5 pixels to 50 pixels. For β, we explore every 0.05 from 0.5 to 0.9. In the benign scenarios, ($\alpha = 20, \beta = 0.85$) has the best correlation between the E2E-LD and F1 score. In the attack scenarios, ($\alpha = 50, \beta = 0.65$) has the best correlation between the E2E-LD and F1 score. However, the results are still similar to those

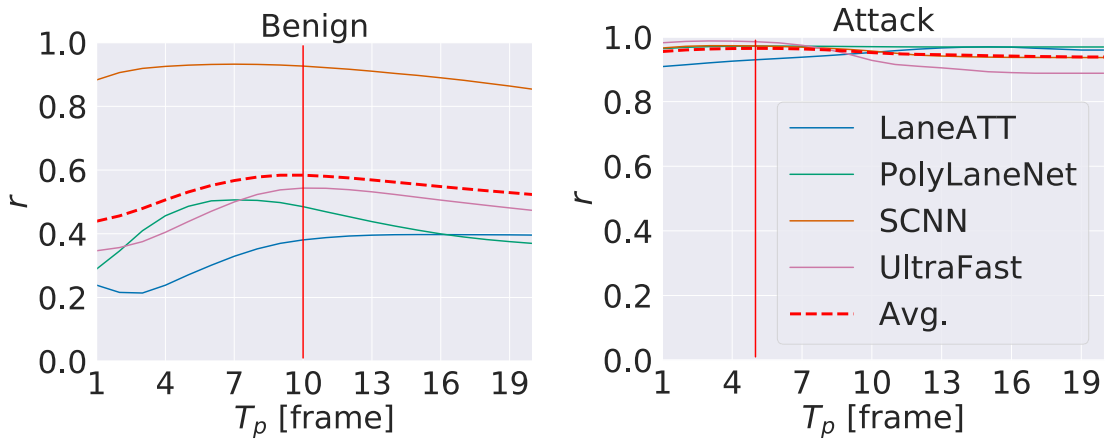


Figure 5.4.2: Pearson correlation coefficient r between E2E-LD and PSLD when T_p is varied from 1 to 20 in the benign and attack scenarios. The red vertical lines are T_p with the largest average r.

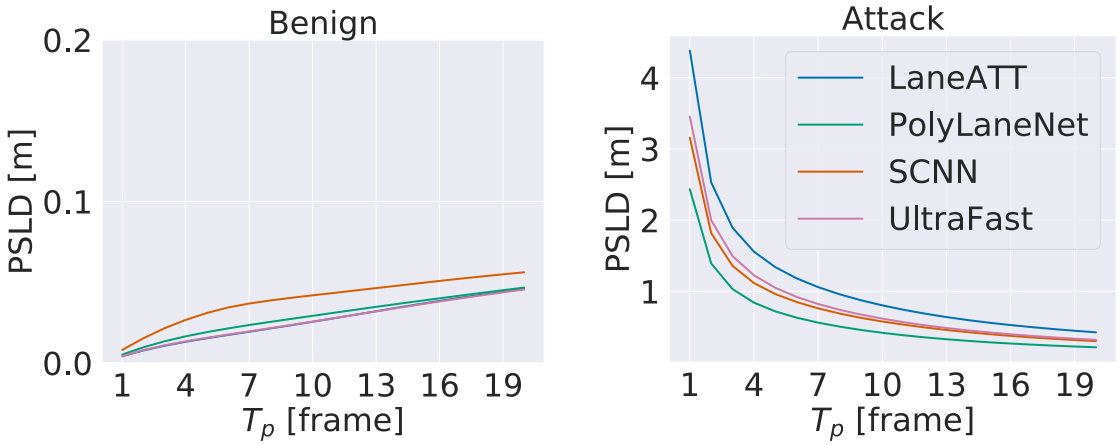


Figure 5.4.3: PSLD for the 4 major lane detection models when T_p is varied from 1 to 20 frames in the benign and attack scenarios.

using the original parameters: SCNN shows the highest accuracy; UltraFast has a higher F1 score than the others, but the correlation is still negative. Thus, such a naive parameter tuning does not resolve the limitations of the conventional metrics.

5.4.3 Consistency of E2E-LD with PSLD

In this section, we evaluate the validity of PSLD as a *per-frame* surrogate metric of E2E-LD.

Evaluation Setup. We follow the same setup as in §5.4.2. We generate the DRP attacks for 100 scenarios in the Comma2k19-LD dataset with the same parameters. For the PSLD, we obtain the ground truth waypoints by the following procedure. We generate a trajectory with the bicycle model and OpenPilot's MPC by using the human driving trajectory as waypoints. We then use the generated trajectory as a ground-truth road center. While we can directly use the human driving trajectory as ground truth, human driving sometimes is not smooth and this approach can cancel the effect of motion models, which have differences from real vehicle dynamics. For the benign scenarios, we calculate the PSLD for each frame in the original human driving. For the attack scenarios, we use the frames synthesized by the method described in 5.3.1 instead of the original frames because the attacked trajectory and its camera frames are largely changed from the original human driving. For example, to obtain the PSLD at frame $t = N$, we simulate the trajectory until $t = N - 1$ and we then calculate the PSLD with the synthesized frame at $t = N$.

Results. Fig. 5.4.2 shows the Pearson correlation coefficient r between E2E-LD and PSLD when T_p is varied from 1 to 20 frames. As shown, the E2E-LD, PSLD has strong positive correlations in both benign and attack scenarios. In particular, there are significant correlations (>0.8) in the attack scenarios. This is because the direction of lateral deviation generally coincides with the attack direction. By contrast, in the benign scenarios, the vehicle drives around the road center with overshooting, and thus the direction of lateral deviation heavily depends on the initial states. Nevertheless, the PSLD has always high positive correlations with E2E-LD (>0.2). In particular, SCNN has strong correlations (>0.8) with E2E-LD in all T_p. We consider that the high correlation can be due to the segmentation approach, which is the only method among the 4 methods that do not use the domain-specific knowledge the lane lines are generally smooth (§5.2.1). The detection of SCNN at the same location tends to be consistent across different frames, i.e., SCNN is less dependent on global information.

Finally, we explore the best T_p for PSLD to proxy E2E-LD. As shown in Fig. 5.4.2, the

Table 5.4.4: Evaluation results of the E2E-LD and PSLD in the benign and attack scenarios. The format is the same as Table 5.4.3.

		Benign		Attack	
		E2E-LD [m]	PSLD [m]	E2E-LD [m]	PSLD [m]
Metric	SCNN [267]	<u>0.21</u>	<u>0.04</u>	0.48	0.58
	UltraFast [274]	**0.18**	**0.03**	0.58	0.62
	PolyLaneNet [318]	0.20	**0.03**	**0.38**	**0.42**
	LaneATT [317]	<u>0.21</u>	**0.03**	<u>0.72</u>	<u>0.80</u>
Corr.	SCNN [267]	-	**0.93*****	-	0.96***
	UltraFast [274]	-	0.54***	-	<u>0.93</u>***
	PolyLaneNet [318]	-	0.49***	-	**0.97*****
	LaneATT [317]	-	<u>0.38</u>***	-	0.95***

^{ns} Not Significant ($p > 0.05$),* $p \leq 0.05$,** $p \leq 0.01$,*** $p \leq 0.001$

average of the correlation coefficients of the 4 methods achieves the maximum at $T_p = 10$ in the benign scenarios and $T_p = 5$ in the attack scenarios respectively. We list the E2E-LD and PLSD with $T_p = 10$ and the corresponding r in Table 5.4.4. As shown, there are strong, statistically significant ($p \leq 0.001$) positive correlations (≥ 0.38) between E2E-LD and PSLD in both cases. The results strongly support the fact that PSLD can measure the performance of lane detection in ALCs based solely on the single camera frame and ground-truth road center geometry. We note that the PSLD is not so sensitive to the choice of T_p. As shown in Fig. 5.4.3, the magnitude relation of the 4 methods is generally consistent for all T_p.

5.5 Discussion

Alternative Metric Design. To improve the existing metrics, we explored other possible design choices. One of the most intuitive approaches is the $\mathcal{L}_1$ or $\mathcal{L}_2$ distance in the bird's eye view. We evaluated the designs and confirmed that these metrics are still leading to erroneous judgment on downstream AD performance similar to the conventional metrics. Details are in the supplementary materials (Appendix F). We note that our metrics are specific to AD, the main downstream task of lane detection. For other downstream tasks, other metric designs can be more suitable.

Domain Shift. In this work, we use lane detection models pretrained on the TuSimple dataset and evaluate them on the Comma2k19-LD. To evaluate the impact of domain shift, we conduct further evaluation and confirm that our observations are generally consistent. Detailed results and discussions are in the supplementary material (Appendix E).

Closed-loop Simulation. To obtain driving-oriented metrics, there are multiple parameters and design choices in the closed-loop simulation. In this study, we follow the parameters in the Comma2k19 datasets and select simple and popular designs, e.g., bicycle model and MPC. Meanwhile, we think that such design differences should only have minor effects on our observations because ALC, Level-2 driving automation, just follows the lane center line, which is designed to be smooth on normal roads.

Evaluation on Other Datasets. Our metrics are applicable to any dataset set that contains position data and its camera frames, but ideally, velocity and ground-truth lane centers should be available. Such information is available in relatively new datasets such as [78, 137]. However, lane annotations are not directly available in the datasets and require considerable effort to obtain from map data and camera frames. To our knowledge, our Comma2k19-LD is so far the only dataset with both lane line annotation and driving information. We hope our work will facilitate further research to build datasets including them.

5.6 Conclusion

In this work, we design 2 new lane detection metrics, E2E-LD and PSLD, which can more faithfully reflect the performance of lane detection models in autonomous driving. Throughout a large-scale empirical study of the 4 major types of lane detection approaches on the TuSimple dataset and our new dataset Comma2k19-LD, we highlight critical limitations of the conventional metrics and demonstrate the high validity of our metrics to measure the

performance in autonomous driving, the core downstream task of lane detection. In recent years, a wide variety of pretrained models have been used in many downstream application areas such as autonomous driving [7], natural language processing [159], and medical [140]. Reliable performance measurement is essential to facilitate the use of machine learning responsibly. We hope that our study will help the community make further progress in building a more downstream task-aware evaluation for lane detection.

Appendix

5.A Detailed Settings of Lane Detection Metrics

Table 5.A.1 shows the parameters and input required to calculate each metric for lane detection. As shown, only E2E-LD requires multiple frames for computation. For E2E-LD and PSLD, we adopt the kinematic bicycle model [225] to simulate the vehicle motion. The only parameter in the kinematic bicycle model is the wheelbase. We use WheelBase=2.65 meters, which is the wheelbase of Toyota RAV4 2017.

Table 5.A.1: Parameter and input of metrics for Lane Detection

	Parameter	Input	Per-frame
Accuracy	α	X_0	✓
F1 score	α, β	X_0	✓
E2E-LD	T_E, WheelBase	$X_0, ..., X_{T_E}, C$	
PSLD	T_p, WheelBase	X_0, C	✓

5.B Detailed Attack Implementation

We use the official implementation of the DRP attack [291]. We also use parameters that are reported to have the best balance between effectiveness and secrecy: the learning rate is 10^{-2}, the regularization parameter λ is 10^{-3}, and the perturbable area ratio (PAR) is 50%. We run 200 iterations to generate the patch in all experiments.

5.C Details of Comma2k19-LD dataset

Fig. 5.C.1 shows the first frame of all 100 scenarios and its lane line annotations. For annotation, human annotators mark the lane line points and check if the linear interpolation results of the markings align with the lane line information. To convert the annotations to the TuSimple dataset format, we sample points every 10 pixels in the y-axis from the interpolated results.

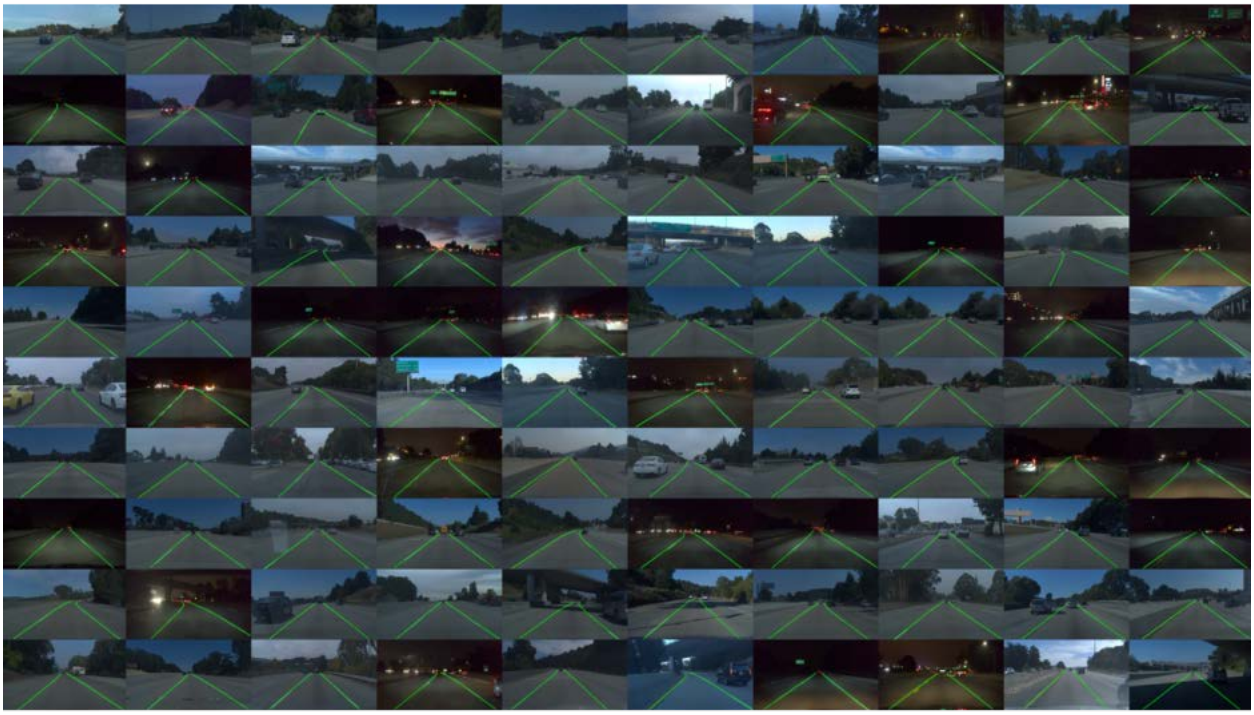
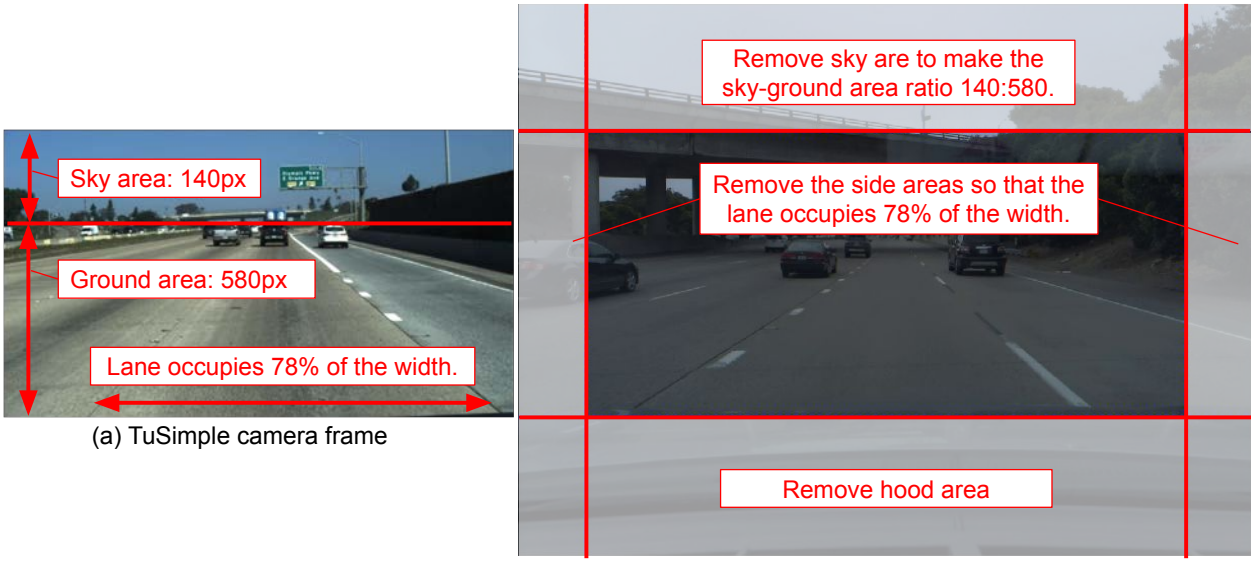


Figure 5.C.1: The first frame of all 100 scenarios and its lane line annotations (green line)

5.D Adaptation to TuSimple Challenge Camera Frames Geometory

In the evaluation of the comma2k19-LD dataset, we use the same pretrained models trained on the TuSimple Challenge training dataset. To deal with the differences in the datasets, we convert the camera frames in the comma2k19-LD dataset to have geometry similar to the camera frames in the TuSimple challenge dataset. Fig. 5.D.1 illustrates the overview of the conversion. We remove the surrounding area and use only the central part of the Comma2k19-LD camera frame to have the same sky-ground area ratio and the same lane occupation ratio in the image width with the ones in the TuSimple dataset.



(a) TuSimple camera frame

(b) Comma2k19 camera frame

Figure 5.D.1: Overview of adapting the camera frames in Comma2k19-LD dataset to the camera frame in the TuSimple Challenge dataset. We remove the surrounding area and use only the central part of the Comma2k19 camera frame to ensure that the comma2k19-LD camera frames have a geometry similar to that of the TuSimple challenge camera frames.

5.E Evaluation of the Domain Shift Effect

In this study, we use lane detection models pretrained on the TuSimple dataset and evaluate them on the Comma2k19-LD dataset. Although both datasets are similar driver's view images, there can be some domain shifts between them. To understand the impact, we trained the 4 models with another 100 scenarios extracted from the Comma2k19 dataset. We run 10 epochs with the data on top of the models pretrained on the TuSimple dataset. For the lane line labels, we use OpenPilot's lane detection results in the dataset. We conduct the same evaluation in §4.2 and §4.3 of the main paper. As shown in Table 5.E.1 and Table 5.E.2, the observations are consistent: SCNN outperforms in the conventional metrics; PolyLaneNet is the most robust in attack scenarios. The Pearson correlation coefficients show almost the same results as the ones in §4.2 and §4.3 that the conventional metrics have strong negative correlations in the benign scenarios and the correlations in the attack scenarios are not statistically significant.

However, the E2E-LD in the attack scenarios are generally higher than the results in §4.2 and §4.3 of the main paper while the E2E-LD in the benign scenarios is generally lower. This indicates that this additional fine-tuning improves the performance in the benign scenarios, but it harms the robustness against adversarial attacks.

5.F Alternative metric design

To improve the conventional metrics, one of the most intuitive approaches is the $\mathcal{L}_1$ or $\mathcal{L}_2$ distance in the bird's eye view because they do not suffer from the problem of the ill parameters discussed in §2.2, and lane detection results from a bird's eye view may be a more adequate to measure of drivability than detection results from a front camera. We actually have considered such metrics before, but we did not finally choose them because, without

Table 5.E.1: Evaluation results of the E2E-LD and the conventional metrics, accuracy and F1 in the benign and attack scenarios. For each metric, the corresponding Pearson correlation coefficient with E2E-LD in the bottom rows. The **bold** and underlined letters indicate the highest and lowest performance or correlation, respectively.

		Benign			Attack		
			Original Parameters ($\alpha = 20, \beta = 0.85$)			Original Parameters ($\alpha = 20, \beta = 0.85$)	
		E2E-LD [m]	Accuracy	F1	E2E-LD [m]	Accuracy	F1
Metric	SCNN [267]	0.20	**0.93**	0.84	**0.52**	**0.67**	0.30
	UltraFast [274]	0.18	0.92	0.84	0.62	0.49	0.16
	PolyLaneNet [318]	**0.13**	**0.93**	**0.86**	0.54	0.62	**0.33**
	LaneATT [317]	0.14	**0.93**	0.85	0.71	0.51	0.12
Corr.	SCNN [267]	-	-0.65***	-0.51***	-	-0.09^{ns}	-0.04^{ns}
	UltraFast [274]	-	-0.63***	-0.60***	-	**0.14**^{ns}	**0.07**^{ns}
	PolyLaneNet [318]	-	**-0.32*****	-0.62***	-	**0.14**^{ns}	0.04^{ns}
	LaneATT [317]	-	-0.57***	**-0.26*****	-	-0.02^{ns}	-0.06^{ns}

^{ns} Not Significant ($p > 0.05$),* $p \leq 0.05$,** $p \leq 0.01$,*** $p \leq 0.001$

Table 5.E.2: Evaluation results of the E2E-LD and PSLD in the benign and attack scenarios. The format is the same as Table 5.4.3.

		Benign		Attack	
		E2E-LD [m]	PSLD [m]	E2E-LD [m]	PSLD [m]
Metric	SCNN [267]	0.20	0.04	**0.52**	0.61
	UltraFast [274]	0.18	**0.02**	0.62	0.66
	PolyLaneNet [318]	**0.13**	**0.02**	0.54	**0.55**
	LaneATT [317]	0.14	0.03	0.71	0.82
Corr.	SCNN [267]	-	**0.93*****	-	0.93***
	UltraFast [274]	-	0.60***	-	**0.99*****
	PolyLaneNet [318]	-	0.65***	-	**0.99*****
	LaneATT [317]	-	0.55***	-	0.78***

^{ns} Not Significant ($p > 0.05$),* $p \leq 0.05$,** $p \leq 0.01$,*** $p \leq 0.001$

some form of control simulation, we find it fundamentally nontrivial to accurately predict the combined effects of detection errors at different lane line positions and with different error amounts on the downstream AD driving. This can be concretely shown in Table 5.F.1. As shown, both such 3D-L1 and 3D-L2 distance metrics have considerably lower correlation coefficient r with E2E-LD compared to our PSLD. They are indeed better than conventional accuracy and F1 score metrics. However, they are still leading to erroneous judgment on downstream AD performance similar to the accuracy and F1 score: e.g., PolyLaneNet is 2nd-worst based on 3D-L1/L2 distance metrics in the attack scenarios, but in E2E-LD it is the best. With our PSLD, such judgment is strictly consistent with E2E-LD (Table 5.4.4).

One reason we observe is that the 3D-L1/L2 metrics can be greatly biased by farther points; those points by design have much less impact on the downstream AD control, but suffer from more detection errors (due to the far distance). One thought is to assign smaller weights to farther points, but how to systematically decide such weights without any form of control simulation is fundamentally nontrivial. Additionally, such a weight-based design can still be fundamentally limited in achieving sufficient AD control relating capabilities.

Table 5.F.1: Pearson correlation coefficient r with E2E-LD. *3D-L1/L2* denote the L1/L2 distances in 3D space following Reviewer 1's suggestion. **Bold** and underline denote highest and lowest scores.

	Benign			Attack		
	PSLD (*ours*)	3D-L1	3D-L2	PSLD	3D-L1	3D-L2
SCNN	**0.93**	0.71	0.65	**0.96**	0.38	0.34
UltraFast	**0.54**	0.24	0.19	**0.93**	0.24	0.21
PolyLaneNet	**0.49**	0.47	0.44	**0.97**	0.33	0.38
LaneATT	**0.38**	0.23	0.17	**0.95**	0.23	0.23
Average	**0.59**	0.41	0.36	**0.95**	0.29	0.29

5.G Details of OpenPilot ALC and its integration with lane detection models

In this section, we explain the details of OpenPilot ALC [29] and the details of its integration with the 4 lane detection models we evaluate in this study. As in [291], the OpenPilot ALC system consists of 3 steps: lane detection, lateral control, and vehicle actuation.

5.G.1 Lane detection

The image frame from the front camera is input to the lane detection model in every frame (20Hz). Since the original OpenPilot lane detection model is a recurrent neural network model, the recurrent input from the previous frame is fed to the model with the image.

All 4 models used in this study do not have a recurrent structure, i.e., they detect lanes only in the current frame. This is because the TuSimple Challenge has a runtime limit of less than 200 ms for each frame. Another famous dataset, CULane [267], does not provide even continuous frames. In autonomous driving, the recurrent structure is a reasonable choice since past frame information is always available. Hence, the run-time calculation latency imposed in the TuSimple challenge is one of the gaps between the practicality for autonomous driving and the conventional evaluation.

5.G.2 Lateral control

Based on the detected lane line, the lateral control decides steering angle decisions to follow the lane center (i.e., the desired driving path or waypoints) as much as possible. The original OpenPilot model outputs 3 line information: left lane line, right lane line, and driving path. The desired driving path is calculated as the average of the driving path and the center line of the left and right lane lines. The steering decision is decided by the model predictive control (MPC) [284]. The detected lane lines are represented in the bird's-eye-view (BEV) because the steering decision needs to be decided in a world coordinate system.

On the contrary, all 4 models used in this study detect the lane lines in the front-camera view. We thus project the detected lane lines into the BEV space with perspective transformation [188, 321]. The transformation matrix for this projection is created manually based on road objects such as lane markings, and then calibrated to be able to drive in a straight lane. We create the transformation matrix for each scenario as the position of the camera and the tilt of the ground are different for each scenario. The desired driving path is calculated by the average of the left and right lane lines and fed to the MPC to decide the steering angle decisions.

In addition to the desired driving path, the MPC receives the current speed and steering

angle to decide the steering angle decisions. For the steering angle, we use the human driver's steering angle in the Comma2k19 dataset in the first frame. In the following frames, the steering angle is updated by the kinematic bicycle model [277], which is the most widely-used motion model for vehicles. For the vehicle speed, we use the speed of human driving in the in the comma2k19 dataset as we assume that the vehicle speed is not changed largely in the free-flow scenario, in which a vehicle has at least 5–9 seconds clear headway [119].

5.G.3 Vehicle actuation

The step sends steering change messages to the vehicle based on the steering angle decisions. In OpenPlot, this step operates at 100 Hz control frequency. As the lane detection and lateral control outputs the steering angle decisions in 20 Hz, the vehicle actuation sends 5 messages every steering angle decision. The steering changes are limited to a maximum value due to the physical constraints of vehicle and for stable and for stability and safety. In this study, we limit the steering angle change to 0.25° following prior work, which is the steering limit for production ALC systems [291].

We update the vehicle states with the kinematic bicycle model based on the steering change. Note that like all motion models, the kinematic bicycle model does have approximation errors to the real vehicle dynamics [225]. However, more accurate motion models require more complex parameters such as vehicle mass, tire stiffness, and air resistance [25]. In this study, since our focus is on understanding the impact of lane detection model robustness on end-to-end driving, the most widely-used kinematic bicycle model is a sufficient choice for simulating closed-loop control behaviors.

5.H Additional Discussions and Results

5.H.1 Additional Discussions

Ground-Truth Road Center. We obtain the ground truth waypoints based on the human driving traces. Ideally, the waypoints should be obtained by measuring roads. However, since this study focuses on the general trends of the 4 lane detection approaches, we consider that the impact of this factor should not have a major effect. If you want to use PSLD to capture more subtle differences between models, the ground truth should be more accurate.

Differentiable PSLD Regularization. We show that PSLD works as a good surrogate for E2E-LD. Next, we may want to minimize this metric directly in the model training. Since the only non-differentiable computation in PSLD is the lateral controller, we can replace this part with a differentiable controller [324, 105] and incorporate it as a regularization term in the loss function for training. Detailed study of this problem is left to future work.

5.H.2 TuSimple Challenge Dataset

Fig. 5.H.1 shows the examples of lane detection results and the accuracy metric in benign scenarios on the TuSimple Challenge dataset [62]. The limitations of the conventional metrics can be found in benign cases as well. As shown, SCNN has always higher accuracy than PolyLaneNet (at most 18% edge). Such a large leading edge is across the dataset as in Table 5.4.2 (89% vs 72% in Accuracy, 75% vs 50% in F1 score). However, for downstream AD their performances are almost the same, with PolyLaneNet actually slightly better (Table 5.4.3 of the main paper).

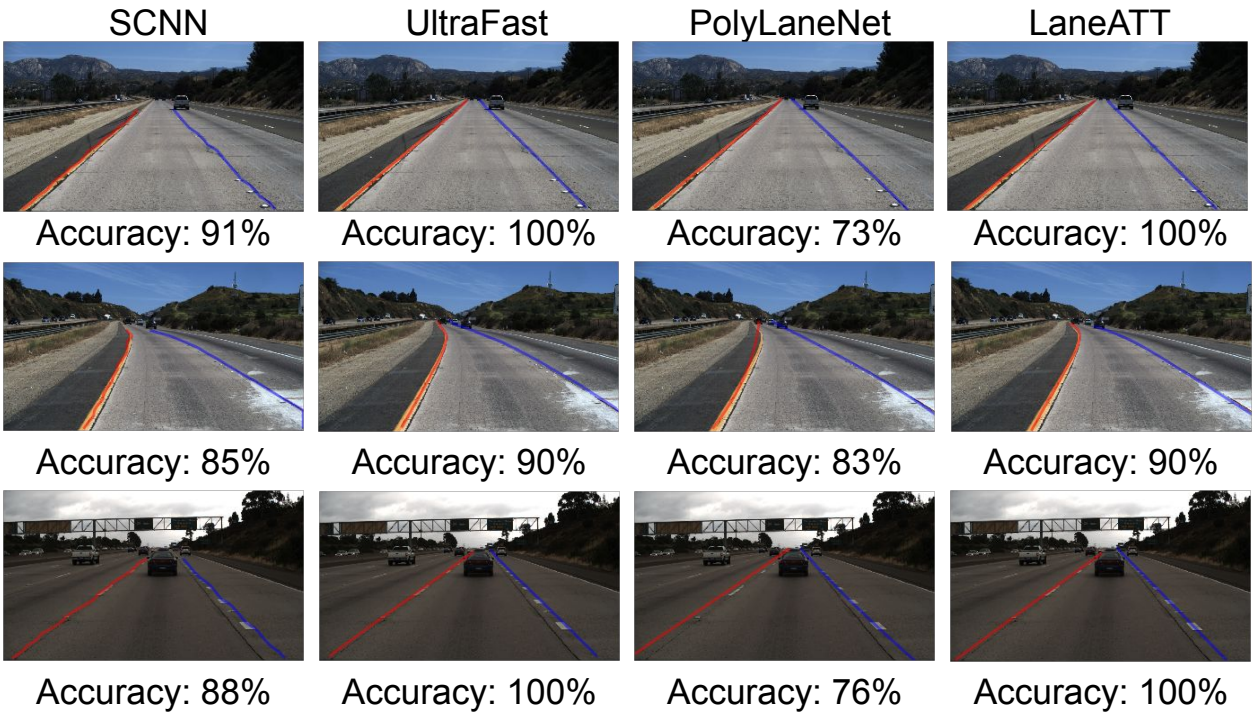


Figure 5.H.1: Examples of lane detection results and the accuracy metric in benign scenarios on TuSimple Challenge dataset [62]. As shown, the conventional accuracy metric does not necessarily indicate drivability if used in autonomous driving

5.H.3 Comma2k19 LD Dataset

We synthesize font-camera frames with a vehicle motion model [277] and perspective transformation [188, 321]. Fig. 5.H.2, 5.H.3, 5.H.4, 5.H.5, show the first 20 frames under attack and their detection results of the 4 lane detection methods, respectively. As shown, the generated images are generally complete and the distortion is very slight.

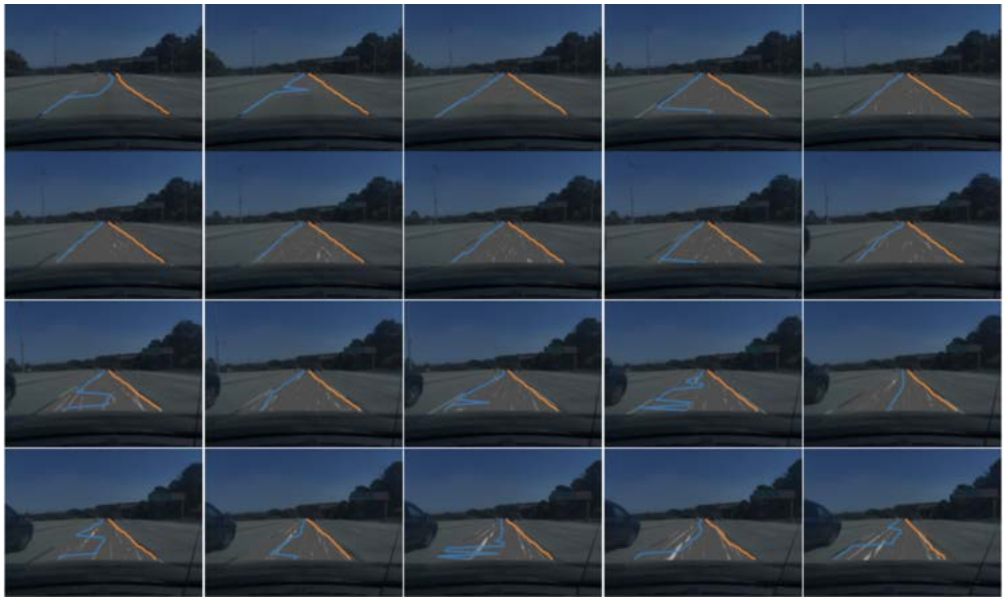


Figure 5.H.2: The first 20 frames (from left-top to right-bottom) of an attack scenario on **SCNN**. The vehicle is deviating to right due to the attack.

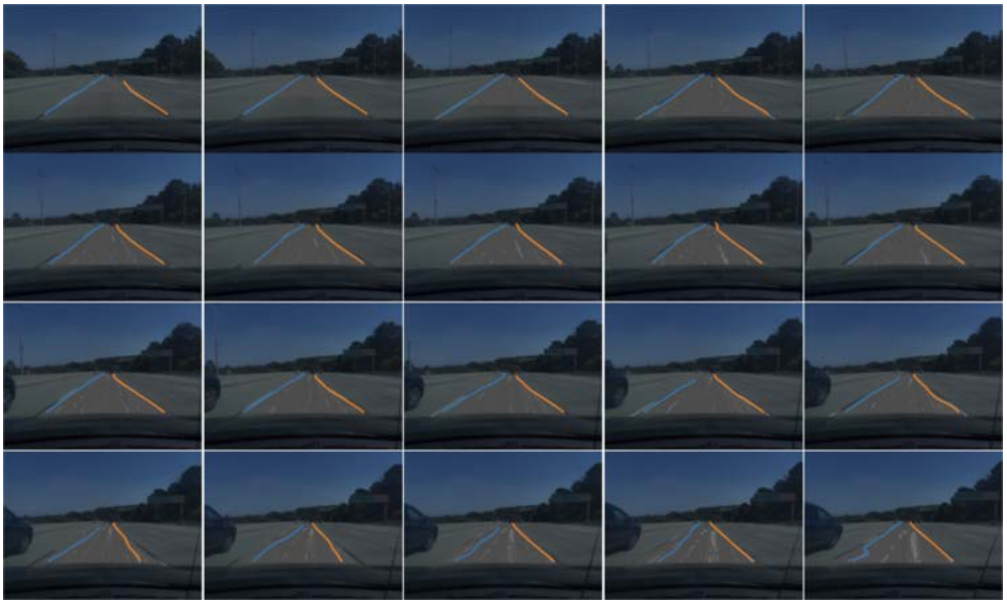


Figure 5.H.3: The first 20 frames (from left-top to right-bottom) of an attack scenario on **UltraFast**. The vehicle is deviating to right due to the attack.

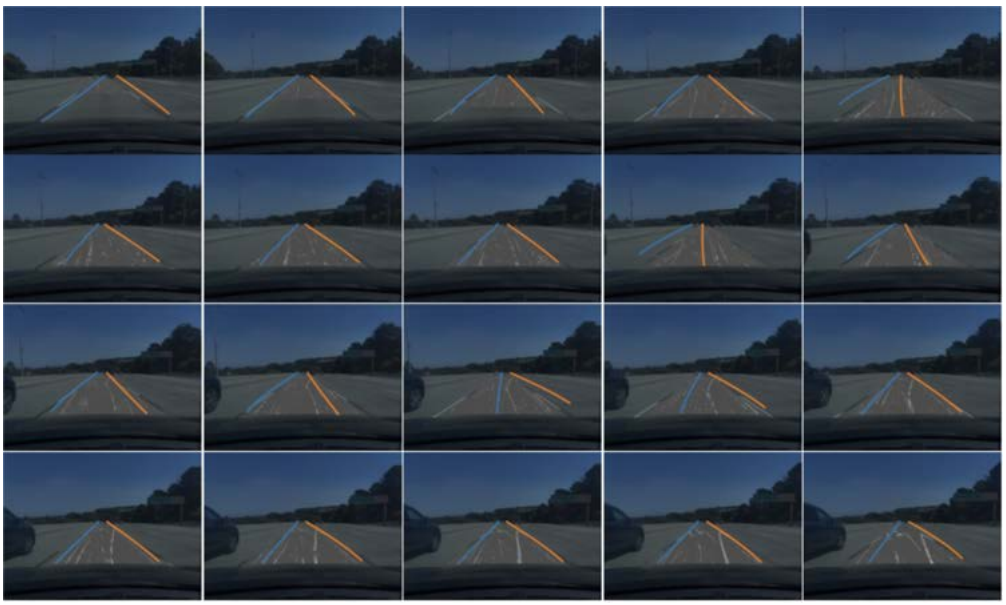


Figure 5.H.4: The first 20 frames (from left-top to right-bottom) of an attack scenario on **PolyLaneNet**. The vehicle is deviating to right due to the attack.

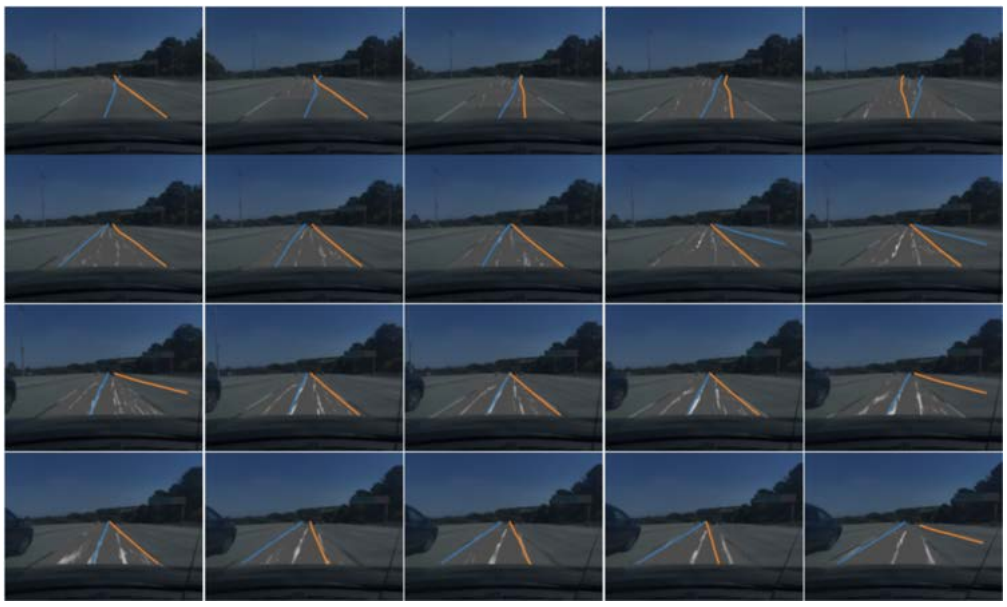


Figure 5.H.5: The first 20 frames (from left-top to right-bottom) of an attack scenario on **LaneATT**. The vehicle is deviating to right due to the attack.

Chapter 6

Leveraging Infrared Laser Reflections to Target Traffic Sign Perception

6.1 Introduction

Every vehicle, whether a connected, autonomous vehicle (CAV), semi-autonomous, or human-driven vehicle, must obey traffic signs. Disobeying signs can cause potential accidents and threaten human life. Recent studies [167, 373, 249, 208, 376, 164, 261] on traffic sign recognition systems show how physical, adversarial attacks can degrade recognition accuracy. Such attacks include projecting shadows [376], projecting visible colored patterns [249, 164, 261, 367], and adding stickers to traffic signs [167, 373, 208], to induce misclassification. However, these prior attacks have a clear limitation in stealthiness. Stickers, strong light projections, or shadows inconsistent with the environment are visible to humans, who can detect and mitigate them. For example, in semi-autonomous vehicles, such as Tesla's [86], drivers are required to stay alert and ready to take manual control of the vehicle if needed. As long as visual changes are inevitable, humans may notice them,

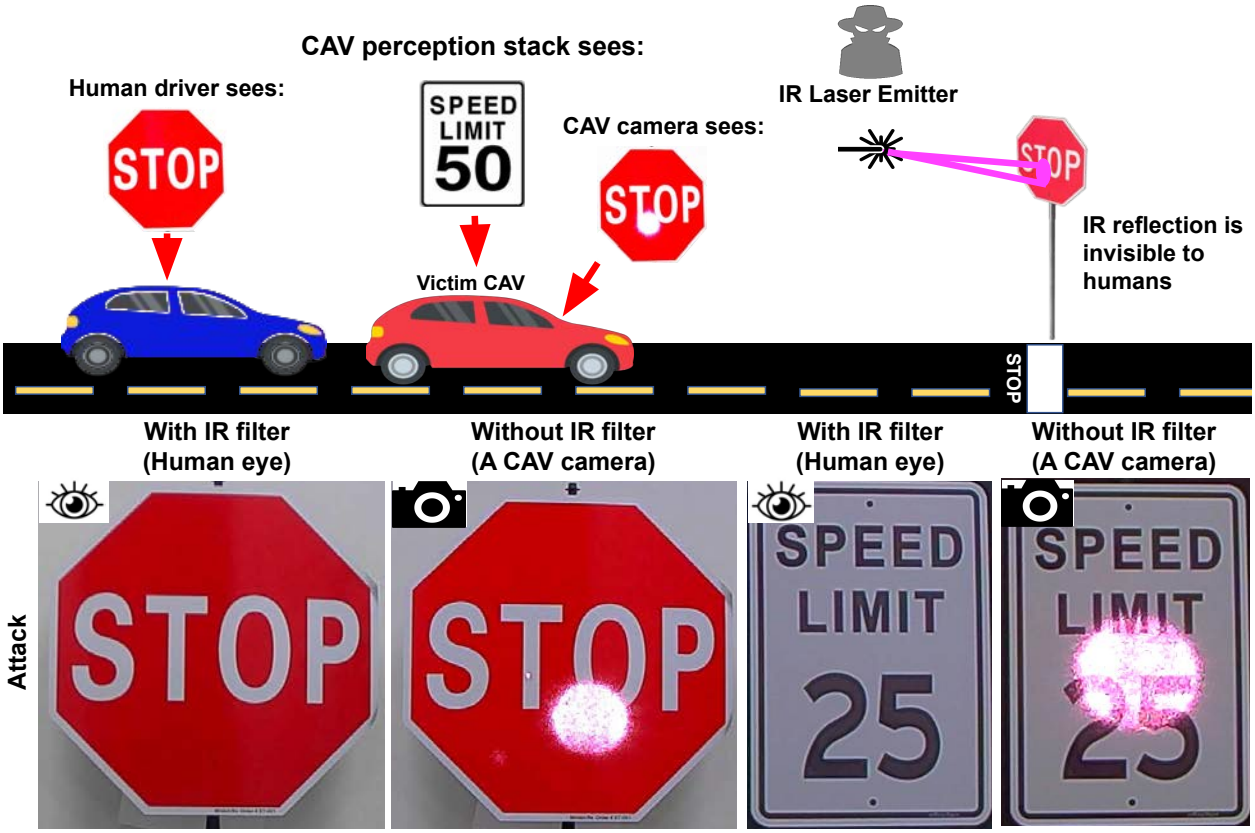


Figure 6.1.1: Overview of our ILR (Infrared Laser Reflection) attack. Unlike cameras without IR filters, humans can not see IR light. When an IR-sensitive camera images an object illuminated by an IR laser, the camera's output is altered at the pixel level. Our attack causes CAV perception stacks to misclassify traffic signs, causing dangerous misinterpretations.

even if they are subtle.

In this work, we design and demonstrate a novel, invisible attack that is able to mislead traffic sign recognition systems by leveraging patterns of infrared (IR) light reflections that are invisible to humans. As shown in Fig. 6.1.1, the reflections are visible to CAV cameras without an IR filter. These cameras capture the attacker's IR light reflections on the target traffic sign and output an altered image that is misinterpreted by the vehicle's perception module in its autonomy stack (e.g., detecting a speed limit sign instead of a stop sign).

Camera sensors are normally sensitive to photons in both visible and infrared wavelengths. Typically, commercial cameras use IR filters to ensure accurate color reproduction and pre-

vent unwanted contamination by infrared photons. However, some autonomous vehicles employ cameras without these filters to improve detection in dark environments [41, 258]. Our attack targets sensors lacking these filters. Moreover, although humans might perceive infrared light reflections through such cameras, CAVs generally do not display the captured images to the drivers, making this attack challenging to detect, or to distinguish from ordinary CAV malfunctions.

For example, the recent I-Can-See-the-Light (ICSL) attack [341] takes advantage of filterless sensors by projecting an IR pattern directly onto the camera image sensor in order to create fake objects and induce SLAM errors. Another work describes an invisible mask attack [380], which uses multiple IR emitters inside a hat to evade face recognition surveillance. However, these attacks only focus on changing traffic light colors and inducing detection errors, leaving unclear the impact of IR-based attacks on CAV traffic sign recognition systems. Furthermore, those attacks either require that the IR light source be aimed continuously and precisely at the target camera on a moving CAV, or only function at short distances. This limits the attacks' practicality in real-world scenarios. These limitations motivate our work.

Our *Infrared Laser Reflection* (ILR) attack causes CAV perception modules to misclassify, or in the worst case misdetect, traffic signs. We use an IR laser source to reflect IR projections off of a portion of a target sign surface. Leveraging the unique properties of laser light reflections, an IR-sensitive CAV camera will see the reflected light and incorporate it into the camera's output images. We call these images *traces*. The vehicle's perception module will then attempt to classify the trace-tainted images from the camera, resulting in misclassification. Unlike ICSL and similar attacks [354, 212], which require precisely aiming at the target sensor and accurately synchronizing timing, we only need to aim a single IR laser at a target traffic sign. The IR laser emitter is static and needs no sophisticated setup to track a moving CAV. We also find that the laser reflection properties can achieve stable misclassification at long distances with minimum power (up to 25 meters away from the target sign, with a laser

power of 26 mW). To maximize the attack effect, we developed a technique for generating optimized traces using the IR laser reflections, which allow the attacker to automatically find the optimal misclassification, minimizing the required laser power, and covering a minimal portion of the target sign with the reflection (7–17% of real-world stop and speed limit sign size in our outdoor evaluation).

We evaluate our ILR attack's effectiveness against two major traffic sign recognition architectures, using images captured with four different IR-sensitive cameras. Our paper is structured as follows: In §6.3, we formulate our threat model. We determine what parameters the attacker can control and what parameters are required to make the attack robust to different conditions. In §6.4, we describe our attack optimization methodology, which incorporates modeling the attack reflection characteristics to automatically find the optimal location of the projection on the target sign while minimizing the required power and reflection size. In §6.5, we demonstrate that our attack achieves up to a 100% attack success rate in the real world against both stop sign and speed limit sign targets under static, indoor laboratory conditions, and a $\geq$80.5% attack success rate in outdoor conditions, with a vehicle moving at increasing speeds. In §6.6, we evaluate ILR against the current state-of-the-art certifiable defense PatchCleanser [347], which has been evaluated against patch attacks that target generic image classification tasks for the ImageNet [157] and CIFAR-10 [227] datasets. We found that PatchCleanser's major intuition does not hold for traffic sign recognition systems, making the defense ineffective against ILR, as PatchCleanser mis-certifies $\geq$33.5% of attack and benign cases. To address this limitation, we propose a potential defense strategy based on the unique features of IR laser reflections. Finally, we discuss the limitations of this study in §6.7.

In summary, our study makes the following contributions:

- We identify ILR, a long-distance and human-invisible attack vector that can cause mis-

classification by traffic sign recognition systems. The ILR attack can not be seen by humans, but can be seen by cameras lacking IR filters. Our attack addresses the aiming and power limitations of previous works by combining invisible laser reflection properties and adversarial optimization.

- We design a novel methodology to optimize attack effectiveness by simulating IR laser projections, and their traces on signs, by modeling reflection size, intensity, and position using a black-box optimization.
- We evaluate the ILR attack against two different traffic sign recognition architectures using four IR-sensitive cameras. We confirm that the ILR attack reaches a 100% attack success rate under indoor laboratory conditions and a $\geq$80.5% attack success rate in outdoor environments with different light conditions and victim vehicle speeds ($\leq$13 km/h).
- We show that the major assumption of the state-of-the-art, certifiable defense, PatchCleanser [347], does not hold in the traffic sign recognition domain. PatchCleanser mis-certifies $\geq$33.5% of cases. We then propose a potential detection technique based on the unique physical characteristics of the IR laser reflections, which achieves a 96% True Positive Rate (TPR) and 6.7% False Positive Rate (FPR) in our proof-of-concept tests.

Details and demo videos are available on our website: https://sites.google.com/view/cav-sec/ilr-attack.

6.2 Background and Related Work

6.2.1 Vision-Based Traffic Sign Recognition

Vision-based traffic sign recognition systems use camera sensor outputs as inputs to fast neural networks, which perform real-time object recognition and classification [201, 166].

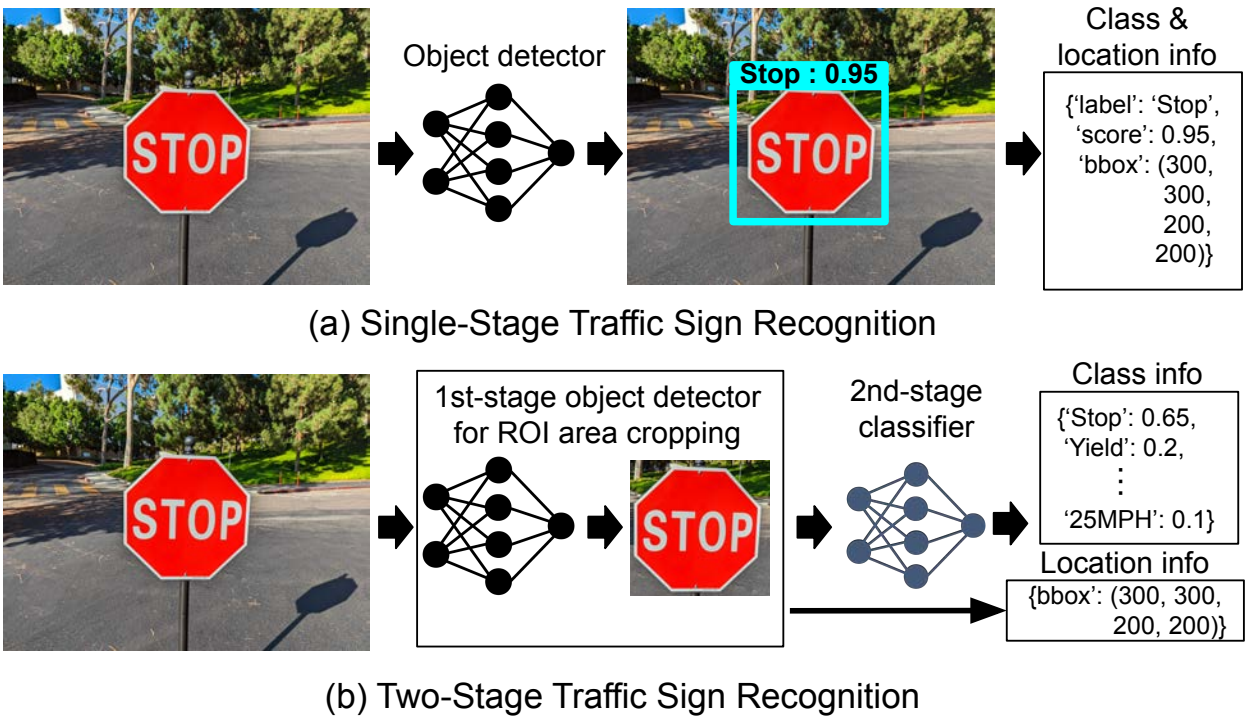


Figure 6.2.1: The two major traffic sign recognition system architectures used in our work. (a) A single-stage architecture detects and classifies traffic signs only with a single object detector; (b) A two-stage architecture's first-stage object detector finds and isolates (crops) the traffic sign in the image. The second-stage classifier provides the sign's class label.

These traffic sign recognition systems have benefits in terms of both capability and cost. They are also essential, as autonomous vehicles *must* recognize road signs in order to operate safely on public roads. This has driven their wide adoption in autonomous driving systems, such as those offered by OpenPilot [29] and Tesla [41]. Ertler et al. [166] identify two major vision-based traffic sign recognition architectures: *single-stage* and *two-stage*, as illustrated in Fig. 6.2.1:

Single-Stage Architectures. Single-stage architectures implement an object detector, such as YOLO [281], using a multi-class classification head to interpret traffic sign types. While the single-stage architecture is advantageous in terms of computational cost, Ertler et al. [166] report that the single-stage architecture does not yield acceptable performance when the number of classes is large, as with the 314 classes in [166]. Hence, typically, a single-stage

architecture is suitable for Level-2 autonomy systems that only need to recognize a limited number of signs.

Two-Stage Architectures. A two-stage architecture, which is capable of handling a large number of different signs, uses a first-stage object detector to crop the image to a "Region of Interest" (ROI) that contains the traffic sign. It then classifies the cropped image in its second stage [166]. Specifically, the first-stage object detector detects the sign's position, with cropping performed regardless of sign type. The second stage then classifies the cropped region with a sign type [147].

Unfortunately, previous work has shown how real-time sign perception can be vulnerable to attacks that affect what the camera sees [167, 141, 373]. Our work focuses on a novel perception attack that can affect both one and two-stage architectures. These are available in production vehicles.

6.2.2 Human Perception of Visible and IR Light

Invisible to humans, infrared light is electromagnetic radiation with wavelengths between 780 nm and 1 mm [308]. The CMOS image sensors used in today's cameras have sensitivity to some IR light. To match the perception characteristics of human eyes, they usually incorporate a filter that cuts out this IR light [108, 102]. However, to improve camera performance for nighttime driving, some production CAVs use cameras without IR filters [325, 41, 258]. Discussion of the prevalence of IR-sensitive cameras is challenging, as manufacturers seldom disclose specific information. To the best of our knowledge though, Tesla Model 3 cameras lack IR filters [341].

Our attack manipulates what the IR-sensitive camera sensor sees by projecting IR light patterns onto an object in the field of view of a CAV's camera. Since the unfiltered image

sensor in the camera can see IR light, the reflection of the projection becomes part of the sensor's output image. Humans cannot react to the attack since they can not perceive it.

As shown in Fig. 6.1.1, the reflection of the IR projection on the sign *is* visible in the output image. This occurs because the sensor can only measure the *intensity* of incident photons at each sensor photosite, not the *wavelength* of each photon. Typically, image sensors have color filters to only allow red, green, or blue photons to hit each photosite [114]. Because this filtering is imperfect, a portion of incident IR light passes through the color filters, reaches the photosites, and is integrated into the output image. Depending on the IR transmittance of these color filters, IR light appears in the output image with a false color, which is usually purple, magenta, or even orange.

6.2.3 Previous Work and Comparisons

Deep Neural Network (DNN) models today are shown to be generally vulnerable to adversarial examples (or adversarial attacks) [316, 178]. These attacks have previously been explored in the physical world [228, 298, 110, 122, 141, 167, 377, 373, 269, 328, 146, 378, 212]. In particular, traffic sign recognition has been shown to be vulnerable to adding small stickers to signs [167, 141, 373], visible pattern projection [249, 164, 240], and shadow shading [376]. Unlike physical patch attacks that leave permanent, detectable artifacts (such as small stickers) on the target sign [167, 373], our attacks avoid destructive changes or physical alterations to the target through the use of light projection and reflection. More specifically, our attack has the following three major differences or advantages over prior, related work:

(1) Invisibility and Attacker Capabilities. As discussed in §6.2.2, our attack is invisible to humans, rendering it more difficult to detect than physical patches [167, 373] or visible colored pattern projections [253, 262, 249, 240]. Previous work, such as ICSL [341], remote attacks [270], and invisible masks [380] use IR light to enhance stealthiness. In contrast to

this work, which uses non-coherent IR LED light, the ILR attack uses coherent laser light. Because coherent light waves are in phase with each other [315], laser light remains in a confined, tightly focused, and persistent beam over long distances, with little attenuation. This property can persist even when reflected from a surface, such as that of a sign, allowing us to optimize our projected patterns.

For example, we demonstrate how our ILR attack can achieve successful misclassification at different victim camera locations. We show this for both day and night conditions, with our laser up to 25 meters away from the target sign, and using only 26 mW of laser power, in §6.3.3 and §6.5. In contrast, LED light beams diverge in flight, attenuating over long distances. This makes them unsuitable for long-distance, confined pattern projection attacks unless a high powered beam is aimed directly at a vehicle's camera. As an example, ICSL [341], an IR LED-based attack, requires its LED light to operate at 30 Watts at 12 meters and must be aimed directly at the victim vehicle's camera (which is likely in motion) in order to successfully create fake objects.

(2) Continuous Tracking. A major challenge of light projection-based attacks on cameras, such as GhostImage [253], Rolling Colors [354], and similar laser spoofing attacks on LiDARs (PLA-LiDAR [212], PRA [125], Adv-LiDAR [130], and Illusion and Dazzle [198]), is the requirement to accurately and continuously aim at the victim sensor. This step is essential to ensure timing synchronization and accurate projection placement, which are needed to generate the correct adversarial pattern. Nevertheless, accurately tracking the sensor location on a CAV at any given moment proves challenging due to the dynamic nature of the vehicle's motion and external disturbances like vibrations, rendering the execution of such attacks difficult in practical scenarios.

To overcome this challenge, the use of reflection instead of direct injection has become a viable strategy in recent proof-of-concept attacks. For instance, AdvLB [164] and AdvSL [240] project visible light spots onto objects to fool object detectors and traffic sign recognition

models. However, such attacks use human-detectable visible light and have not shown effectiveness in real-world moving driving scenarios. In contrast, we realize that optimized invisible laser reflections are capable of inducing stable and prolonged misclassification in moving CAVs by targeting single-stage and two-stage traffic sign recognition architectures without the need for tracking and accurate projection. We demonstrate the effective application of our technique to moving vehicle scenarios in §6.5.4.

(3) Ambient Light Variations. Another challenge of light-based attacks is their effectiveness under different lighting conditions. Previous work that used projected lights [261, 249] and shadows [376] only succeeded under specific lighting conditions, such as at night. Success depends in part on the light source used. For example, non-coherent LED sources, as well as diffuse light from projectors, undergo scattering, causing it to spread out. When there is strong ambient light, the increased total luminance reduces the contrast between the projected pattern and its background. This can significantly reduce the attack success rate, rendering the attack impractical in bright environments. We show how ILR can succeed under diverse lighting conditions in §6.5.2.

Defense Strategies. Wang et al. [341] propose a defense against the ICSL attack. Their defense strategy distinguishes active street lights from IR light sources. Unlike active street lights, IR light sources do not reflect off of roadways. If the source lacks a reflection, it is not an active street light. They also use differences in reflection colors for real street lights versus the non-reflected sources used for attacks in order to distinguish between active and IR sources. Our attack can not be detected or mitigated using their solution since, unlike active street lights, street signs lack active illumination.

The ILR attack is a type of perception attack that changes how a small portion of a target traffic sign is perceived, as shown in Fig. 6.1.1. Thus, it can be considered an adversarial patch attack [122, 167, 141, 373]. Defenses against adversarial patch attacks should apply in theory. So far, there are two types of defenses against adversarial patch attacks: (i) empirical

defenses, such as the detection of anomalous patterns in attack patches [193, 182, 252, 373, 364, 376], and (ii) certified defenses with theoretical guarantees [359, 347, 346, 234]. Since the former is vulnerable to adaptive attacks [359], we focus on certified defenses, especially PatchCleanser [347], which is the current state-of-the-art. We evaluate our ILR attack against this defense and propose potential alternative defense strategies in §6.5.1.

6.3 Threat model and attacker capabilities

Fig. 6.1.1 shows an overview of the ILR attack. The attack goal is to cause the vision-based traffic sign recognition system of a victim CAV to misclassify a target traffic sign, such as a stop sign, as a different type of sign, such as a yield sign. We focus on untargeted attack scenarios. Sign misclassification can cause the CAV to behave dangerously, such as by braking unexpectedly or not stopping at an intersection. To do this, the adversary uses an IR laser to project an infrared light pattern onto a target sign with a specific size and position relative to that sign. While invisible to humans, the IR laser's reflected pattern is visible to the CAV's camera sensor. This results in the perception system's sign misclassification.

Prior Knowledge and Assumptions. We assume that the attacker can obtain specifications for the victim camera, such as the presence of the IR filter, by using public information such as datasheets and teardown reports [362, 254]. This is similar to the assumptions made by prior attacks on CAV cameras [354, 341]. Note that we assume the camera's internal settings, such as exposure time, are unknown to the attacker.

We also assume that the attacker has a basic understanding of IR light and optics in order to control the location of the projected IR pattern on the traffic sign surface as in previous work [130, 125]. The attack is remote and does not require any firmware access or information about the images captured by the camera in the victim CAV. However, we assume the

attacker has access to, or can purchase, a similar camera, and can empirically study and infer the properties of the camera as in previous work [354, 224]. We assume that the attacker has knowledge of the traffic sign recognition model used by the victim CAV and has black-box access to it as an oracle (for example, by reverse-engineering the vehicle communication [214]). Specifically, the attacker can learn the recognition results, including the confidence scores, of similar models but cannot directly access the victim CAV's model or its internal parameters (e.g., model weights).

Attack Scenarios. The attacker selects a target traffic sign and places an IR laser emitter in line of sight, up to 25 meters away, based on our evaluation setup, from the traffic sign along the roadside where the victim CAV is likely to pass by. The attacker can find a suitable location in advance, and the attack device can be secretly installed to reduce the possibility of being detected by others, as was done in prior work [249, 261]. Note that the suitable location of the emitter is adjustable by the attacker using appropriate lenses and supports.

6.3.1 Attack Modeling

We formulate the ILR attack as in Fig. 6.3.1, where the distance of the victim AV camera from the traffic sign, d_{vs}, and the distance of the victim AV camera from the attacker IR emitter d_{av} change dynamically as the victim CAV moves. We model the ILR attack with the parameters listed in Table 6.3.1.

The *attack parameters* represent the factors controlled by the attacker and include (1) the distance of the attacker's laser from the traffic sign, d_{as}; (2) the laser beam power (in mW), P_a; (3) the diameter of the projected IR pattern, D; and (4) the location of the center of the IR pattern in the traffic sign surface coordinates, (x_b, y_b). The attacker can optimize various combinations of parameters to maximize the attack's effectiveness. Throughout this work, we consider a circular IR pattern with a diameter D to evaluate our ILR attack. This is

Table 6.3.1: Definition of parameters

Attack Parameters	Parameter Description
d_{as}	Distance: attacker ↔ sign
D	Diameter of IR pattern
P_a	Laser power
(x_b, y_b)	IR Pattern center coordinates

Scenario Parameters	Parameter Description
d_{vs}	Distance: victim ↔ sign
d_{av}	Distance: attacker ↔ victim
L	Intensity of ambient light

because it is the easiest pattern to create for a non-sophisticated attacker by positioning a lens or an iris in front of the laser emitter. Thus, we define the *size* of the projected pattern in terms of the diameter value D. More elaborate shapes and patterns can be achieved with more specialized equipment.

Finally, *scenario parameters* represent environmental factors not controllable by the attacker, such as the ambient light intensity L, the longitudinal distance between the victim camera and the traffic sign d_{vs}, and the lateral distance between the victim camera and the attacker setup d_{av}, as shown in Fig. 6.3.1. Note that we define only the minimum set of required parameters in our attack formulation necessary for the attacker to pursue a successful attack, as demonstrated in §6.5.

In our analysis, we consider the CAV camera output to be a stream of still images, and we call the set of still image pixels altered by our ILR attack an *attack trace* (see Fig. 6.3.2). Every change in the attacker parameters in Table 6.3.1 independently affects the camera's output image, resulting in different attack traces. Using these assumptions, we describe an attack model optimization that accounts for *temporal image noise* (random fluctuations in victim camera's output) in terms of image pixel intensity (photon count) values [327] that occur as stray photons hit different sensor photosites across consecutive image frames. We thus evaluate the attack's effects on CAV sign classification over multiple consecutive camera image frames. A detailed methodology description is provided in §6.4.

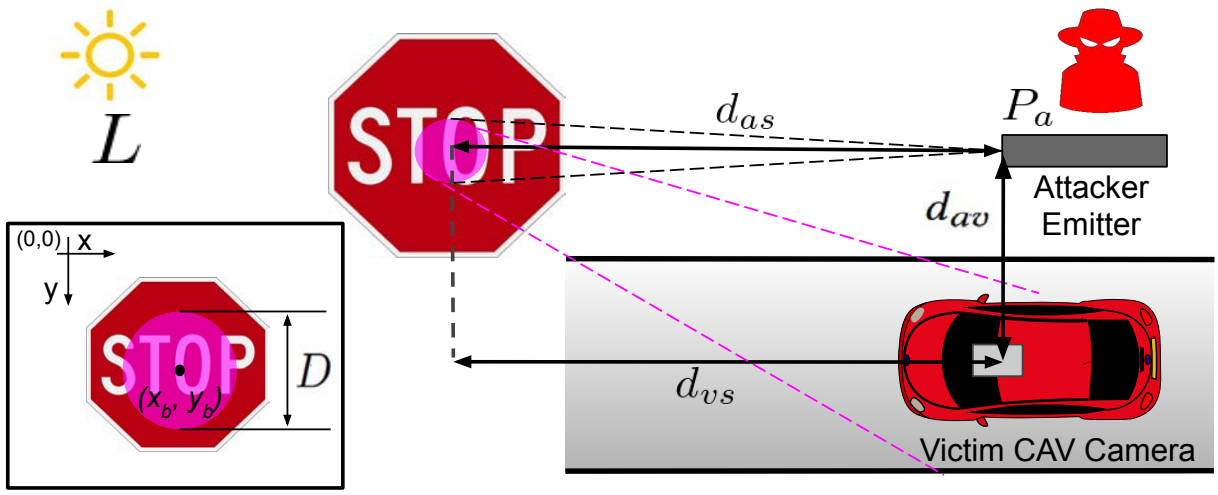


Figure 6.3.1: Overview of parameters of ILR attack

6.3.2 Physics of IR Laser Reflections

To define the attacker's capability to conduct the attack, it is necessary first to understand the impact of the reflected, projected IR pattern on the output images of the CAV camera. As described in §6.2.3, laser light is a coherent source where all the light waves are in phase. In contrast with diffuse light, laser light preserves some properties of the original beam when reflected. Thus, when a laser beam strikes an ideal reflective surface, the reflected beam will be similarly directional and focused, preserving all the properties of the original beam.

Traffic signs, such as the ones in Fig. 6.1.1, generally use high-quality corrosion-resistant aluminum alloy sheets to meet Manual on Uniform Traffic Control Devices (MUTCD) standards [158]. When a laser beam illuminates a rough surface that is not perfectly reflective, such as a traffic sign, the laser beam is scattered in all directions by surface irregularities. These scattered light waves interfere with each other and produce a *speckle pattern* visible in the CAV camera from different viewing angles. Since the scattered waves are still coherent, they preserve the same directionality and circular shape as the incident laser beam. A small portion of the light is diffused, adding noise to the captured image while also attenuating the light, as shown in Fig. 6.4.1. The ratio of coherent to incoherent light depends on surface

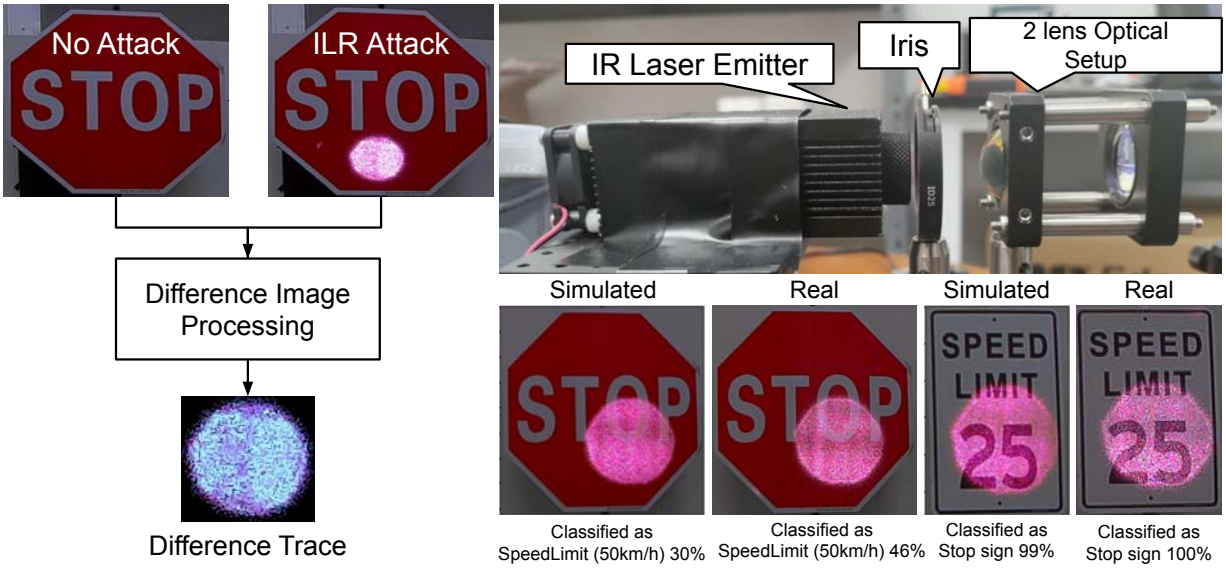


Figure 6.3.2: Overview of Image Difference-based IR Trace Modeling (Left). The IR Laser module has a two-lens optical setup (right-top). A comparison of the simulated IR pattern with our modeling and the corresponding real-world IR pattern.

roughness, laser light wavelength, and camera locations, all of which affect the visibility and complexity of the speckle pattern [320].

6.3.3 ILR Attack Capability Study

In this section, we investigate the ability of an attacker to perform an ILR attack. Our study aims to define (1) the relationship between the emitted IR power and the resulting range of pixel intensity variations in the captured images, (2) the correlation between the size of the projected IR pattern and image pixel intensity variation in the resulting speckle, and (3) the maximum achievable distance from which the attacker can make a successful attack. Note that the experiments in this section are conducted in a controlled, closed, indoor environment, except for the maximum distance experiment. We use real-world aluminum stop, and 25 mph speed limit, signs as targets. We also use a Leopard camera with an OnSemi AR032ZWDR image sensor as the victim camera [233] (referred to as OnSemi in the rest of the paper). OnSemi's camera is an automotive camera used by Baidu Apollo [107].

Attacker Setup. The attacker setup, used in all experiments in this work, consists of an IR laser emitter that projects an IR pattern with a controllable size onto the target traffic sign's surface. We use a 780 nm IR laser module from CivilLaser [148] with a maximum output power of 1 W. It projects a collimated beam with a 0.7 cm diameter and 0.75° divergence angle. The attacker controls the power P_a of the laser by changing the input current to the laser module. Laser modules generally consist of a stack of multiple edge-emitting laser diodes. These diodes have different parallel and perpendicular divergence angles, resulting in an elliptical beam [295]. Thus, we place an iris in front of the laser module to create a circular IR pattern from the original elliptical beam. Finally, the divergence angle of the projected laser beam is regulated by adjusting the distance between a two-lens optical setup as shown in Fig. 6.3.2. This design allows the attacker to control the projected circular pattern diameter D. Details on laser safety are described in §6.7 and Appendix 6.I.

Laser Power vs Pixel Intensity. In a controlled, indoor scenario, we set $d_{as} = 3$ m, the maximum distance achievable indoors. The room is illuminated by artificial ambient light, L, at 100 Lux. We then position the victim OnSemi camera such that d_{av} is 0.3 m and d_{vs} is 3 m. We set the circular IR pattern's diameter D to 15 cm and, starting from 0 mW, we increase the power up to 80 mW and measure the average difference in RGB pixel values created by the speckle as illustrated in Fig. 6.3.3. We observe that the minimum laser power required for the attacker to alter the image's pixel values and create a speckle is 2.4 mW – less than the power emitted by a laser pointer. This can be achieved by operating our laser module at 0.25% of its capability. We further notice that the 8-bit intensity variation created for blue pixels is larger than for red and green pixels (by at least 30) for laser powers greater than 20 mW. The red and green channel intensity variations follow a similar trend, as shown in Fig. 6.3.3 (top).

Similar to the laser power variation, the room's ambient light impacts pixel intensities by varying the victim camera's auto-exposure-controlled light sensitivity. Note that we treat the

victim camera as a black box. The bottom graph in Fig. 6.3.3 shows that the average 8-bit pixel intensity variation for the attack traces decreases with increasing room ambient light, I, at a logarithmic rate of $-40.1 \cdot \log(I)$ with a measured offset based on the transmitted laser beam power.

IR Pattern Size vs Pixel Intensity. The attacker can use the two-lens optical setup to control the size of the projected circular IR pattern. For a given distance d_{as}, we observe that the attacker can achieve a circular diameter D between 3.5 cm and 30 cm based on our hardware setup. The details about laser power attenuation with respect to beam size increase are provided in Appendix §6.A. We then characterize the pixel intensity variation for laser power increasing from 20 mW to 80 mW. We observe that the intensity offset decreases linearly at increasing sizes of IR patterns, as shown in Fig. 6.3.4.

Attenuation at Increased Lateral Distance. We verify the speckle intensity attenuation in the captured images at increasing lateral distance from the emitter (d_{av}). We place the emitter at 3 m (d_{as}) away from a stop sign and then measure the variation of pixel intensity values in the RGB channels of the captured attack trace images. We observe pixel intensity drops of up to 18% when the victim camera moves from 0 to 1.5 m (approximately the distance from the center of a roadway lane). Since this attenuation is negligible compared to the attenuation due to IR pattern size and emitter distance from the target sign, we only consider those factors in our attack optimization design described in §6.4.

6.4 ILR Optimized Attack Methodology

Based on the attacker capabilities described in §6.3.3, we design an optimization framework to automatically generate effective ILR attacks in terms of the optimal IR pattern location, the minimum circular pattern diameter, and the minimum laser power required by the attacker

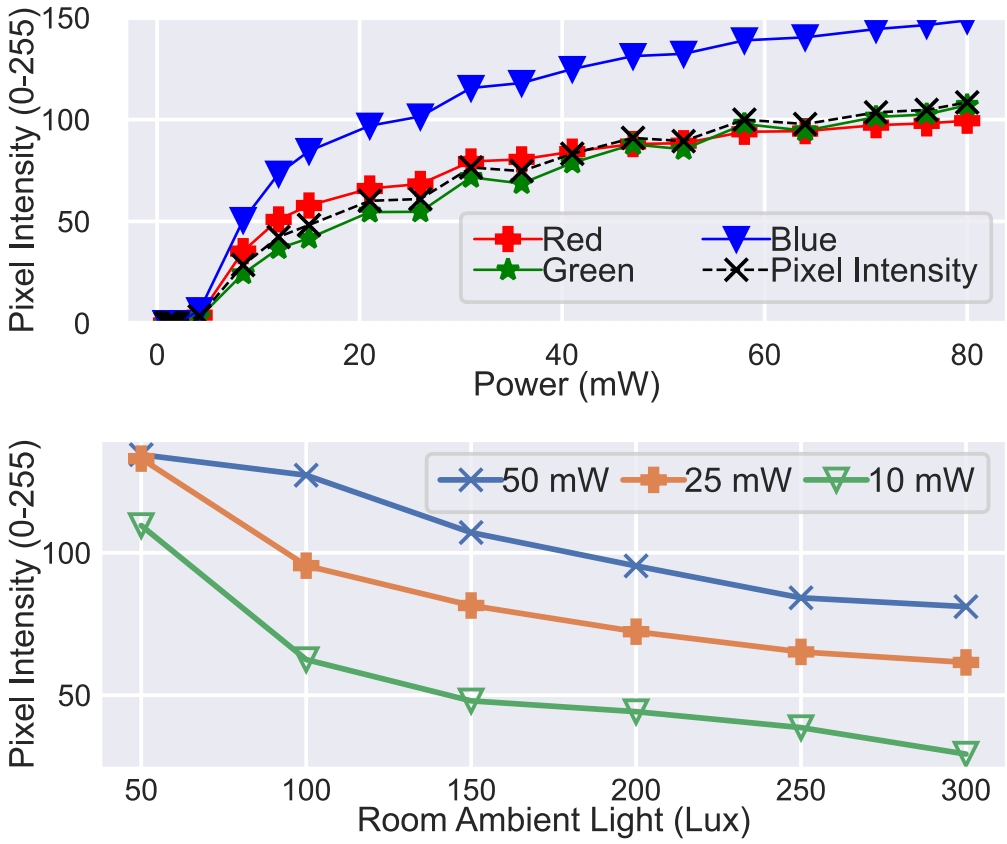


Figure 6.3.3: 8-bit RGB pixel intensity and overall pixel intensity variation of the attack traces for $D = 15$ cm IR pattern at increasing power (top). The impact of artificial ambient light on pixel intensity offset (bottom).

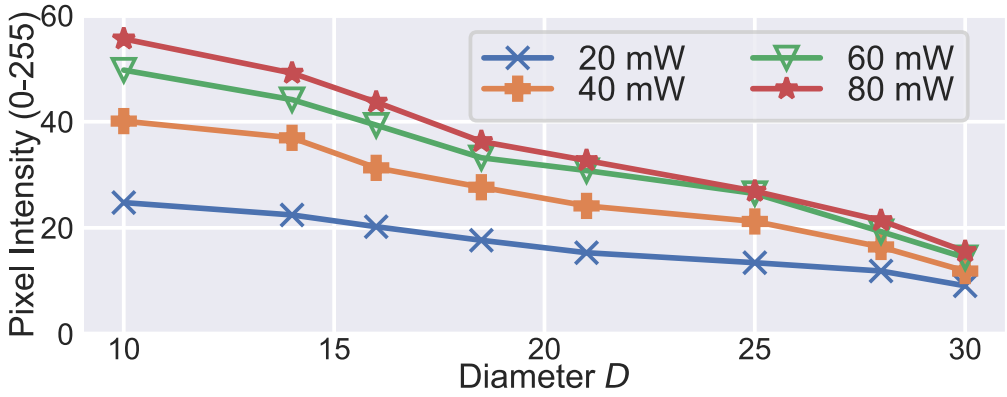


Figure 6.3.4: The offset in 8-bit pixel intensity values at increasing IR pattern size D at different laser powers P_a.

to achieve misclassification. Fig. 6.4.1 shows an ILR attack generation overview. To obtain optimized attacks, the framework performs: (1) image difference-based IR trace modeling

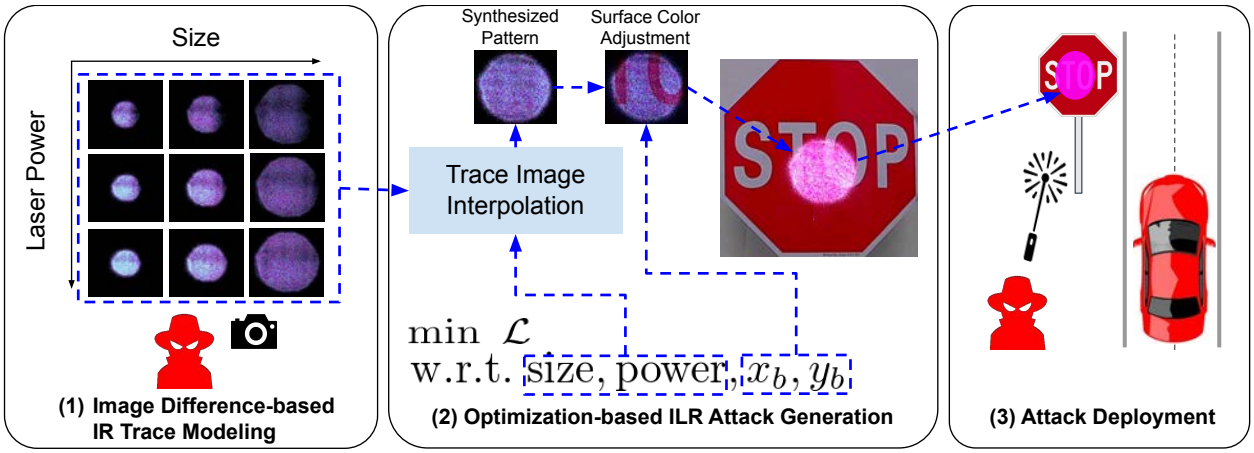


Figure 6.4.1: Overview of ILR attack generation. (1) The attacker first collects IR traces on the targeted traffic sign for different laser powers and sizes, and (2) optimizes the attack w.r.t size, power, and location of the projected IR pattern. We apply a color adjustment based on the base color of the traffic sign. (3) The attack is deployed and verified in real-world scenarios.

(§6.4.1 and §6.4.2), (2) optimization-based ILR attack generation (§6.4.3), and (3) attack deployment on attack scenarios (§6.5).

6.4.1 Image Difference-based IR Trace Modeling

To optimize ILR attack effectiveness, we first synthesize the attack traces captured by the victim camera. As described in §6.3, the IR pattern projection generates a speckle (attack trace) in the output images. Accurately synthesizing attack traces is challenging since they result from multiple, randomly phased, coherent waves. Thus, we model the phenomena by collecting and applying image differencing to the attack traces to extract RGB intensity offsets while varying attacker parameters, such as emitted power and circular IR pattern size, as shown in Fig. 6.3.2. More details are in Appendix 6.B.

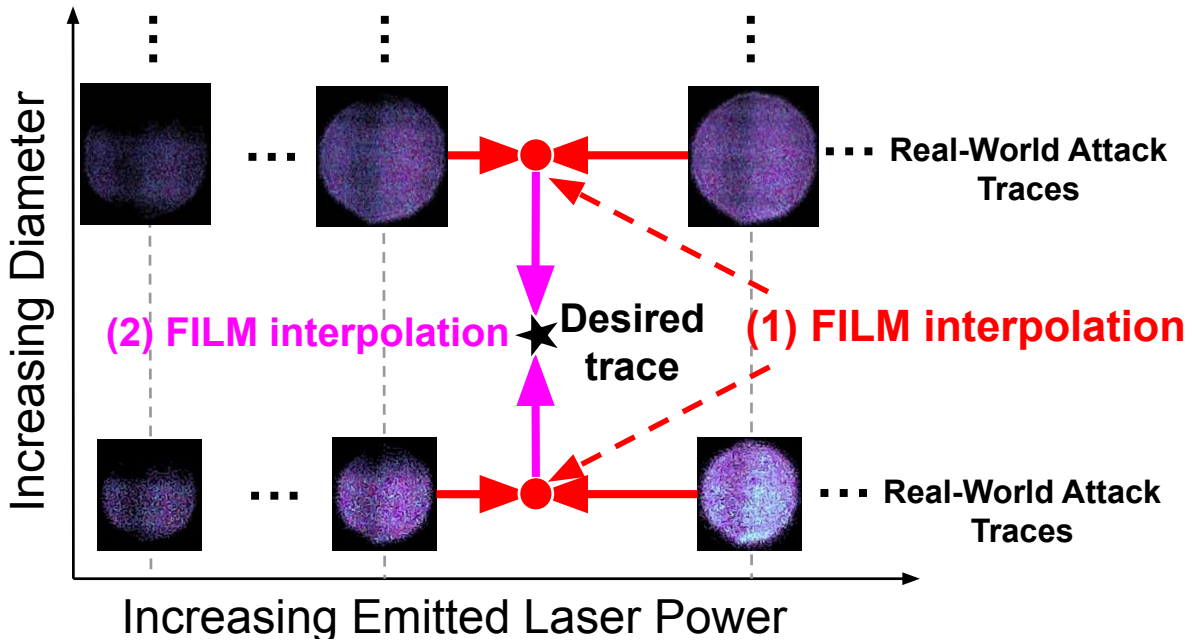


Figure 6.4.2: Overview of the DNN-based interpolation using FILM [280]. The method interpolates two traces at increasing emitted power and IR pattern diameters D.

6.4.2 Trace Image Interpolation

Collecting all possible real-world traces for all laser powers and IR pattern sizes is infeasible. Naive interpolation with averaging does not work for our attack, as averaging cancels out the speckle patterns. For this reason, we design a method to derive attack traces by interpolating real-world traces. To preserve the local spatial information while interpolating, we adopt a recent DNN-based frame interpolation algorithm, FILM [280]. FILM generates slow-motion videos from very similar photos. As shown in Fig. 6.4.2, we build the interpolation process for increasing laser powers and trace diameters and obtain intermediate attack traces as video frames. We note that this methodology can be applied to different traffic signs by adjusting the trace color or by collecting the appropriate real-world traces. We use the official pre-trained model in [280].

6.4.3 Optimization-based ILR Attack Generation

Finally, we design a black-box optimization formulation to optimize the image difference-based IR trace modeling (§6.4.1) and trace image interpolation (§6.4.2). This technique allows the simulation of an attack-influenced image with arbitrary trace position (x_b, y_b), laser power P_a, and trace diameter D, to find the optimal configuration. For other parameters listed in Table 6.3.1, we do not directly optimize the parameters, but consider them in the expectation over transformation (EoT) technique [110, 122] to be robust against changes to them. The attack formulation can be written as follows:

$$\begin{aligned} \min \quad & \mathbb{E}_{X \sim \mathrm{EoT}(X_{ILR})} \left[\mathcal{L}(X, \theta) \right] \\ \text{s.t.} \quad & X_{\mathrm{ILR}} = \mathrm{TraceModeling}(X_{\mathrm{base}}, T, x_b, y_b), \\ & T = \mathrm{Interp}(D, P_a) \end{aligned} \tag{6.1}$$

where the diameter of trace D, laser power P_a, and the position of attack trace (x_b, y_b) are the decision variables and θ is the targeted DNN model's parameter set. $\mathrm{Interp}(\cdot)$ is a function of the trace image interpolation. $\mathrm{TraceModeling}(\cdot)$ is a function of the image difference-based IR trace modeling used to get a simulated attack-influenced image, X_{ILR}. X_{base} is a benign (base) image containing the target traffic sign. In $\mathrm{EoT}(X_{\mathrm{ILR}})$, we sample images with the EoT technique from the simulated image X_{ILR}. In the EoT, we add Gaussian noises, change color brightness, and apply rotation and shear. $\mathcal{L}$ is a loss function. In this study, we simply minimize the confidence value of the target class — our attack is an untargeted attack as discussed in §6.3. As the attack formulation Eq. (6.1) is not differentiable, we use a black-box optimization method to find effective D, P_a, and (x_b, y_b). We adopt the Tree-structured Parzen Estimator algorithm [117] in Optuna [103]. We note that the optimization generally converges to the global minimum since the search space is small (4 variables).

6.5 Evaluation

We evaluate the ILR attack with respect to effectiveness, generality, robustness, and transferability in the real world. We also evaluate the effectiveness of ILR attacks in outdoor moving victim scenarios.

6.5.1 Attack Effectiveness and Generality Evaluation

In this section, we evaluate our attack on real-world aluminum, stop, and 25 mph speed limit signs in a controlled, closed, indoor environment with the setup described in §6.3.3, as shown in Fig. 6.1.1. We place the victim camera and IR emitter at $d_{vs} = d_{as} = 3$ m in front of the target sign. For collecting the traces for our optimization model, we increase the laser power P_a from 2.4 to 80 mW and the diameter D from 10 to 30 cm, based on our assumed attacker capabilities. The artificial ambient light, L is set to 100 Lux. We use the OnSemi camera as a default for this evaluation if not mentioned otherwise. Table 6.5.2 lists all the four cameras tested in the generality study. Note that we evaluate the physically realized ILR attack in the real world after applying our optimization methodology, not the digitally simulated IR patterns. We then compare our results against a baseline random attack in which an IR laser beam hits random portions of the target signs. Details of the random attack are in Appendix 6.E.

Table 6.5.1: Benign Performance of the object detectors and classifiers for traffic sign recognition. The architecture of the CNN model is in Appendix 6.D. YOLOv3 is evaluated in APb. Others are in mAP.

Object Detector (Training Dataset)	mAP/APb	Classifier (Training Dataset)	Acc.
Faster-RCNN [283] (ARTS [104])	84.3	CNN (ARTS [104])	81%
Faster-RCNN [283] (Mapillary [166])	18.3	CNN (LISA [259])	99%
YOLOv3 [282] (COCO [243])	33.8	CNN (GTSRB [201])	98%

Table 6.5.2: Target cameras considered in our evaluation.

	Leopard OnSemi	**Raspberry Pi HQ v1.1**	**LifeCam HD-3000**	**Leopard OmniVision**
				
Sensor	AR032ZWDR	IMX477	N/A	OV10635
Usage	CAV	General	WebCam	CAV
Resolution	1920×1080	4056×3040	1280×720	1280×800
Lens f	6 mm	16 mm	N/A	6 mm
Max FPS	30	90	30	30
FOV	H - 60°	44.6° × 33.6°	D - 68.5°	D - 68.5°

Targeted Traffic Sign Recognition Models

Table 6.5.1 lists the targeted object detectors and classifiers and their benign performances. For the *single-stage architectures*, we train an object detection model with the ARTS [104] and Mapillary [166] datasets. As the Mapillary dataset includes worldwide traffic signs, we only use the stop and speed limit signs used in the United States. For the stop sign only, we evaluate YOLOv3 [282] trained with the COCO dataset [243], which is a generic object detector but does not contain US-style speed limit signs. Thus, we evaluate different datasets for stop and speed limit signs. We use 0.3 as the object-detection threshold of confidence score, following conventional practice [139].

For the *two-stage architectures*, we manually crop the ROI area for each sign to focus on the analysis of the second-stage classification. For the second-stage classifiers, we train classification models with three datasets, one trained on European traffic signs and the other on U.S. signs. For the European traffic signs, we trained a CNN classification model on the GTSRB [201] dataset. For the U.S. traffic signs, we trained CNN models on the LISA [259] and ARTS [104] datasets. We use a CNN model architecture that is among the best performers on the GTSRB dataset [299]. Details are in Appendix 6.D.

Evaluation Metrics

We design two evaluation metrics based on our threat model (§6.3) and attack design: the attack success rate (ASR) and the simulation consistency rate (SCR). ASR measures the percentage of cases in which a sign is misclassified or undetected, thus satisfying our attack goal, as discussed in §6.4.2. SCR is defined as the percentage of cases in which the classification caused by the ILR attack is consistent between physical and digital scenarios. We use SCR to evaluate the quality of attack trace modeling (§6.4.3). We do not consider the SCR for the single-stage architecture since a successful attack typically just prevents the object detector from detecting the traffic signs. ASR is our primary metric. SCR is necessary to evaluate the validity of our attack design.

Results

Table 6.5.3 shows the effectiveness of the ILR attack against single-stage and two-stage traffic sign recognition systems. Our ILR attacks show significantly higher effectiveness than the random attack, with a 100% success rate for all models. These results indicate that while the IR laser traces fool traffic sign recognition systems, effective attack optimization is needed to cause a significant impact on recognition. Table 6.5.4 shows the ASR and SCR of the ILR attack compared with "w/o interp.", which only explores the discrete power and size values of the collected IR patterns without interpolation, and "Spline Interp.", which is a baseline method that adapts a cubic spline interpolation. Details on this method are described in Appendix 6.C. The DNN-based interpolation method has the highest ASR and SCR with 100% ASR and 92.5% SCR on average. This reveals that high-quality interpolation is essential to find effective ILR attacks.

Generality to Different Cameras. To further study the impact of different cameras, we evaluate the attack's effectiveness using a Raspberry Pi HQ v1.1 camera [279] with a

Sony IMX477 image sensor [309], a Microsoft LifeCam HD-3000 camera [256], and another automotive camera with an OmniVision OV10635 image sensor [232, 266] (referred to as OmniVision) with the IR filter removed. As shown in Fig. 6.5.1, the perceived IR pattern varies between different cameras, as they vary in their sensitivities to infrared wavelengths. Table 6.5.5 lists the ASR and SCR of the ILR attack on the four tested cameras. The ILR attack is always successful, with an ASR of 100%. On the other hand, the speed limit sign SCR was 0% for the Raspberry Pi HQ v1.1 and LifeCam cameras. We observed that IR trace color is camera sensor dependent. Thus, the relationship between trace color, diameter, D, and power, P_a, is also sensor dependent. This affects the accuracy of simulated IR traces.

Maximum Achievable Distance. To evaluate the maximum achievable distance from the emitter to the target traffic sign, the experiment was conducted in both indoor and outdoor scenarios in a controlled environment. Using our setup, we verify the ASR against the speed limit sign using the LISA model [259] configured as described in §6.5.1. Our results show that with our minimal setup, ILR consistently succeeds (100% ASR) up to 25 meters away from the target sign with a power of 26 mW. Long-range attacks are possible because of the laser beam properties described in §6.3.2. Beyond 25 m, the speckle intensity loss and beam divergence prevent coherent pattern shape projection, dropping the ASR to zero. More sophisticated optics can be used to increase the attack range.

Attack with Saturated IR Speckle. The optimization methodology focuses on minimizing the laser power necessary for the attacker to achieve a successful attack. We can further evaluate ILR by considering an attacker whose goal is to achieve camera sensor saturation using the projected IR laser speckle. We consider an IR speckle to be saturated if $> 50\%$ of the pixels in the pattern are saturated, i.e., intensity = 255. For this analysis, maintaining the same attack scenario as the maximum achievable distance evaluation, we optimize the trace only with respect to the trace diameter D and the coordinate position of the center of the trace (x_b, y_b). We increase the IR beam power to achieve saturation when the trace

Table 6.5.3: Attack effectiveness of single-stage and two-stage traffic sign recognition systems with ASR and SCR.

			Random Attack		ILR Attack	
			ASR	SCR	ASR	SCR
Stop Sign	Single-Stage	Faster R-CNN (ARTS)	0%	N/A	**100%**	N/A
		YOLOv3 (COCO)	0%	N/A	**100%**	N/A
	Two-Stage	CNN (ARTS)	0%	N/A	**100%**	**100%**
		CNN (GTSRB)	20%	N/A	**100%**	70%
Speed Limit	Single-Stage	Faster R-CNN (ARTS)	60%	N/A	**100%**	N/A
		Faster R-CNN (Mapillary)	**100%**	N/A	**100%**	N/A
	Two-Stage	CNN (ARTS)	64%	N/A	**100%**	**100%**
		CNN (LISA)	10%	N/A	**100%**	**100%**

diameter is optimized. Our results for the OnSemi camera exhibit a 100% ASR in all evaluated two-stage models. We observe that the optimized trace diameter for stop sign attacks drops from 21.25 cm to 17.5 cm on average and from 31.5 cm to 27 cm for speed limit sign attacks under saturation conditions. This evaluation reveals how attackers can achieve high attack success regardless of the model tested at the cost of increasing laser power.

Generality to Different Laser Wavelengths To evaluate the attack generality against different laser wavelengths, we conducted experiments with 830-nm and 980-nm laser modules (in addition to our 780 nm laser). For each of the laser modules, we collect the IR traces and optimize for the attack trace individually. We find that the ILR attack can achieve a 100% ASR on stop and speed limit signs with both tested laser modules. 37 and 17 mW laser powers were required for the 830 and 980 nm laser modules respectively to attack the stop sign. Similarly, 44 and 26 mW laser powers respectively were required in the case of speed limit signs. We observe that for higher frequency modules, a lower laser power is required to attack a stop sign. We hypothesize that this is because the IR traces created by high-frequency lasers tend to appear with a more blue-shifted (higher contrast) hue in the camera image, when compared with the stop sign's red surface color.

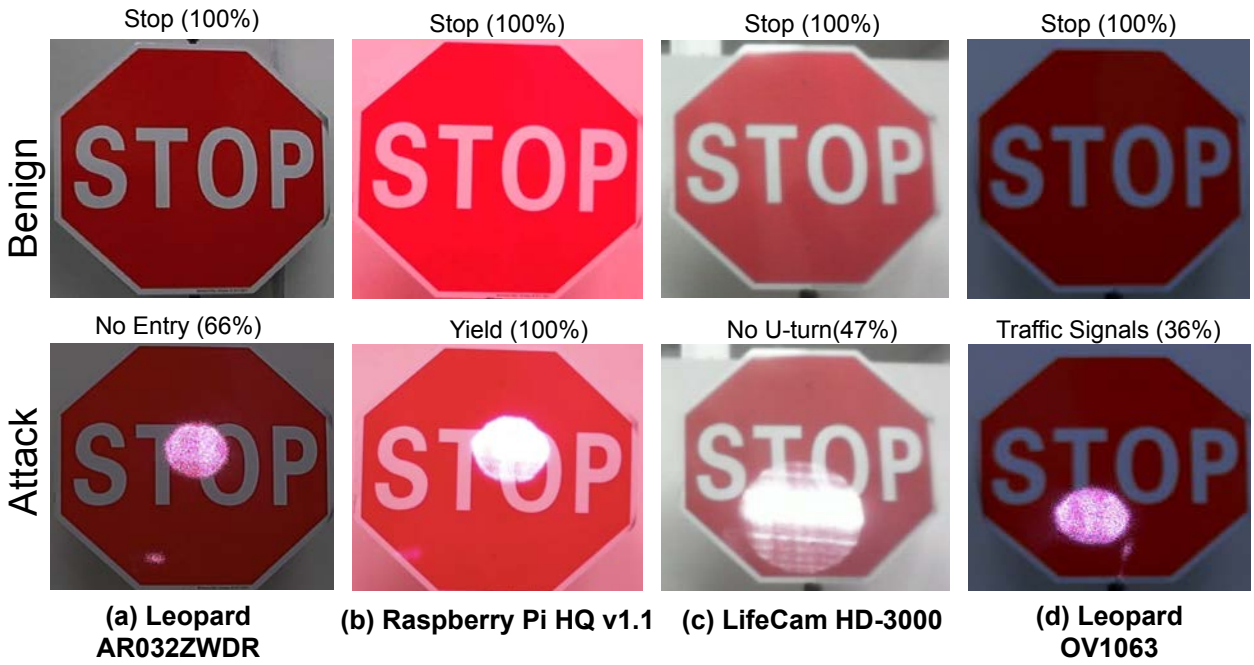


Figure 6.5.1: Examples of captured images of the 4 IR-sensitive cameras in benign and during a successful attack. The detected colors differ as each camera has a different sensitivity to IR.

Table 6.5.4: Evaluation of the interpolation method. "w/o interp." only optimizes the attack with discrete laser powers and sizes without interpolation. "Spline Interp." is a baseline spline-based method detailed in Appendix 6.C.

		DNN-based Interp.		w/o Interp.		Spline Interp.	
		ASR	SCR	ASR	SCR	ASR	SCR
Stop Sign	CNN (ARTS)	**100%**	**100%**	20%	20%	**100%**	**100%**
	CNN (GTSRB)	**100%**	70%	90%	90%	80%	80%
Speed Limit	CNN (ARTS)	**100%**	**100%**	**100%**	**100%**	**100%**	0%
	CNN (LISA)	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**

6.5.2 Attack Robustness Evaluation

We evaluate the robustness of the ILR attack under varied ambient lighting conditions and camera positions using the same indoor setting as detailed in §6.5.1.

Robustness to Different Ambient Lightings. Fig. 6.5.2 shows the ASR of the ILR

Table 6.5.5: Attack effectiveness on the 4 different cameras.

		OnSemi		Raspberry Pi HQ		LifeCam HD-3000		OmniVision	
		ASR	SCR	ASR	SCR	ASR	SCR	ASR	SCR
Stop Sign	CNN (ARTS)	100%	100%	100%	100%	100%	100%	100%	20%
	CNN (GTSRB)	100%	70%	100%	100%	100%	100%	90%	90%
Speed Limit	CNN (ARTS)	100%	100%	100%	0%	100%	100%	100%	100%
	CNN (LISA)	100%	100%	100%	0%	100%	0%	100%	100%

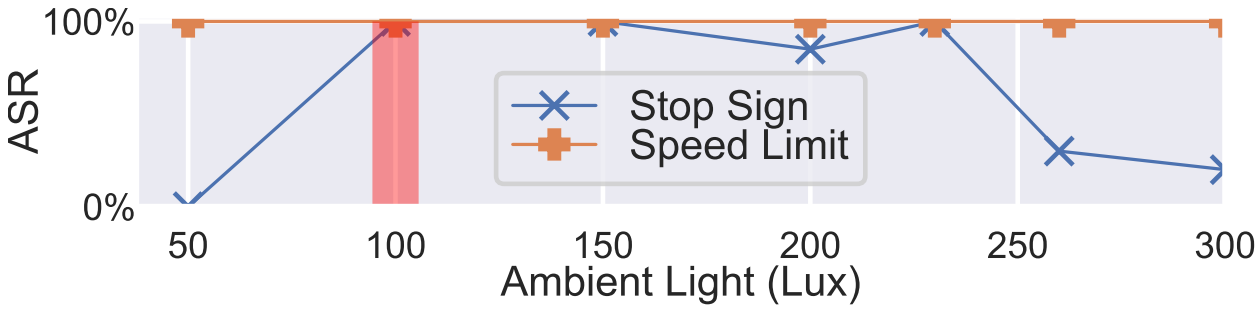


Figure 6.5.2: Attack success rates under different ambient lights. The attack is generated under 100 Lux at the red bar.

attack against second-stage classifiers with increasing ambient light levels. The attack is generated at 100 Lux and evaluated for 7 other artificial light levels, ranging from 50 to 300 Lux. For the stop and speed limit signs, we use the CNN classifier models trained with the GTSRB and LISA datasets, respectively. As shown, the ILR attack against the stop sign shows high robustness between 100 and 230 Lux, but its ASR drops significantly above 230. In contrast, the attack against the speed limit sign shows high robustness, with 100% ASR for all light settings. We believe the difference in performance is due to differences in contrast between the traffic sign surface colors and the laser speckle. On a white sign, the speckle has a higher contrast than on a red sign. The speckle color is dependent upon the laser wavelength and camera sensor used, as described in §6.3.2.

Robustness to Different Object Detectors in Single-Stage Architecture. Table 6.5.6 lists the ASR for single-stage architecture object detectors at increasing distances d_{vs} between the camera and the traffic sign. The attack is generated for all the models at a

Table 6.5.6: ASR for single-stage architecture under 4 different distances d_{vs} between the camera and the sign.

Target Sign	Detection Model	4 m	5 m	6 m	7 m
Stop Sign	Faster R-CNN (ARTS)	**100%**	**100%**	**100%**	**100%**
	YOLOv3 (COCO)	0%	0%	**100%**	0%
	YOLOv5 (COCO)	10%	90%	**100%**	**100%**
Speed Limit	Faster R-CNN (ARTS)	**100%**	**100%**	**100%**	**100%**
	Faster R-CNN (Mapillary)	**100%**	**100%**	**100%**	**100%**
	YOLOv5 (ARTS)	**100%**	**100%**	**100%**	**100%**

fixed distance ($d_{vs} = 6$ m) and evaluated for the others. As shown, the ILR attack reaches high attack effectiveness for the speed limit sign with 100% ASR at all tested distances and models. For the stop sign, the ILR attack is effective against Faster R-CNN trained on the ARTS dataset but not always effective against YOLOv3 and YOLOv5. We believe these variations are due to the architectural differences in object detectors. The Faster R-CNN model, a two-shot object detector, finds region proposals and classifies those regions. It thus has a high ASR similar to the second-stage classification model as discussed in §6.5.1. YOLOv3 and YOLOv5, single-shot object detectors, perform the two steps simultaneously. This strategy may improve robustness as it can take into account global features from the region proposal. These results indicate that single-stage traffic sign recognition with a single-shot object detector can provide effective mitigation against ILR attacks. However, we note that the current object detectors are still not able to handle several types of different traffic signs, as discussed in §6.2.1.

Robustness to Different Camera Positions. Fig. 6.5.3 and 6.5.4 show the ASR and SCR of the ILR attack at increasing longitudinal (d_{vs}) and lateral (d_{av}) distances of the victim camera. The attack is optimized with the traces collected at $d_{vs} = 2$ m and $d_{av} = 1$ m and evaluated with real-world experiments at all other victim camera positions. As shown, the lateral direction has a higher impact on attack success than the longitudinal direction. We believe that the ROI cropping and resizing before applying the CNN model inference can

Table 6.5.7: Attack robustness to different angles between the laser emitter and the targeted traffic sign.

		Left-20°		Left-10°		Right-10°		Right-20°	
		ASR	SCR	ASR	SCR	ASR	SCR	ASR	SCR
Stop	GTSRB	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
Sign	ARTS	50%	50%	**100%**	**100%**	**100%**	**100%**	70%	70%
Speed	LISA	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	0%
Limit	ARTS	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**

cancel the effect of the longitudinal differences, while lateral differences change the viewing angle of the traffic signs, significantly altering the speckles in the resulting images. As the attack is not optimized for the viewing angle, attack performance is degraded despite applying EoT techniques (See §6.4.3). Nevertheless, the ASRs remain high, particularly within 1 m lateral translations. Since a road lane is approximately 3.0-3.6 meters wide in real-world scenarios, degradation from different camera viewing angles does not have a major impact on attack performance. For SCR, the stop sign typically has higher values than the speed limit sign, while the speed limit sign has higher ASRs.

Robustness to Different Laser Projection Angles. Table 6.5.7 lists the ASR and SCR for the ILR attack against the second-stage classifiers at different angles between the laser emitter and the targeted traffic sign. The attack is generated with the laser emitter in front of the target traffic sign and evaluated for four different laser projection angles, spanning a total of 40° (±20° relative to the plane of the traffic sign). As shown, the attack on the stop sign in the GTSRB model has high robustness while for the ARTS model, the ASR drops at 20° in both directions. The attack on the speed limit shows a 100% ASR for all projection angles for both models. The slight performance degradation for the stop sign in ARTS is consistent with the IR speckle pattern variations observed in the camera position and ambient light experiments.

Robustness to Inaccuracy in First-Stage Object Detection. We evaluate robustness

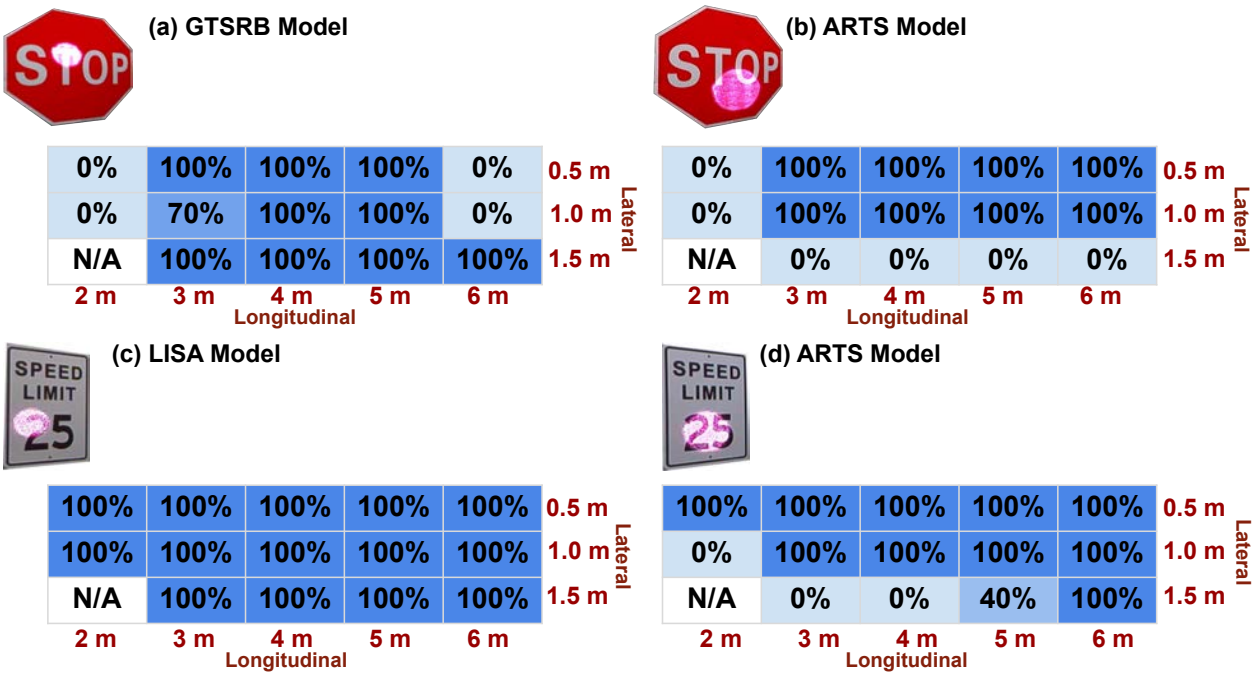


Figure 6.5.3: ASR for two-stage architecture model with 14 different camera positions. N/A: the traffic sign is not visible.

against the inaccuracies in the first-stage architecture, which can affect the ROI cropping, and consequently alter the input area to the second-stage classification model. The ILR attack shows high robustness against displacement errors of $\leq$8%. More detailed results are shown in Appendix 6.F.

6.5.3 Attack Transferability Evaluation

In this section, we evaluate ILR attack generality for different classifiers in the two-stage approach, and for different object detectors in the single- and two-stage approaches. We follow the same experimental setup as in §6.5.1.

Transferability to Different Model Architectures. Table 6.5.8 lists the ASR of transferred attacks generated with the CNN models and applied to 3 types of models: DenseNet121 [203], EfficientNet B0 [319], and ResNet50 [194]. The speed limit sign has significantly higher at-

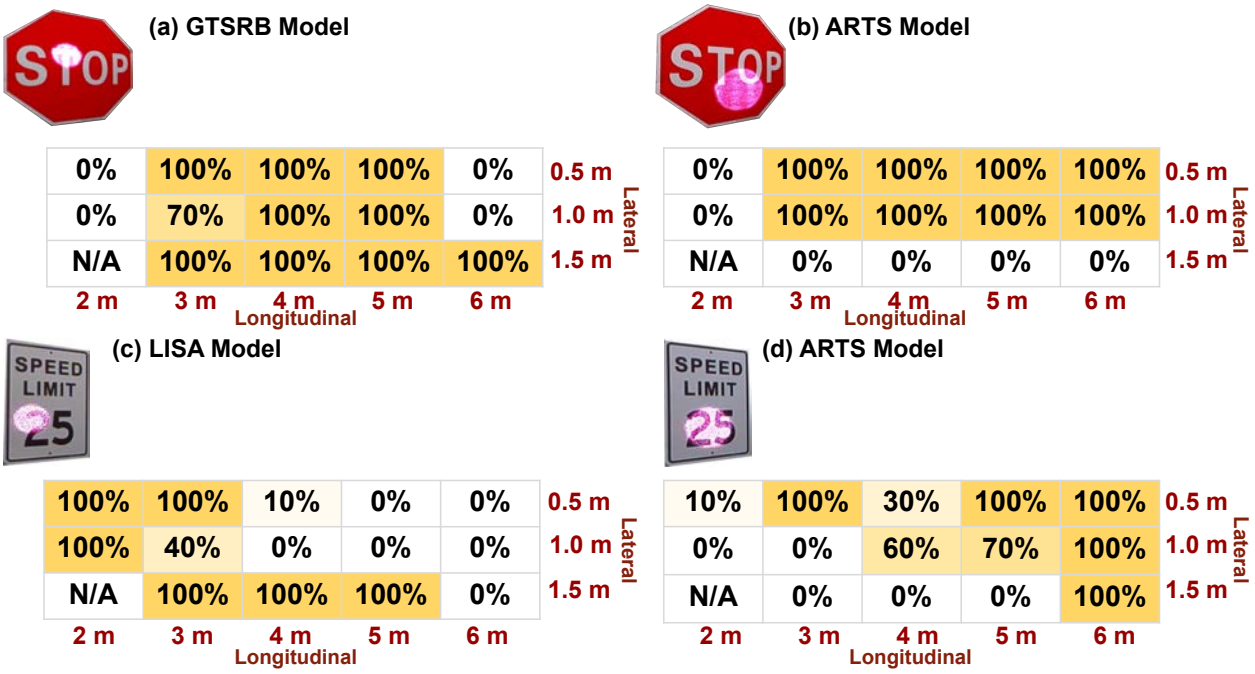


Figure 6.5.4: SCR for two-stage architecture model with 14 different camera positions. N/A: the traffic sign is out of FOV.

tack transferability to other models, as the ASR is always 100%. Meanwhile, the stop sign has a low transferability due to the low contrast between the speckle color and the stop sign surface color (red). However, the transferability on EfficientNet has an ASR of 100%. We hypothesize that the ILR attack causes perturbations in features to which the model has greater sensitivity than it does for adversarial patch attacks. [167, 141, 373]. Our results show that the ILR attack can be transferred from one model to another if the two models rely on the same robustness features to determine their predictions, as with the CNN and EfficientNet, but may fail if the features differ, as exemplified by DenseNet121 and ResNet50.

Transferability to Different Training Datasets. We evaluate ILR attack transferability across three datasets (ARTS GTSRB and LISA) using the same CNN model (CNN model details are listed in Appendix 6.D). Our evaluation shows high transferability (100% ASR) for all the datasets. We believe this is due to the large, manipulated area of traffic signs resulting from the speckle. Compared to the results in §6.5.3, the model architecture has

Table 6.5.8: ASR of the transfer attacks between different model architectures. The attacks are generated by the source model and evaluated in the transferred model.

	Source Model		Transferred Model		
	Dataset	CNN	DenseNet121 [203]	EfficientNet B0 [319]	ResNet50 [194]
Stop Sign	ARTS	**100%**	0%	0%	0%
	GTSRB	**100%**	0%	**100%**	0%
Speed Limit	ARTS	**100%**	**100%**	**100%**	**100%**
	LISA	**100%**	**100%**	**100%**	**100%**

Table 6.5.9: Transfered ILR attack success rates (ASRs) for different object detectors.

		Target model				
		Stop			Speed	
		Faster R-CNN (ARTS)	YOLOv3 (COCO)		Faster R-CNN (ARTS)	Faster R-CNN (Mapillary)
Source Model	Faster R-CNN (ARTS)	**100%**	20%	Faster R-CNN (ARTS)	**100%**	**100%**
	YOLOv3 (COCO)	**100%**	**100%**	Faster R-CNN (Mapillary)	**100%**	**100%**

a more significant impact on attack transferability than does the training data set since it influences the features used for the classification.

Transferability between Different Object Detectors. Table 6.5.9 lists the ASRs for ILR attacks transferred between different object detectors. As in §6.5.3 and §6.5.3, the ILR attack shows higher transferability even against object detectors. However, the YOLOv3 model trained on the COCO dataset appears more robust with only a 20% ASR. The model trained on the COCO dataset is used for generic object detection rather than specific for traffic sign recognition. Thus, it only has a single class for traffic signs (the stop sign). For this reason, it becomes harder to alter the legitimate prediction of the stop sign with small IR traces compared to a model trained on several different traffic signs. This result indicates that the first-stage object detector in the two-stage approach can be robust against ILR attacks while the second-stage classifier is still vulnerable. We will discuss it in §6.7.

6.5.4 Outdoor Evaluation

To study the effectiveness of the ILR attack in realistic scenarios, we evaluate the attack against the second-stage classification models in a controlled outdoor scenario with the setup

described in §6.5.1 under different ambient light conditions (e.g., day and night). Fig. 6.5.5 shows an overview of the evaluation scenario and victim camera view during day and night natural light conditions. We evaluate the two automotive camera sensors: OnSemi and OmniVision. The results of the OmniVision are detailed in Appendix §6.H.

Static Scenarios

We follow the same experimental setup as in §6.5.1 and perform 10 trials for each experiment.

Nighttime Attack. We collect attack traces for optimization with the victim camera placed at a 5 m longitudinal distance and a 1 m lateral distance, and then perform the optimized attack in the real world. Similarly to §6.5.1, we set d_{as} to 3 m. We measure an average ambient light of 120 lux. For GTSRB stop sign recognition, we achieve 100% ASR and SCR using 45 mW of power and an average trace diameter of 23 cm, equivalent to 7% of the entire traffic sign surface. For LISA, as used for speed limit signs, we achieve 100% ASR and 20% SCR using 46 mW of power, covering 17% of the traffic sign. Finally, we achieve a 100% ASR and SCR for ARTS on the stop sign. For the speed limit sign, the attack causes a 100% ASR and a 0% SCR with a laser power of 115 mW and an average trace diameter of 28 cm, covering 10.6% of the stop sign. We observe that ILR requires higher power compared to the indoor setting because of the different outdoor illuminance compared with artificial light. We believe the degradation of SCR is due to illuminance instability, which we also notice in all our outdoor experiments.

Daytime Attack. During the day, we measured an average ambient light of 982 lux. In this case, we set a shorter distance $d_{as} = 1.5$ m to reach the required power and pattern size of our optimization methodology for safety constraints. For ARTS used for stop sign recognition, we achieve a 100% ASR and SCR, using a power of 226 mW and 31 cm trace diameter, equivalent to 13.1% of the stop sign surface. For the speed limit sign, we achieve a

100% ASR for both ARTS and LISA, using a power of 52 mW and an average trace diameter of 17.5 cm, covering an average of 13% of the speed limit sign. The SCR for ARTS and LISA on the speed limit sign are 50% and 90%, respectively. Finally, for GTSRB, we achieve an ASR of 100% and an SCR of 80% on speed limit with a laser power of 115 mW and an average trace diameter of 31 cm, covering 7.9% of the traffic sign surface.

Dynamic Driving Scenarios

We collect the attack traces using the same setup as in the static scenarios (§6.5.4). We recorded videos with the victim camera placed in a car moving towards the traffic sign (from 12 meters away from the traffic sign) at three increasingly high speeds: 5, 8, and 13 km/h (approximately 3, 5, and 8 mph)[1].

In this case, the ASR is calculated as the percentage of successful misclassification in terms of the number of successful frames among all the frames collected by the camera.

Results. Table 6.5.10 shows the ASR in the outdoor driving scenarios for the OnSemi camera. The results are consistent with the indoor robustness experiments in §6.5.2, i.e., the ILR attack achieves an ASR >99% for all the tested speeds on ARTS and LISA models. For the stop sign, on the other hand, we observe an ASR >90% for ARTS and >80.5% in GTSRB at all speeds. The ASR for the ARTS detection model is 100% in all scenarios. The low SCR for attacks on speed limit sign classification is due to the high sensitivity of the models for the speed limit classification compared to stop sign classification. Appendix §6.H Table 6.H.2 shows results for the OmniVision camera. These results show that our attack achieves high effectiveness in outdoor, moving scenarios, especially in night driving conditions.

[1]We did not evaluate at higher speeds due to the safety and spatial constraints of our testing facility. The demo videos of our experiments are available at https://sites.google.com/view/cav-sec/ilr-attack

Figure 6.5.5: Overview of the outdoor experimental scenarios. The setup is used in the daytime (left). The camera view during the attack is at nighttime (Right-top) and daytime (Right-bottom).

Table 6.5.10: ASR of the OnSemi camera in the outdoor driving scenarios.

	Stop Sign				Speed Limit			
	ARTS		GTSRB		ARTS		LISA	
Speed	ASR	SCR	ASR	SCR	ASR	SCR	ASR	SCR
Night Scenario								
5 km/h	**100%**	**100%**	99%	90%	**100%**	0%	99%	31%
8 km/h	**100%**	**100%**	92%	91%	**100%**	0%	**100%**	0%
13 km/h	**100%**	**100%**	85%	85%	**100%**	0%	99%	16%
Day Scenario								
5 km/h	98%	82%	85%	57%	**100%**	18%	**100%**	98%
8 km/h	**100%**	88%	88%	46%	**100%**	50%	**100%**	87%
13 km/h	91%	75%	80%	40%	**100%**	58%	**100%**	98%

6.6 Defense Evaluation

In this section, we evaluate existing defenses against patch attacks on ILR attacks and propose a new defense strategy.

6.6.1 Evaluation of generic defenses against patch attacks

While invisible to humans, ILR attack traces are visible in camera images. Thus, existing defense methods against adversarial patch attacks [167, 373] are theoretically applicable. So far, two types of defenses against adversarial patch attacks have been proposed: empirical defenses, such as the detection of anomalous patterns in attack patches [193, 182, 364, 252, 373, 376]), and certifiable defenses, which provide theoretical guarantees [359, 346, 234, 347]. As the empirical defenses are known to be generally vulnerable to adaptive attacks [359], we focus on certifiable defenses. PatchCleanser [347] is the current state-of-the-art for defending classifiers against adversarial patch attacks.

Experimental Setup. We evaluate the defense capability of PatchCleanser [347] on second-stage classifier models in the two-stage architecture. This architecture scales to a large number of classes and is thus applicable to a more general set of CAV systems. We use the same models as in §6.5.1 – CNN models trained on the ARTS, GTSRB, and LISA datasets. We generate ILR attacks against 20 scenarios with 2 lateral positions (0.5 m and 1 m) at 2 m from the traffic signs. We limit the diameter of the ILR trace to 12 cm, corresponding to approximately 10 pixels in the image. In order to focus on defense capabilities, we used our simulated, rather than physical, attack for our evaluation. PatchCleanser requires the estimated attack patch size as a parameter, thus, we set the patch size to 9 and 12 pixels. The 9-pixel patch is equivalent to the 2%-pixel patch scenario in [347] (note that this size is smaller than our ILR traces), while the 12-pixel patch (4%-pixel patch) is designed to cover an ILR trace 10 pixels in diameter. For the other parameters, we follow the official implementations [347].

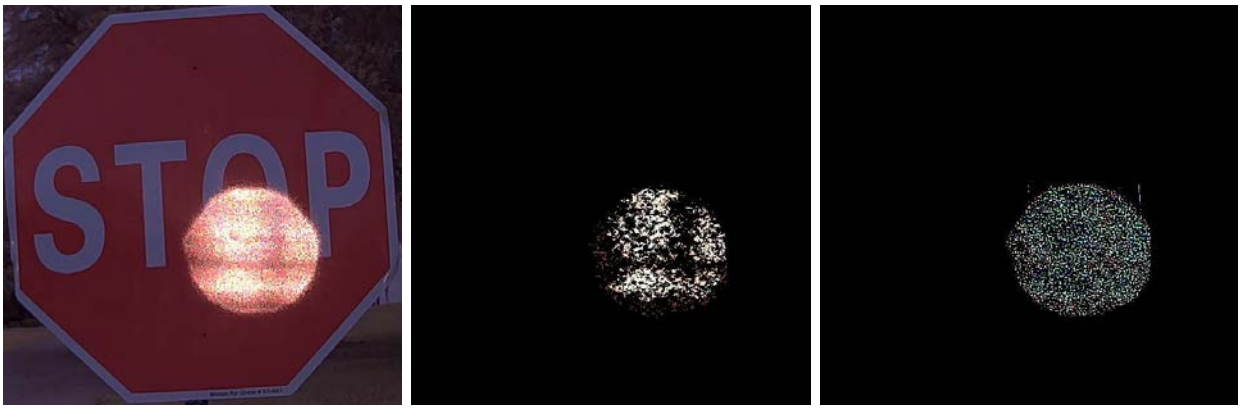
Results. Tables 6.6.1 and 6.G.1 (in Appendix 6.G), show the defense evaluation of PatchCleanser against the ILR attacks in the 2%-pixel patch scenario and 4%-pixel patch scenario, respectively. The accuracy without any defenses is the accuracy without the PatchCleanser.

The clean accuracy is the percentage of correct labels after PatchCleanser is applied. The certified accuracy is the percentage of correct labels that the PatchCleanser can certify. The mis-certified (false positive) FP is the percentage of *incorrect* labels PatchCleanser certifies.

As shown, our results indicate that PatchCleanser either does not benefit, or decreases, model performance for traffic sign recognition. For example, ILR attacks are not successful (100% accuracy without PatchCleanser) against the ARTS model on the stop sign because we limit the trace size. Nevertheless, PatchCleanser cannot handle them correctly as the clean accuracy is 0%. As shown in the **bold** numbers in Table 6.6.1 and 6.G.1, PatchCleanser degrades accuracy by an average of $\geq$12% for benign cases and 25% for attack cases for ILR attacks. We believe this is because PatchCleanser's main assumption does not apply to traffic sign recognition. PatchCleanser states that *"model predictions on images without adversarial pixels are generally correct and invariant to the masking operation."* This consideration generally holds for image classification tasks whose classes tend to be inferable even if some parts of the image are missing, such as the ImageNet [157] and CIFAR-10 [227] datasets. However, traffic sign recognition is an exception to this intuition. For example, a 35 mph sign could be classified as an 85 mph sign if the left side of "3" is altered [242]. Fig. 6.G.1 in Appendix 6.G shows examples of PatchCleanser mis-certifying examples of two-rounded, masked images. A two-round mask hides important text on the traffic sign and causes misclassification in all 36 combinations. Thus PatchCleanser's agreement-based defense strategy fails in those cases. Our evaluation shows that while certifiable defenses are typically considered more effective than empirical defenses [359], this does not mean that they are effectively applicable in every domain. More details are discussed in Appendix 6.G.

Table 6.6.1: Defense evaluation of PatchCleanser against the ILR attacks with the *2%-pixel patch*, as used in the original PatchCleanser paper [347]. The certified TP is the rate of correct labels that PatchCleanser can certify. The mis-certified FP is the rate of *incorrect* labels PatchCleanser certifies.

	Benign					Attack				
	Stop Sign		Speed Limit			Stop Sign		Speed Limit		
	GTSRB	ARTS	LISA	ARTS	Avg.	GTSRB	ARTS	LISA	ARTS	Avg.
No Defense Acc.↑	93%	93%	100%	93%	95%	15%	100%	0%	0%	29%
Clean Acc.↑	93%	93%	**71%**	**71%**	**82%**	15%	**0%**	0%	0%	**4%**
Certified Acc.↑	0%	64%	0%	0%	16%	0%	0%	0%	0%	0%
Miscertified FP↓	0%	0%	29%	21%	13%	5%	90%	100%	20%	54%



(a) Bounding box area of the captured image with ILR attack trace

(b) Image portion after speckle color analysis

(c) Spatial Frequency Evaluation

Figure 6.6.1: Speckle Detection on a stop sign. (a) Real-world attack trace during daytime. (b) Color masking result. (c) Spatial Frequency analysis of the selected portion.

6.6.2 Proposed Alternative Defense Strategies

While certifiable defenses are not sufficient to eliminate ILR, and applying optical IR filters defeats the advantage of using the IR light components to detect obstacles, alternative strategies can be adopted to evaluate the trustworthiness of traffic sign recognition. We propose a detection strategy based on the physics-based characteristics of laser light reflections.

Speckle Features. As described in §6.3.2, laser beam light generates speckle patterns

when coherent light diffuses off of a rough surface (such as that of a traffic sign), causing interference. The resulting reflected speckle pattern appears as a random distribution of bright and dark spots in images. The pattern varies based on speckle location, surface roughness, camera settings, captured images pixel resolution, and the optical power of the reflection, as shown in 6.4.1.

The spatial pixel frequency of an image refers to the rate of change in intensity values from one pixel to its neighboring pixels. Laser speckles exhibit high spatial frequency, indicating that adjacent pixels have significant variations in intensity in smaller spaces [180]. Additionally, the speckle can only manifest as a monotonous false color, such as magenta, purple, or orange, depending on image sensor type, the IR light, and the ambient light condition described in §6.2.2. Leveraging these unique features [179, 358], various defense strategies can be adopted to identify and locate ILR attack traces within images, independently of speckle size and shape.

Color-Frequency Detection. Taking inspiration from image processing techniques [296, 310, 177], a detection methodology can begin by extracting salient regions from captured traffic sign images that may contain false colors as a result of an ILR attack, A spatial frequency analysis is then performed to determine whether selected regions also manifest higher spatial frequencies, indicating a potential attack. For instance, for the OnSemi camera, our outdoor experiments show that at high ambient illuminance, the speckle color falls within the range of `#FFB266` to `#CC6600`, while during low ambient light, it ranges between `#CF9FFF` and `#DA70D6`. These color ranges can be extracted from the camera output and used as references for the subsequent frequency analysis.

Proof-of-Concept Analysis. To test the feasibility of the approach, as a proof of concept, we implement the strategy on a random selection of real-world images collected with the OnSemi camera from all of the outdoor attack scenarios tested in §6.5.4. We build two sample datasets containing both speed limit and stop signs, collected during daytime and nighttime

respectively. Each dataset consists of 75 benign samples and 75 attack samples. We then extract from the ARTS model output the traffic sign bounding box areas to perform the analysis. To apply the methodology, we first use a color masking technique, with potential IR light color ranges measured empirically. Next, to differentiate IR speckles from naturally occurring image components, we apply a Gaussian smoothing low-pass filter [177] to separate the low-frequency components of the image, and to retain regions with high-frequency components, as shown in Figure 6.6.1. We then identify a potential ILR attack if more than 1% of an extracted region exhibits high spatial frequencies. This threshold is chosen because benign traffic sign images typically contain few high spatial frequency components. This preliminary test achieves a 96% TPR and a 2.7% FPR for the daytime data. For the nighttime data instead, the methodology shows 92% TPR and 6.7% FPR. More sophisticated techniques such as color segmentation [144] and cross-correlation [339, 356] can also be used for detecting known patterns.

6.7 Discussion and Limitations

CAV Safety Implications. As demonstrated in §6.5.4, the ILR attack can achieve a nearly 100% ASR in outdoor driving scenarios, particularly at night. This can severely undermine CAV safety. Furthermore, we note that the ILR attack can be easily enhanced if the attacker uses multiple laser emitters to affect wider areas or to generate saturated IR speckles. Using our methodology, the attacker can alter traffic signs of any size by scaling the power and size of the attack trace proportionally, and of any color by adjusting the traces according to the traffic sign surface color, as shown in §6.4.1. Thus, we recommend that CAV companies either use IR filters, or deploy adequate defenses such as the one proposed in §6.6.2.

Single-Stage v.s. Two-Stage Architectures. We observe that both single-stage and two-stage architectures are vulnerable to the ILR attack in §6.5.1 and §6.5.2. The generic

object detector trained with the COCO dataset shows higher robustness to ILR, thus it might appear suitable to use for a single-stage architecture, or the first stage of a two-stage architecture. However, the single-stage architecture is not applicable to higher SAE autonomy levels [287], as it does not scale to a large number of traffic sign classes [166].

Daytime Attack under Strong Sunshine. While we confirmed that the ILR attack is robust to some lighting conditions in §6.5.2, the ILR attack is not likely to work in strong sunshine without raising the laser power to Class 4 (above 500mW) [17]. We could not evaluate this condition because of safety hazards and the uncontrollability of the sunshine level. However, our ILR attack should achieve at least equal or higher robustness than the attacks that use incoherent light, such as projection of an image of a person on the ground [261] or projection of an adversarial pattern onto a traffic sign [249] as they use incoherent lights not robust to reflection.

Laser Safety. All of our real-world experiments were conducted in closed controlled environments by trained personnel. For outdoor experiments, the power of a class 3-B laser was used (see Appendix 6.I for the details).

6.8 Conclusion

We propose ILR, a novel, invisible attack vector that can cause CAV traffic sign recognition systems to misclassify traffic signs. Our attack uses an IR laser to reflect a pattern onto a target sign that is invisible to humans, but visible to CAV cameras that lack IR filters. Unlike previous attacks, our attack does not require continuous aiming of a moving CAV.

To maximize attack efficacy, we designed a novel methodology to optimize the attack using image difference-based IR trace modeling and interpolation. We evaluated the effectiveness of our attack against two major traffic sign architectures achieving a 100% success rate for

indoor experiments and ≥80.5% in outdoor driving scenarios. Finally, we determined that certifiable defenses have limited applicability to the traffic sign recognition domain. Thus we propose an alternative defense technique based on speckle detection.

Appendix

6.A IR Laser Power Attenuation

The laser optical power emitted by the attacker laser module can be given by $P = I \cdot A$, where I is the beam's irradiance (power per unit area) and A is the cross-section area of the laser beam. For an attacker distance d_{as} from the target traffic sign, the irradiance of the beam decreases according to inverse square law as $I = P/(4\pi \cdot d_{as}^2)$. The cross-section area of the beam A at d_{as} can be measured as $A = (\pi \cdot d_{as}^2 \cdot \tan^2\theta)/4$, where θ is the divergence angle, controlled by the attacker using the two lens setup. Thus the resulting laser optical power at the traffic sign surface P_f at d_{as} with divergence angle θ can be given by $P_f = (P_a \cdot \tan^2\theta)/(4\pi)$.

6.B Laser Speckle Modeling Details

To accurately synthesize the pixel distribution of the attack traces, we use image differencing to extract the RGB pixel intensity variation in our controlled closed indoor scenario (we follow a similar procedure for outdoor scenarios except for the illuminance setting).

For the indoor setting, we use the same attack setup described in §6.3.3 and we locate the victim camera at a 0.5 m d_{av} from the IR emitter. We set d_{as} to 3 m and the room ambient light to 100 Lux. We then capture the images of a stop sign without and during the attack

and extract the pixel intensity variation caused by the IR patterns in the form of RGB pixel difference between both images. These intensity differences are then applied on to the benign traffic sign image to simulate IR beams targeted at different sign locations for optimization as shown in Fig. 6.3.2.

We account for the temporal image noise in the target camera by averaging ten consecutive frames for each benign and attack case. Further, we observe that the RGB pixel offset values depend on the camera's perceived surface color for the selected traffic sign. To model this, we first collect the attack traces with a baseline traffic sign surface color (e.g., the RGB pixel values that correspond to the red color of the stop sign). We then synthesize the IR pattern on the target traffic sign surface colors (different from the baseline), by measuring the average offset in the RGB values between the baseline and the target traffic sign color.

6.C Pixel-wise Spline-based Interpolation

As a baseline method for the trace image interpolation, we design the pixel-wise spline-based interpolation in which we simply apply the cubic spline interpolation for each pixel. This method consists of three steps, (1) Spline fitting: The real-world traces are synthesized by a pixel-wise cubic interpolation function to model the RGB pixel distribution changes for each trace individually; (2) Spline interpolation: For a desired emitted laser power, two adjacent real-world traces are used to calculate the RGB pixel values of the trace; and (3) using weighted averages between the real-world traces and the traces generated in step (2), the RGB pixel values are derived at the desired diameter.

6.D CNN Model Architecture

Table 6.D.1 lists the architecture of the CNN model we use. The input image size is 60×60 pixels. This model achieves one of the highest performances on the GTSRB dataset [299].

Table 6.D.1: CNN model architecture.

Layer (type)	Output Shape
0. Input	(batch_size, 60, 60, 3)
1. Conv2D	(batch_size, 28, 28, 16)
2. Conv2D	(batch_size, 26, 26, 32)
3. MaxPooling2D	(batch_size, 13, 13, 32)
4. BatchNorm	(batch_size, 13, 13, 32)
5. Conv2D	(batch_size, 11, 11, 64)
6. Conv2D	(batch_size, 9, 9, 128)
7. MaxPooling2	(batch_size, 4, 4, 128)
8. BatchNorm	(batch_size, 4, 4, 128)
9. Flatten	(batch_size, 18432)
10. Dense	(batch_size, 512)
11. BatchNorm	(batch_size, 512)
12. Dropout	(batch_size, 512)
13. Dense	(batch_size, 43)

6.E Detailed Setup of Random Attacks

A naive attacker might decide to project the IR pattern onto random locations on a sign and observe the behavior of the victim CAV. To show the consequences of a random attack on both single-stage and two-stage architectures, we use the setup discussed in §6.3.3 to collect the IR traces from the OnSemi camera [233] for the stop and speed limit signs. For this analysis, we set the pattern diameter to $D = 15$ cm as in our indoor evaluation. To avoid sensor saturation, we used a 51 mW laser power. We then randomly pick a location to place the IR pattern within the traffic signs. As shown in Table 6.5.3, the random attack has an ASR of $\leq$20% against the stop sign, and $\leq$20% against the speed limit sign, for the two architectures. We use the random attack as a comparison baseline to demonstrate our

optimized ILR attack methodology's effectiveness in §6.4.

6.F Robustness to Inaccuracy in First-Stage Object Detection

While we focused more on attacking the second-stage classification model for the two-stage architecture, we also evaluated how inaccuracies in the first stage can change the automatic bounding cropping and consequently alter the input of the second-stage classification model. To evaluate the impact of the inaccuracy on the classification results, we apply vertical and horizontal translation noise to our manually annotated bounding boxes. For the stop sign, we use the CNN model trained on the GTSRB dataset. For the speed limit sign, we use the CNN model trained on the LISA dataset. Fig. 6.F.1 shows the ASR for random vertical and horizontal displacement. Since the bounding box sizes are different for each image, we use the percentage over the width and height of the bounding box as a displacement level, δ, instead of the corresponding sizes in pixels. Given δ, we generate a random number under the uniform distribution $\mathcal{U}(-\delta, \delta)$ and displace the bounding box based on the result. For example, a 10-pixel displacement will be applied on a bounding box with 100-pixel height and width if the random number is 10%. As shown, bounding box inaccuracy has a greater impact on the stop sign than the speed limit sign. The ASR and SCR for the stop sign decrease with increasing noise levels. In contrast, the ASR for the speed limit sign is always 100%, while the SCR eventually starts to decrease around a noise level of 8%.

We hypothesize that these results are due to the shape, and the resulting pixel RGB values, of the attack beam traces. For example, the majority of attacks against the speed limit signs are classified as stop signs, as shown in Fig. 6.3.2. This suggests that the beam and stop sign may have similar features, which result in similar classifications. Thus, small translations of

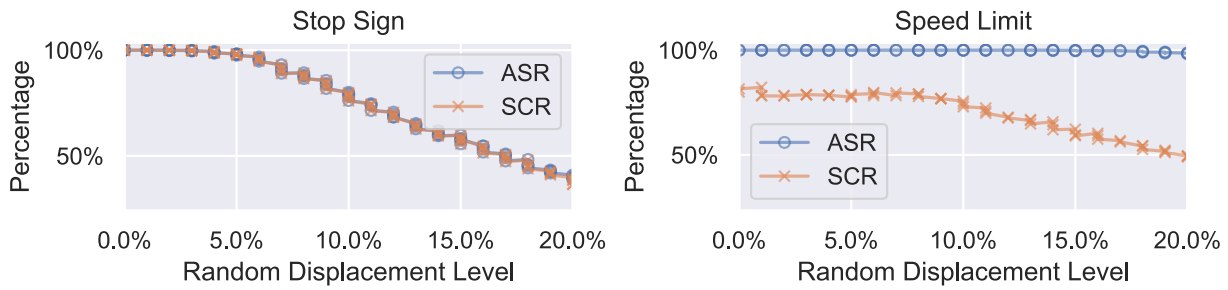


Figure 6.F.1: Evaluation results of bounding box position noise detected in the first stage. Given noise level δ, the resulting perturbation, $\mathcal{U}$, obeys: $\mathcal{U}(-\delta, \delta)$ = percentage of bounding box width or height.

the IR spot can negate attacks, resulting in the correct stop sign classification.

6.G Considerations over PatchCleanser

Table 6.G.1: Defense evaluation of PatchCleanser against the ILR attacks with the *4%-pixel patch*, which can cover the ILR trace. The certified TP is the rate of correct labels that PatchCleanser can certify. The miscertified FP is the rate of *incorrect* labels but PatchCleanser certifies.

	Benign					Attack				
	Stop Sign		Speed Limit			Stop Sign		Speed Limit		
	GTSRB	ARTS	LISA	ARTS	Avg.	GTSRB	ARTS	LISA	ARTS	Avg.
No Defense Acc.↑	93%	93%	100%	93%	95%	15%	100%	0%	0%	29%
Clean Acc.↑	**86%**	**43%**	**64%**	**71%**	**66%**	15%	**0%**	0%	0%	**4%**
Certified Acc.↑	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Miscertified FP↓	7%	50%	36%	21%	29%	0%	100%	100%	85%	71%

PatchCleanser and PatchGuard [346], and current state-of-the-art defenses, assume that classification models can return a correct prediction even if a small portion of the image is masked. However, this assumption does not always hold for traffic sign recognition, since any portion of the sign has the potential to be important for correct classifications. As listed in Tables 6.6.1 and 6.G.1, the certified accuracy of PatchCleanser is significantly lower than the reported (>60% certified accuracy) on ImageNet [157]. Even for benign cases, the certified accuracies are 16% for the 2%-pixel patch scenario and 0% for the 4%-pixel patch scenario.

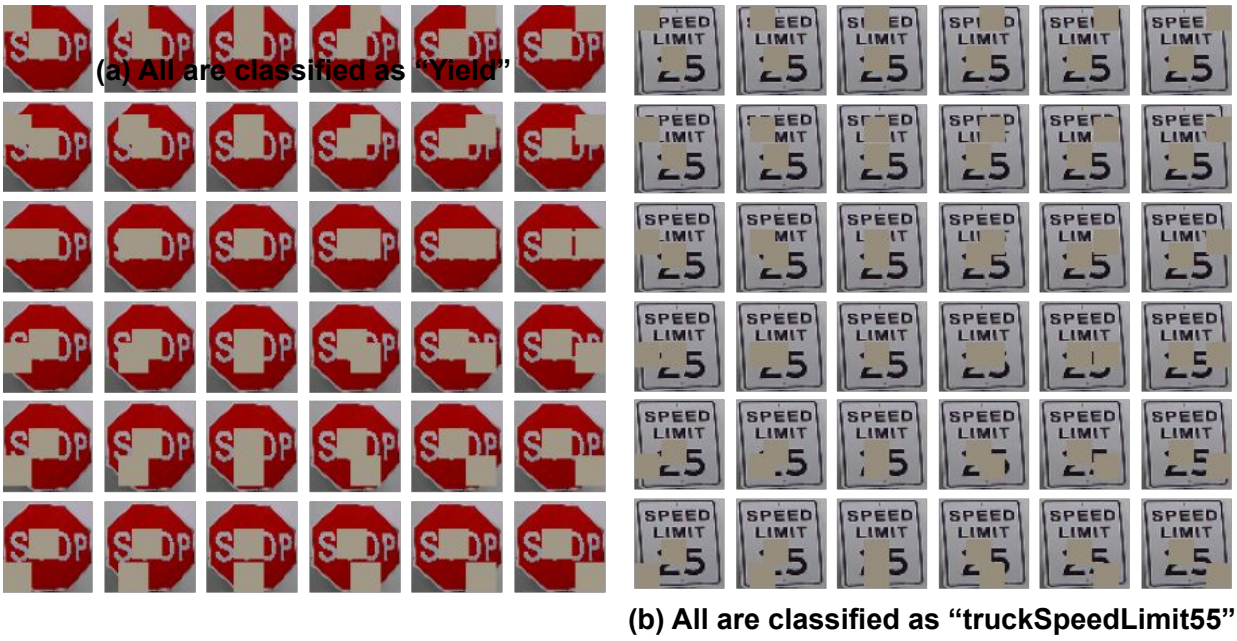


Figure 6.G.1: Mis-certified examples of two-round masked images. (a) For the stop sign, all images are classified as a "Yield" sign in the 4%-pixel patch scenario. (b) For the speed limit sign, all images are classified as a "truckSpeedLimit55" sign in the 2%-pixel patch scenario.

In the 2%-pixel patch scenario, the clean accuracy in the benign case (82% on average) is close to the reported clean accuracy (>80%) on ImageNet, but it drops to a 66% average in the 4%-pixel patch scenario. For the attack cases, the clean accuracies are even worse (4%) than the accuracy without PatchCleanser (29%). Furthermore, PatchCleanser mis-certifies 33.5% of cases for the 2-pixel patch scenario and 50% of cases for the 4-pixel patch scenario (averages of the underlined numbers in Tables 6.6.1 and 6.G.1) and does not have any correctly certified cases for the 4-pixel patch scenario. This means that the two-round masking of PatchCleanser itself works as an attack, with prediction agreement occurring for a wrong label. Fig. 6.G.1 shows mis-certified examples of the two-round masked images. The two-round mask hides important text on the traffic sign and causes misclassification in all 36 combinations. These images might also be challenging for humans to classify correctly.

Our ILR attack can break another prerequisite for PatchCleanser – that the attack trace size is known in advance. The size of an ILR attack trace is relative to the target sign size and

it can be increased without reducing attack stealthiness using our methodology described in 6.4 (note that to maintain a constant trace intensity, the attacker would need to change the laser power based on the distance). Additionally, the circular shape of the ILR attack trace cannot be used in PatchCleanser as it is since a significant number of patches are required for circular masks to meet the $\mathcal{R}$-covering conditions.

6.H Outdoor Evaluation of the OmniVision Camera

Table 6.H.1: ASR of ILR attacks on Omnivision in the outdoor static scenarios.

		Night.		Day	
		ASR	SCR	ASR	SCR
Stop	ARTS	100%	100%	100%	20%
Sign	GTSRB	100%	100%	100%	90%
Speed	ARTS	100%	100%	100%	50%
Limit	LISA	100%	0%	100%	100%

Table 6.H.2: ASR of ILR attacks on Omnivision in the outdoor dynamic scenarios.

	Stop Sign				Speed Limit			
	ARTS		GTSRB		ARTS		LISA	
Speed	ASR	SCR	ASR	SCR	ASR	SCR	ASR	SCR
Night Scenario								
5 km/h	100%	95%	100%	92%	95%	0%	100%	0%
8 km/h	100%	78%	100%	85%	89%	0%	100%	6%
13 km/h	100%	85%	100%	90%	96%	0%	100%	1%
Day Scenario								
5 km/h	100%	54%	99%	39%	100%	18%	100%	96%
8 km/h	100%	10%	99%	94%	100%	50%	100%	100%
13 km/h	100%	11%	100%	80%	100%	58%	100%	100%

6.I Considerations on Laser Safety

In our experiments, we use the maximum emission power of 80mW in controlled indoor scenarios and 115mW in outdoor controlled scenario (daytime), below the 3-B class laser limit

(= 500mW) [17]. The maximum permissible exposure (MPE) [326] of a 780 nm continuous class 3-B laser with an exposure time $t > 10$ seconds is given by $MPE = 10^{2\cdot(w-0.7)} \cdot 10^{-3}$, where w is the wavelength of the IR laser. Using this equation, at $w = 780$ nm, $MPE = 0.33$ mW/cm^2. As an example, for a given 45 mW optical power, the emitter energy is equivalent to 57.7 mW/cm^2, nearly 175 times more than the MPE. However, the IR beam's energy can be reduced to below the MPE values by increasing the beam diameter to 3.6 times the original size (1.3 cm for our setup). From this analysis, an IR pattern diameter of nearly 5 cm is required in order to follow MPE guidelines. Using our ILR attack configuration, which considers a diverging beam, the resulting IR pattern diameter at 45 mW is 17 cm.

Chapter 7

Intriguing Properties of Diffusion Models: An Empirical Study of the Natural Attack Capability in Text-to-Image Generative Models

7.1 Introduction

Denoising diffusion probabilistic models (DDPM) [197], or simply diffusion models, have shown breakthrough performance in image generation. Once the DDPM demonstrates the generation capability of photo-realistic images and human-level illustrations, numerous diffusion models, such as DALL-E 2 [278], Stable Diffusion [285], and Firefly [100], are actively released and widely available through APIs or as open-source models. This technical breakthrough has facilitated many applications in various fields such as the arts [369], medicine [220], and autonomous driving [211]. While diffusion models have brought sig-

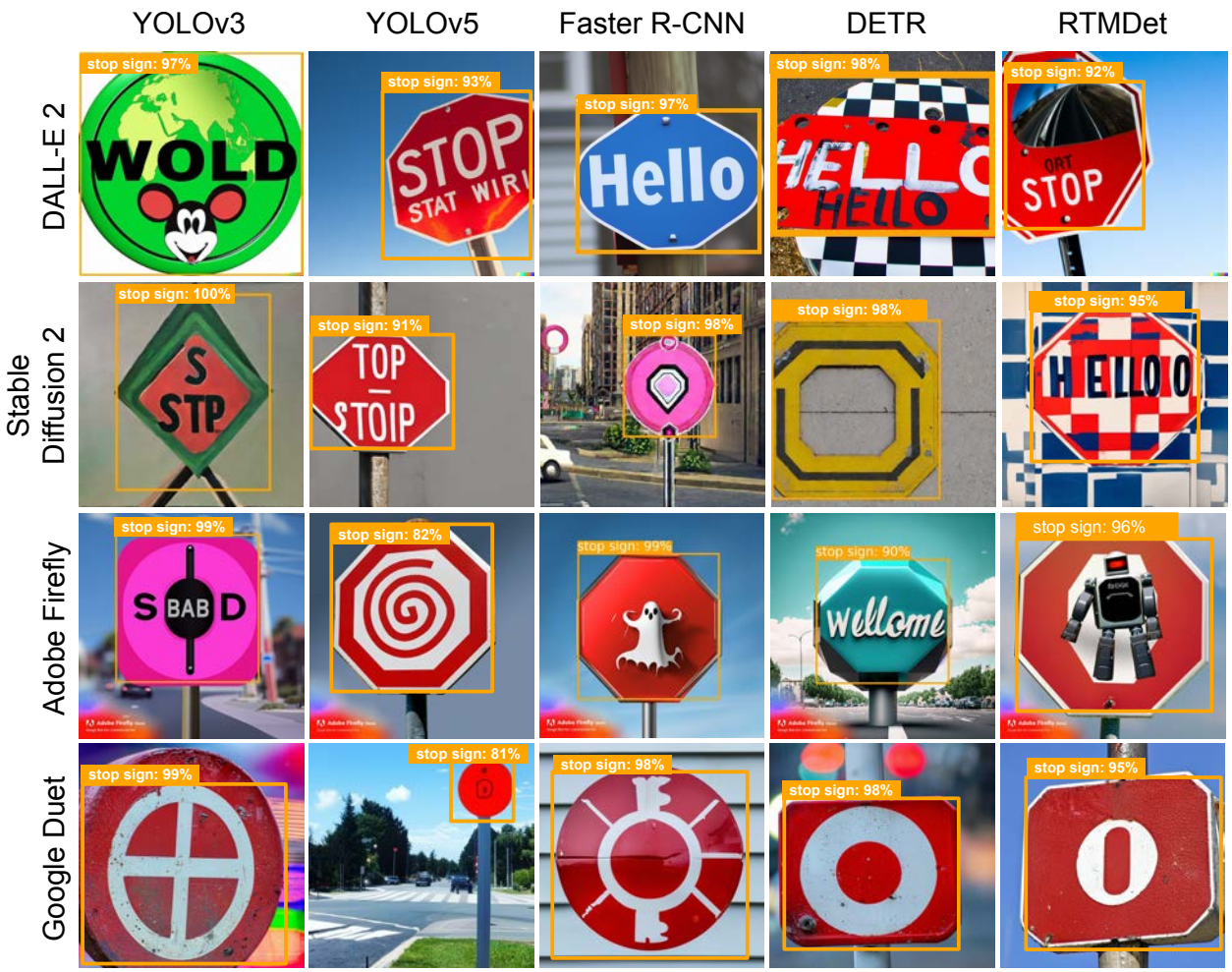


Figure 7.1.1: Examples of the natural attack capability in diffusion models (row). The images are generated with prompts that intentionally remove essential features to humans while keeping "stop sign" in the prompt. Even without these essential features, object detectors (column) still detect these objects with high scores.

nificant benefits to these areas, recent studies have also raised concerns regarding the new security and privacy risks introduced by diffusion models. Chen et al. [138] show that the diffusion models can generate more transferable and imperceptible adversarial attacks. Chen et al. [142] generate more natural and stealthy perturbations with guidance from diffusion models. Carlini et al. [133] demonstrate that the diffusion models memorize training images and can emit them.

These recent studies motivate us to further investigate the security risks of diffusion models. In this work, we discover simple but intriguing properties of the images generated by text-

to-image diffusion models, in which text prompts guide the image generation process via a contrastive image-text supervision model, such as OpenAI CLIP [276]. Fig. 7.1.1 shows representative examples that motivate this work. We generate the stop sign images using state-of-the-art diffusion models with text prompts that intentionally break the fundamental properties that humans use to identify stop signs (e.g., red color and octagonal shape) in the human visual system (HVS) [184, 173, 206], while the text prompt still contains the object name "stop sign".

As shown, the diffusion models faithfully follow our instructions and generate images that should not be recognized as stop signs since legitimate stop signs should not be blue or rectangular. However, many state-of-the-art Deep Neural Network (DNN)-based object detectors still recognize these examples as stop signs with surprisingly high confidence. These results suggest that these object detectors can be highly affected by imperceptible features embedded by diffusion models. We recognize this phenomenon as a new type of adversarial attack because the fundamental properties of the target object should be removed by maliciously designed text prompts. We name it the Natural Denoising Diffusion (NDD) attack and pursue the following question in this study:

Do images generated by diffusion models have natural attack capability against DNN models?

The rest of this paper is structured to validate the question. We first construct our dataset, named the Natural Diffusion Denoising Attack (NDDA) dataset, to systematically understand the natural attack capability in diffusion models (§7.3.1). We use 3 state-of-the-art diffusion models to collect the images with and without robust features that play essential roles in the HVS. Following prior works [184, 173], we define 4 robust features: shape, color, text, and pattern.

Secondly, we conduct an attack capability analysis of the NDD attack against state-of-the-art object detection models (§7.3.2) and image classification models (§7.3.3) on the NDDA

dataset. We find that object detectors and classifiers typically maintain their detection, even though the text prompt guides them to remove robust features entirely or partially. For example, 32% of the generated stop signs are still detected as stop signs even though all robust features are guided to be removed. This result means that text-to-image diffusion models embed intriguing features that are imperceptible to humans but generalizable to DNN models if the subject word (e.g., stop sign) is in the prompt. We confirm that all diffusion models we evaluate have the natural attack capability to enable the NDD attack.

Third, we conduct a large-scale empirical study to further evaluate the validity and quantify the natural attack capability in diffusion models by answering 6 novel research questions (§7.4). We conducted a user study to evaluate the stealthiness of the NDD attacks because valid adversarial attacks need to be not only effective against the DNN models but also stealthy against humans. For example, humans will not be fooled if the robust features are not correctly removed since it should remain a legitimate stop sign. As a result, we identify the high stealthiness of the NDD attacks: The stop sign images generated by altering their "STOP" text have 88% detection rate against object detectors while 93% of humans do not regard it as a stop sign. Furthermore, we evaluate the impact of the non-robust features that are predictive but incomprehensible to humans on the natural attack capability in diffusion models with an analysis inspired by Ilyas et al. [206], which proposes a methodology to train "robustified" classifiers against the non-robust features. By comparing the robustified and normal classifiers, we illustrate that the non-robust features play a meaningful role in the natural attack capability in diffusion models.

Finally, we discuss our findings and the limitations (§7.5). In summary, the contributions of this work are as follows:

- We discover a new security threat, the NDD attack, which exploits the natural attack capability of the diffusion model to generate model-agnostic and transferable adversarial

attacks via simple text prompts that are designed to remove robust features.

- We construct a new large-scale dataset, named the NDDA dataset, to systematically evaluate the natural attack capability in diffusion models. We cover all 4 robust features essential to the HVS: shape, color, text, and pattern.
- We performed a large-scale empirical study to systematically evaluate the natural attack capability by answering 6 research questions. The NDD attack can achieve an attack success rate of 88%, while being stealthy for 93% of human subjects in the stop sign case. We also find that the sensitivity to the non-robust features has a high correlation with the natural attack capability.
- We confirm the model-agnostic and transferable attack capability of the NDD attack on a Tesla Model 3, which identifies 8 out of 11 (73%) printed attacks as stop signs.

Dataset release. NDDA dataset is on the our website [98].

7.2 Related Work

7.2.1 Denoising Diffusion Model

DDPM [197] is a generative model that exploits the intuition behind nonequilibrium thermodynamics. Training a diffusion model involves both forward and reverse diffusion processes. In the forward process, the model perturbs a clean image with Gaussian noise. In the reverse process, the diffusion model learns to remove this noise for the same number of time steps. In short, diffusion models are image denoisers that learn to reconstruct the original images from noisy images. This procedure is simple but shows remarkable performance in producing high-quality images. Dhariwal et al. [160] shows that diffusion models achieve much higher quality metrics, such as FID, inception score, and precision, than prior works such as GANs [120].

The text-to-image diffusion model [278, 288, 285] is a variant of diffusion models that can flexibly control the output image via text prompts. Due to its easy usability, major diffusion models (e.g., DALL-E 2, Stable Diffusion, and FireFly) adopt this text-based guidance. To inform the text information in the generation process, contrastive image-text supervision models, such as CLIP [276] and OpenCLIP [205], are integrated into their training and inference processes.

7.2.2 Adversarial Attacks

DNN models are known to be generally vulnerable to adversarial attacks [316, 178, 370, 168, 251, 129], which can alter the predictions of DNN models by adding small changes that are not noticeable to humans. The early-stage works [316, 178] originally use a subtle imperceptible perturbation on the entire input image as their attack vector. Recent research has identified that adversarial attacks can be achieved by broader attack vectors that are any stealthy changes to the human perception such as putting small patches [340, 359, 292, 300] or placing stickers on the target [168]. Natural adversarial examples [195] demonstrate that even clean natural images can be used as adversarial attacks. To this extent, the NDD attack is similar to the natural adversarial examples, but the natural adversarial examples do not have any guarantees that can generate an attack against targeted scenarios (e.g., stop sign) as they just find out-of-distribution samples in the existing images. Furthermore, we find that the non-robust features [206] play a large role in the NDD attack (RQ4 in §7.4). Ilyas et al. [206] report that the adversarial attacks are not caused by a bug in DNNs but instead by non-robust features that are predictive but incomprehensible to humans. We thus consider that the NDD attack is enabled by similar root causes as the traditional adversarial attacks, i.e., enabled by non-robust features, rather than by out-of-distribution samples.

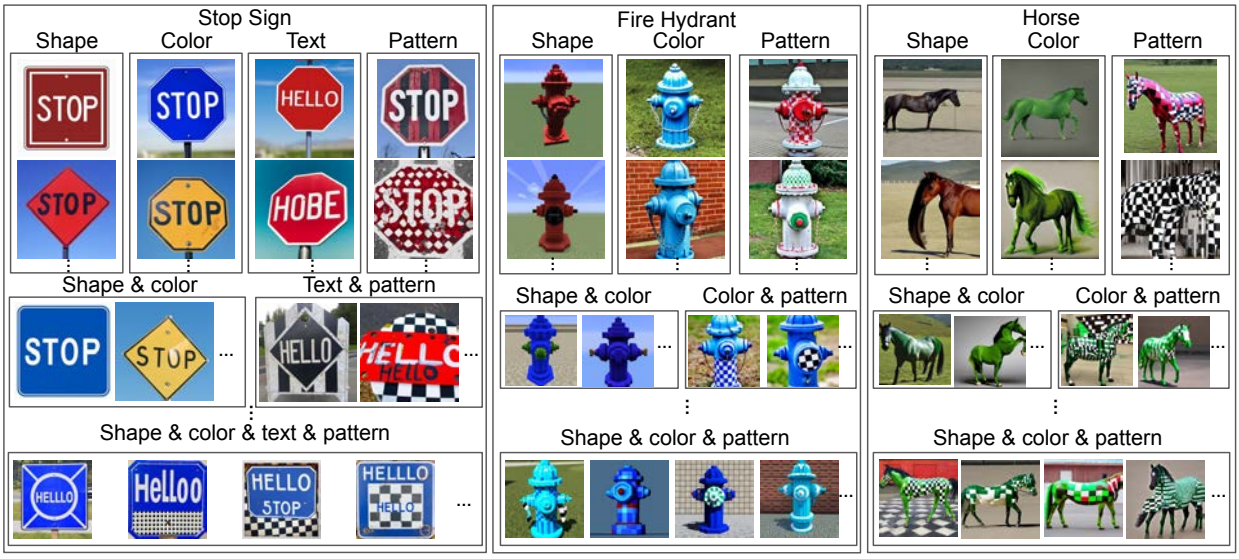


Figure 7.3.1: Overview of the Natural Denoising Diffusion Attack (NDDA) dataset. We alter or remove the 4 types of robust features partially or entirely. For the stop sign, we alter the text on it considering its importance to be recognized as a stop sign. For each set of robust features, we generate images with 3 diffusion models for 3 object classes.

Table 7.3.1: Templates of text prompts and examples for the "stop sign" object to remove the 4 robust features.

Removed Robust Features		Text prompt to remove/alter robust features	
		Format	Example: Stop sign
Benign prompt		[Subject]	Stop sign
Shape		[Shape] [Subject]	Square stop sign
Color		[Color] [Subject]	Blue stop sign
Texture	Text	[Subject] with "[Text]" on it	Stop sign with "hello" on it
	Pattern	[Subject] with a [Pattern] paint on it	Stop sign with a checkerboard pattern paint on it

7.3 Attack Capability Analysis

To systematically evaluate the natural attack capability of the diffusion models, we first construct a new large-scale dataset, called the Natural Denoising Diffusion Attack (NDDA) dataset. With the NDDA dataset, we then confirm the effectiveness of the NDD attack against state-of-the-art object detectors and image classifiers.

7.3.1 NDDA Dataset Design

The design goal for the NDDA dataset is to systematically collect images both with and without robust features upon which human perception relies. For this sake, the major challenge is to identify what kinds of robust features are essential in our human visual system (HVS) for object recognition. Although the complete mechanism of the HVS is not fully understood, prior works [184, 173] identify that shape, texture, and color are the most important features for the HVS to identify objects. Therefore, in this study, we follow the prior works and define them as robust features for object recognition. To further explore the motivated examples in Fig. 7.1.1, we decompose the texture into text and pattern because text has a special meaning for human perception. For example, people may not consider a sign to be a stop sign if it does not have the exact text "STOP" on it.

Table 7.3.1 lists the templates of text prompts and examples for the "stop sign" label to remove or alter the 4 robust features. In this case, we consider 16 different combinations with and without robust features and generate 50 images for each combination. The 3 object classes are selected from the classes of the COCO dataset [243]: a stop sign for an artificial sign, a fire hydrant for an artificial object, and a horse for a natural object. We select these 3 classes because of their relatively higher detection rates than others in our preliminary experiments on generated images. We adopt the COCO's classes to make the experiments easier as we can utilize many existing pretrained models on the COCO dataset. Fig. 7.3.1 shows an overview of our datasets. More details are in tAppendix.

7.3.2 Attack Capability against Object Detectors

We evaluate the attack capability of the NDD attack against object detectors with the NDDA dataset to validate the generality of the motivated examples shown in Fig. 7.1.1.

Experimental setup. We obtain the inference results of all images in the NDDA dataset with 5 popular object detectors: YOLOv3 [282], YOLOv5 [215], DETR [131], Faster R-CNN [283], and RTMDet [250]. For YOLOv5, we use their official pretrained model; For the others, we use the pretrained models in MMDetections [139]. All models are trained on the COCO dataset [243]. We use 0.5 as the confidence threshold for all models.

Results. Table 7.3.2 shows the detection results of the stop sign images in the Natural Diffusion Attack dataset generated by 3 diffusion models. The detection rate is calculated by whether one or more stop signs are detected in the input image. As shown, the majority of the images are still detected as stop signs even though we remove a robust feature. While YOLOv5 shows slightly higher robustness, all object detectors still detect stop signs in the majority of images: On average, the detection rate for all object detectors is $\geq$37%, meaning that 37% of the images generated by the diffusion models have the potential to be used as adversarial attacks. We also observed similar results on the other labels. More detailed results are in Appendix.

7.3.3 Attack Capability against Image Classifiers

We also observe that the NDD attack is highly effective against image classifiers. For example, $\geq$47% of the generated stop sign images are still classified as stop signs even though all 4 robust features are removed. More details on the setup and results are in Appendix.

7.3.4 Implications of Attack Capability Analysis

The NDD attack shows quite high attack effectiveness against both object detectors and image classifiers. These results imply that diffusion models can be utilizable to generate model-agnostic and highly transferable adversarial attacks with significantly less effort than

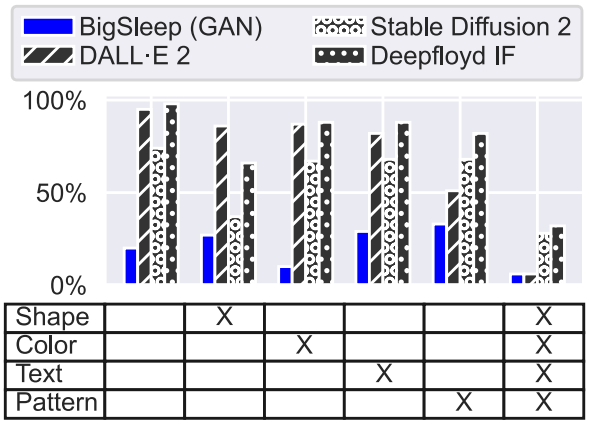
prior works [134, 252, 145] that need iterative attack optimization processes. However, the current analysis is not sufficient to fully conclude the vulnerability of DNN models against NDD attacks because adversarial attacks must satisfy 2 requirements: effectiveness against DNN models and stealthiness to humans. For example, diffusion models may ignore the text prompts and just merely generate legitimate stop signs. In the next section, we systematically evaluate the stealthiness of the NDD attack and explore potential root causes.

Table 7.3.2: Detection rates of 5 object detectors on the stop sign images in the NDDA dataset generated by the 3 diffusion models. **Bold** and underline denote highest and lowest scores in each row.

	Removed Robust Features				Object Detectors					
	Shape	Color	Text	Pattern	YOLOv3	YOLOv5	DETR	Faster R-CNN	RTMDet	Avg.
DALL-E 2					**100%**	76%	**100%**	**100%**	**100%**	95%
	✔				**98%**	36%	**98%**	**98%**	**98%**	86%
		✔			**98%**	48%	94%	**98%**	**98%**	87%
			✔		82%	32%	**100%**	**100%**	94%	82%
				✔	52%	10%	50%	52%	**90%**	51%
	✔	✔	✔	✔	**12%**	0%	6%	4%	8%	6%
				Avg.	74%	34%	75%	75%	**81%**	68%
Stable Diffusion 2					74%	50%	84%	**86%**	76%	74%
	✔				30%	24%	46%	**60%**	26%	37%
		✔			**78%**	40%	**78%**	72%	66%	67%
			✔		58%	56%	**90%**	70%	66%	68%
				✔	56%	48%	**90%**	76%	72%	68%
	✔	✔	✔	✔	30%	6%	**48%**	30%	24%	28%
				Avg.	54%	37%	**73%**	66%	55%	57%
Deepfloyd IF					**100%**	88%	**100%**	**100%**	**100%**	98%
	✔				68%	52%	70%	**84%**	56%	66%
		✔			**100%**	58%	92%	94%	94%	88%
			✔		84%	78%	94%	**96%**	88%	88%
				✔	80%	64%	88%	**90%**	88%	82%
	✔	✔	✔	✔	**60%**	0%	34%	34%	32%	32%
				Avg.	82%	57%	80%	**83%**	76%	76%

7.4 Empirical Analysis

We perform an extensive empirical study to further explore the characteristics and root causes of the NDD attack by answering 6 research questions (RQs).



Shape		X				X
Color			X			X
Text				X		X
Pattern					X	X

Figure 7.3.2: Average detection rate of stop sign images over the 5 object detectors for 4 models. The "x" mark means the removed robust features.

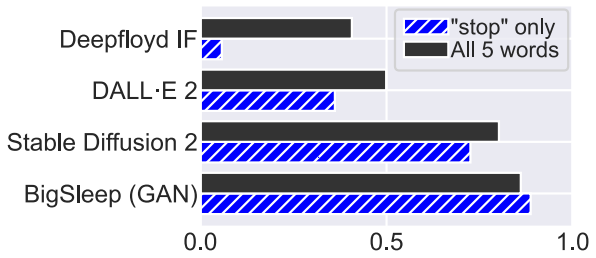


Figure 7.3.3: Averaged normalized Levenshtein distances between the given word and the detected text by OCR. The black bar is the averaged distance of all 5 words; the blue bar is only for the "stop" word.

7.4.1 RQ1: Does the natural attack capability exist in the previous image generation models?

We first investigate whether the natural attack capability exists in the previous image generation models.

Experimental setup. As the state-of-the-art image generation methods before diffusion models, we evaluate BigGAN [120]. To generate images guided by text prompts, we use BigSleep [101] that guides the image generation process of BigGAN by OpenAI CLIP [276]. We evaluate the 5 object detectors in Table 7.3.2. We use a detection threshold of 0.5. For each model, we evaluate 6 combinations of partial or complete removal of the robust features.

Results. Fig. 7.3.2 shows the average detection rate of stop sign images generated by each image generation model over the 5 object detectors. As shown, all diffusion models have significantly higher detection rates than the BigSleep GAN model. In particular, Stable Diffusion 2 and Deepfloyd IF have detection rates of $\geq$28% even if all robust features are removed or altered. Meanwhile, BigSleep has much lower detection rates (7.5%). We thus consider that *the natural attack capability has existed slightly in the prior image generation*

models, but it becomes significantly severe in diffusion models. Diffusion models have higher image generation capabilities than GANs, but it may also enhance their ability to inject more non-robust features into generated images.

7.4.2 RQ2: How stealthy is NDD attack against humans?

To be a valid adversarial attack, the NDD attack should satisfy the two requirements: effectiveness against DNN models and stealthiness against humans. We so far have confirmed the effectiveness against DNN models as in Table 7.3.2, but it does not mean that the attack is stealthy. For example, the diffusion models may ignore the text prompts and just generate legitimate objects. To answer the questions, we perform a user test to investigate the stealthiness against humans and the validity of generated images.

Experimental setup. We recruited 82 human subjects on Prolific [56], a crowdsourcing platform specialized for research purposes. Human subjects are asked to answer yes or no to whether the object of interest (e.g., stop sign and fire hydrant) is in the presented image or not. Considering the reasonable experimental time for human subjects to maintain their concentration, image generation models were limited to the following: Deepfloyd IF, the diffusion model with the highest detection rates in Table 7.3.2, and BigSleep, a state-of-the-art GAN-based model. We generate 3 images per text prompt for each image generation model; for the baseline, 3 real images are presented. The evaluation images were chosen from a pool that can fool at least one object detector in Table 7.3.2, i.e., all the images are valid attacks. More details are in our questionnaire form [95].

Results. Table 7.4.1 lists the results of our user study. The detection rate is the proportion of users who answer "yes", i.e., they identify the target object in the presented image. As shown, DeepFloyd IF's images on the benign prompts have similarly high detection rates as the real images. On the other hand, the BigSleep images are not detected as the target

objects as the detection rates are ≤4%. This indicates that the images generated by the state-of-the-art GAN-based model are not only not perceived by humans but also are not effective attacks against object detectors, i.e., the generated images are far from realistic. For the images without robust features, their detection rates are much lower than the real images and the images of benign prompts. We observe that the different objects have different sensitivities to each robust feature. For example, the text is very important to the stop sign as the human detection rate is 7% when the text on it is altered. DeepFloyd IF thus can easily generate effective adversarial attacks for stop signs because 93% of the human subjects did not see the images as stop signs even though 88% of the generated images are detected by stop signs as in Table 7.3.2. This result indicates that *the natural attack capability in diffusion models can indeed generate valid adversarial attacks that are stealthy to humans.*

7.4.3 RQ3: Does the incapability of text generation correlate with the natural attack capability?

We observed that the NDD attack has a high stealthiness to humans through our user study and found that the different objects have different sensitivities to different robust features. In particular, the text on a stop sign shows high importance for humans to identify a stop sign as in Table 7.4.1. Meanwhile, the object detectors are influenced by the shape or pattern rather than the text as in Table 7.3.2. Motivated by this, we further evaluate the impact of text generation capability in the natural attack capability.

Experimental setup. We generate the images with the 3 diffusion models (DALL-E 2, Stable Diffusion 2, and DeepFloyd IF) and on the GAN-based model (BigSleep). We use the text prompt format: "text of [word]", with "[word]" being one of the following: hello, welcome, goodbye, script, and stop; 20 images per word are generated with different seeds. Given a "[word]", we apply an optical character recognition (OCR) method, specifically the pipeline

Table 7.4.1: Results of our user study. The detection rate is the ratio that the human subjects identify the targeted object in the presented image. Cell color means the magnitude of the detection rates: Greener means that more people think the object is in the image, and reddish means that fewer people think so.

	Real Image	Benign Prompt		Removed Robust Feature	Detection Rate
Stop sign	65%	BigSleep	4%	Shape	3%
				Color	1%
				Text	2%
				Pattern	1%
				All	1%
		Deepfloyd	56%	Shape	19%
				Color	11%
				Text	7%
				Pattern	25%
				All	8%
Horse	82%	BigSleep	1%	Shape	1%
				Color	2%
				Pattern	4%
				All	4%
		Deepfloyd	59%	Shape	7%
				Color	14%
				Pattern	2%
				All	7%
Fire hydrant	88%	BigSleep	3%	Shape	0%
				Color	2%
				Pattern	1%
				All	2%
		Deepfloyd	57%	Shape	1%
				Color	31%
				Pattern	48%
				All	2%

provided by the Keras-OCR package [260], for generated images and calculate the normalized Levenshtein (edit) distance between [word] and the recognized sentence(s), which are joined with a space if multiple are recognized. We expect the Levenshtein distance to be 0 for identical pairs of text and 1 for completely different pairs of text.

Results. Fig. 7.3.3 shows the averaged normalized Levenshtein (edit) distances of all 5 words and only "stop". As shown, the Deepfloyd IF can generate the most accurate text; DALL-E 2 and Stable Diffusion 2 are the second and third while BigSleep is the worst. This order is the same as the order of average detection rates of the object detectors in Table 7.3.2. In particular, the Deepfloyd IF shows a high capability to generate "stop" texts. This

result is consistent with the observation in RQ2 that the stealthiness against humans is significantly improved on the DeepFloyd IF when the robust feature of the text is removed. In summary, the experimental results show that the text generation capability of image generation models has a certain similarity to their natural attack capability. The capability to generate a complex pattern such as the alphabet may correlate with the capability to generate non-robust features that are too subtle to the HVS. We may use this characteristic to design a simple defense against the NDD attack, i.e. we can differentiate the NDD attack by checking if the text "stop" is in it for the stop sign attack. Although these empirical results are still not enough to fully support the edit distance-based method as a metric to measure the natural attack capability, *the metric can be used as a simple sanity check for the image generation models and as a simple defense against NDD attacks on stop signs and other objects with text.*

7.4.4 RQ4: Are non-robust features responsible for the natural attack capability?

This study is motivated by the concept of robust and non-robust features proposed in [206] that the adversarial attack is not a bug but instead caused by non-robust features that are predictive but incomprehensible to humans. In the user study, we have observed that diffusion models introduce some features that are imperceptible to humans but important to the DNN models. While we may call these features non-robust features by definition of the concept, we perform a further evaluation to directly evaluate the effect of non-robust features by following the methodology in [206].

Experimental setup. According to [206], we first create the "robustified" dataset to train robust classifiers. Since our NDDA dataset uses the same labels as the COCO dataset [243], we convert the COCO dataset into a dataset for the multiclass classification task by cropping

Table 7.4.2: Accuracy of the robust and normal classifiers on the stop sign images in the NDDA dataset. The benign means the images generated with the benign prompts; The NDD means the NDD attack that removes all robust features.

	Robustified classifier			Normal classifier		
	Benign	NDD	Diff.	Benign	NDD	Diff.
DALL-E 2	1.00	0.34	0.66	1.00	0.70	0.30
Stable Diffusion 2	0.78	0.58	0.20	0.80	0.66	0.14
DeepFloyd IF	1.00	0.92	0.08	1.00	0.98	0.02
Avg.	0.93	0.61	**0.31**	0.93	0.78	**0.15**

the images within their bounding boxes, randomly selecting 500 images for each class, and "robustify" the images. We train not only the robustified classifier with the dataset but also a normal classifier with the original, non-robust dataset, again with 500 random images per class. ResNet50 [194] is the model architecture for both classifiers.

Results. Table 7.4.2 shows the accuracy of the robust and normal classifiers on the stop sign images in the NDDA dataset. As shown, both the normal and robustified classifiers have higher accuracy when the robust features exist, but the robust classifier's accuracy decreases more than the normal classifier's when all robust features are removed. This means that there is a clear correlation between the sensitivity to the non-robust features and the natural attack capability. We thus consider that *the non-robust features play an important role in enabling the natural attack capability in the diffusion models.* For the other classes, we include the results in Appendix.

7.4.5 RQ5: Is the natural attack capability caused by sharing training datasets?

We evaluate the impact of sharing the training dataset on the natural attack capability. If the evaluating object detectors and diffusion models may share the training dataset, the natural attack capability can be just due to the non-generalizable features in the dataset, i.e., both may just fit into particular noises in the dataset. The large diffusion models, such

Table 7.4.3: Accuracy of the classifiers on the images generated by the DDPM. The rows indicate the splits used to train the DDPM. The columns indicate the splits used for training the classifier.

	MNIST		Fashion MNIST		CIFAR-10	
DDPM\Classifer	Split 1	Split 2	Split 1	Split 2	Split 1	Split 2
Split 1	1.00	1.00	0.98	0.94	0.71	0.64
Split 2	1.00	1.00	0.99	0.93	0.67	0.67
Abs. Diff.	0.00	0.00	0.01	0.01	0.04	0.03

as DALL-E 2 and Stable Diffusion 2, are likely to use famous, publicly available datasets and may use the COCO dataset [243], which the 5 object detectors in Table 7.3.2 use for training. In this RQ, we thus assess the impact of the dataset sharing to see if it can be a major source of the natural attack capability.

Experimental setup. We randomly split the training datasets of MNIST [230], FashionMNIST [348], and CIFAR-10 [226] into 2 splits, respectively. We train simple CNN classifiers and conditional DDPM model [268] for all splits. For each conditional DDPM, we generate 100 images for each class. We then evaluate the difference in the accuracies of the generated images between when the diffusion model and the classifier use the same split and when they use different splits. If the hypothesis is true, these accuracies should have a large gap.

Results. Table 7.4.3 shows the accuracy of each pair of conditional DDPMs and classifiers. The row means which split is used to train the conditional DDPM model; the column means which split is used to train the CNN model. The accuracy is the percentage of the correct classification on the 100 generated images with the corresponding DDPM model. As shown, there is no significant difference between when the diffusion model and the classifier use the same split and when they use different splits. We thus think that *the natural attack capability of diffusion is not due to data sharing but rather due to the non-robust features embedded by diffusion models.*

7.4.6 RQ6: Is the natural attack capability general enough to attack against real-world systems?

To demonstrate the model-agnostic and transferable attack capability of the NDD attack, we evaluate the NDD attacks against the real-world traffic sign detection system. We adopt the threat model of appearing attack [373], which tries to causes a false positive detection of a stop sign.

Experimental setup. We printed 11 images of the NDD attack as in Fig. 7.4.1, showed them to the windshield cameras of the Tesla Model 3, and checked the detection results displayed on the monitor in front of the driver's seat. We select the 11 attack images based on the confidence scores of object detectors in the digital space, especially for YOLOv5, which shows the highest robustness as in Table 7.3.2.

Results. Fig. 7.4.1 shows the successful NDD attacks against Tesla. As shown, 8 of the 11 printed attacks can successfully fool Tesla's commercial traffic sign system as the detected stop signs appear on the driver's monitor. This means that the *NDD attack has 73% of the attack success rate, which is a surprisingly high success rate considering that we do not perform any optimization processes to attack the Tesla.* All attacks are simply generated by diffusion models with simple text prompts. Furthermore, we did not have any special considerations when printing the attacks. We just use a commodity printer, roughly stick papers together with transparent tape, and use normal printing paper. More demos and details are on our project website: https://sites.google.com/view/cav-sec/ndd-attack.

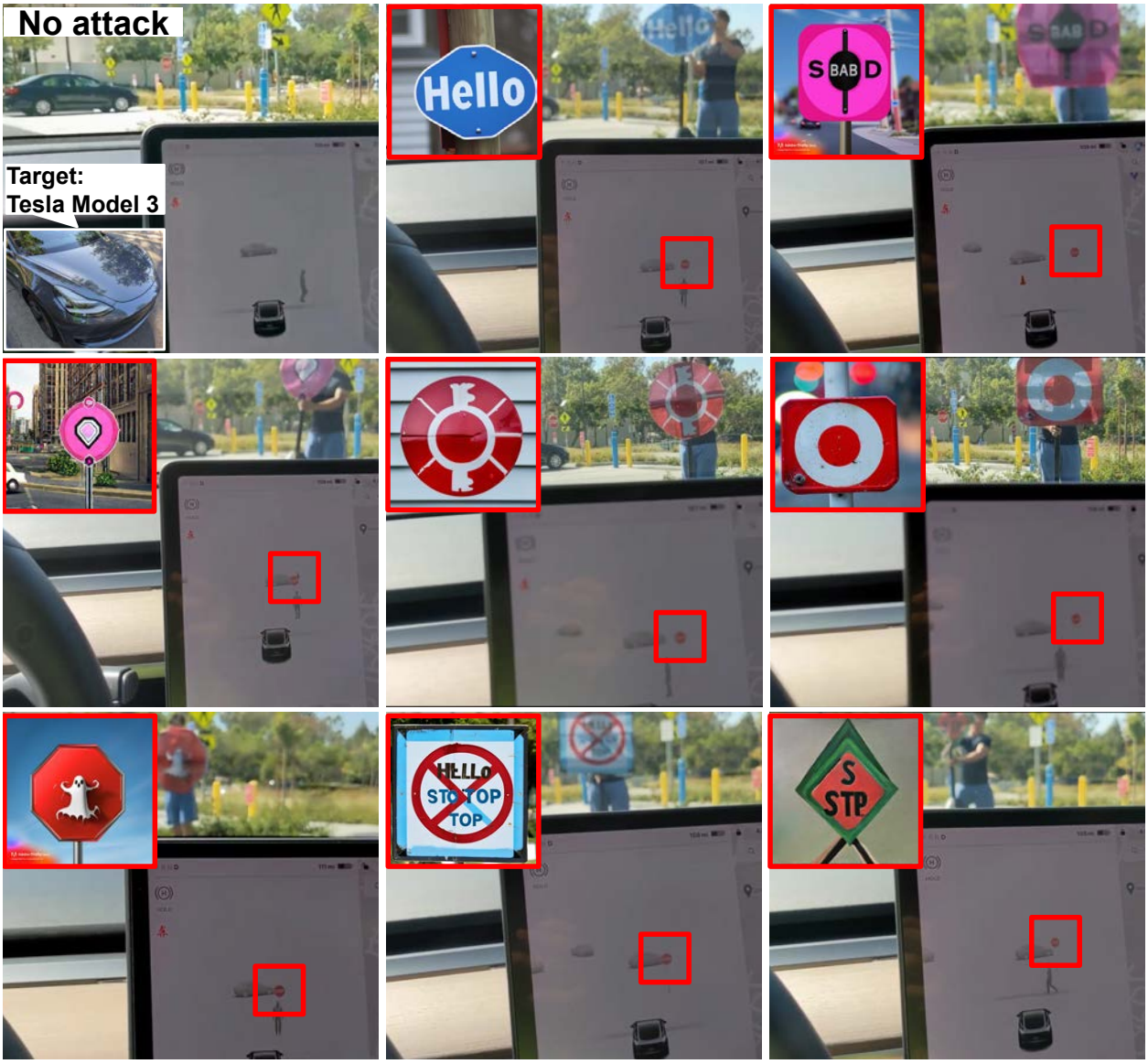


Figure 7.4.1: Successful NDD attacks against Tesla Model 3. We demonstrate that 8 out of 11 printed attacks can successfully fool the commercial traffic sign detection system in Tesla. The left-top images are the original images generated from diffusion models.

7.5 Discussion and Limitation

We discuss the implications of our findings and the limitations of this work, especially the potential negative societal impacts raised by this work.

Safety Implications: We demonstrate the attack effectiveness of the NDD attack against

the Tesla Model 3 but also find that these attacks were not as robust at different distances. Thus, this vulnerability may not pose an immediate threat to fast-driving Tesla or other autonomous vehicles. However, the possibility of affecting a driving vehicle cannot be completely ruled out. We note that the effect of the attack is unlikely to be a coincidence, as these attacks are detected as stop signs even though they do not resemble similar to legitimate stop signs at all. If the attack is reddish and hexagonal, it could be a coincidence. However, it cannot be considered a coincidence that such an attack with blue, purple, green, or non-hexagonal shapes is detected as a stop sign at such a high rate. Furthermore, the current NDD attack does not have any special considerations to attack the Tesla Model 3. As in prior work [112, 138], we may improve the attack by integrating diffusion models into attack generation processes. We hope that our study can facilitate further research to assess the security threat of diffusion models. We have performed a responsible vulnerability disclosure to Tesla before the public release of this work.

Countermeasures: A naive but possible defense for the stop sign attack is the OCR-based attack detection as discussed in RQ3. For the NDD attack removing the text feature, we found that it can achieve 92% true positive rate with 4% false positive rate on the images generated by Deepfloyd IF. However, the true positive rates drop to $\geq$40% if one of the other robust features is removed. Another approach is the "robustified" training [206] as observed in RQ4, but it remains a mitigation measure. So far, no generic defense against adversarial attacks has been reported [109, 132, 331]. Further research efforts are needed in this area.

Definition of Attack Success: As discussed in RQ2, valid adversarial attacks should satisfy the two requirements: effectiveness against DNN models and stealthiness against humans. From this aspect, the stealthiness of all images in the NDDA dataset has not been fully confirmed. However, human annotation for all images did not yield a feasible choice due to the cost, which motivated us to pursue RQ2. We further note that we use state-of-the-art 3 diffusion models, which faithfully follow the text prompt as shown in Fig. 7.3.1.

Root Causes of Natural Attack Capability: Through this study, we empirically explore possible root causes of the natural attack capability and identify potential clues that the natural attack capability is due to the non-robust features injected by diffusion models. However, our empirical study is not sufficient to fully conclude the root causes. For example, the methodology used in RQ4 to extract (non-)robust features proposed by Ilyas et al. [206] is not designed to analyze the natural capability of artificially generated images. Considering the impact on safety-critical applications such as autonomous driving, we are presenting our current best-effort empirical analysis along with our dataset. We hope our study can facilitate further theoretical or more large-scale empirical studies to identify the root causes of the natural attack capability in diffusion models.

Evaluation with More Models and Categories: In this work, we focus on 3 popular diffusion models and 3 object classes with relatively high detection rates to perform deeper analysis on each RQ. Meanwhile, we keep updating the dataset with more diffusion models and object classes for the sake of the dataset's comprehensiveness. The latest version of the NDDA dataset includes 7 diffusion models (Dall-E 2 [154], Dall-E 3 [155], Stable Diffusion 2 [285], Deepfloyd IF [311], Stable Diffusion 1.5 [285], MidJourney [94], and Google Duet [181]) with 15 object classes (stop sign, car, dog, hot dog, traffic light, zebra, fire hydrant, frog, horse, bird, boat, air plane, bicycle, cat, and carrot) to benefit future studies. In the supplementary materials, we evaluate the detection rates of the 15 classes and find that the majority of classes have certain levels of vulnerability against the NDD attack. Current candidates for robust features are chosen to ensure removal (e.g., blue for stop sign); future updates will include more variations for the sake of dataset comprehension on our website. In total, the NDDA dataset has 45,820 images. Each class has $\geq$50 images for the models with API access and $\geq$20 images for other models.

Ethical Considerations: We have gone through the IRB process for the user study. In the experiment on the Tesla Model 3, we ensured that the attacks were not visible to other

Tesla vehicles driving on public roads.

7.6 Conclusion

In this study, we identify a new security threat of the NDD attack that leverages the natural attack capability in the diffusion models. To systematically evaluate the characteristics and root causes of the NDD attack, we conduct a large-scale empirical study using our newly constructed dataset, the NDDA dataset, in which we generate images with state-of-the-art diffusion models while intentionally removing the robust features that are essential to the HVS. We demonstrate that the images without robust features are still detected as the original object and are stealthy to humans. For example, the stop signs with altered text are still detected as stop signs in 88% of cases but are stealthy to 93% of humans. We find that the non-robust features contribute to the natural attack capability. To evaluate the realizability of the NDD attack, we demonstrate the attack against the Tesla Model 3 and confirm that 73% of the NDD attacks are detected as stop signs. Finally, we discuss the implications and limitations of our research. We hope that our study and dataset can help the community to be aware of the risk of the natural attack capability of diffusion models and facilitate further research to develop robust DNN models.

Appendix

7.A Detailed Overview of NDDA Dataset

The latest NDDA dataset consists of the following 15 classes: stop sign, car, dog, hot dog, traffic light, zebra, fire hydrant, frog, horse, bird, boat, air plane, bicycle, cat, and

carrot with 6 diffusion models (Dall-E 2 [154], Dall-E 3 [155], Stable Diffusion 2 [285], Deepfloyd IF [311], Stable Diffusion 1.5 [285], MidJourney [94], and Google Duet [181]). The examples of the images generated by each diffusion model are shown in Fig. 7.G.3. For future versions of the dataset, we plan to generate additional variations of the prompts for each subject to capture a greater variety of NDD attacks. At the time of writing, the following 5 classes have a second variation: stop signs, horses, fire hydrants, cars, and cats. As shown in Fig. 7.A.1, the dataset is organized into 6 "diffusion parent folders" that separate each diffusion model's set of images, which in turn contains multiple folders for each of the 15 object classes from COCO. Each object class folder then contains multiple subfolders that hold the NDD attack images, and these subfolders' names are the text prompts used to generate the set of NDD attack images. For example, if images were generated using Stable Diffusion 2 with the text prompt, "blue dog", the path to this prompt subfolder would be "ndda_dataset/stable_diffusion_2/blue_dog/". For the purposes of submission, we downsample the images to a quarter of their original dimensions and only include 1 image per prompt subfolder.

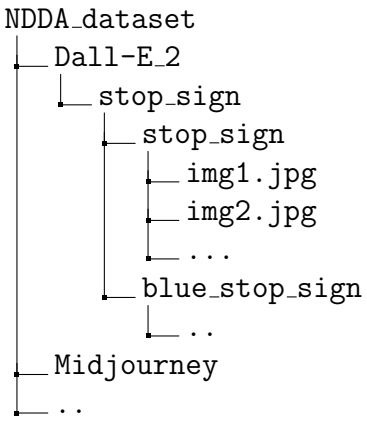


Figure 7.A.1: Overview of the NDDA dataset directory structure, using the Dall-E folder as the example diffusion folder. For each object class folder (stop sign in this case), there are multiple prompt folders that contain ≥ 50 images for models with API access or ≥ 20 images for models w/o API access.

7.B Additional Results of Natural Attack Capability on Object Detectors

Table 7.B.1 and 7.B.2 show the detection results for the fire hydrant and horse classes in the NDDA dataset generated by 3 diffusion models. The results of the stop sign are shown in the main paper. As shown, the majority of the images are still detected as keeping the targeted objects. Even if all robust features are removed, the 3 diffusion models are always able to generate effective attacks for 3 models other than YOLOv3 and YOLOv5.

Table 7.B.1: Detection rates of 5 object detectors on the **fire hydrant** images in the NDDA dataset generated by the 3 diffusion models. **Bold** and underline denote highest and lowest scores in each row.

	Removed Robust Features			Object Detectors					
	Shape	Color	Pattern	YOLOv3	YOLOv5	DETR	Faster	RTMDet	Avg.
DALL-E 2				96%	34%	**100%**	98%	**100%**	86%
	✔			18%	0%	8%	4%	**40%**	14%
		✔		98%	38%	**100%**	96%	**100%**	86%
			✔	58%	4%	88%	84%	**92%**	65%
	✔	✔	✔	0%	0%	4%	2%	**16%**	4%
Stable Diffusion 2				94%	96%	**100%**	**100%**	20%	82%
	✔			6%	0%	14%	**20%**	**20%**	12%
		✔		92%	86%	94%	**100%**	**100%**	94%
			✔	94%	82%	**100%**	98%	98%	94%
	✔	✔	✔	0%	0%	4%	**6%**	4%	3%
Deepfloyd IF				98%	84%	**100%**	**100%**	**100%**	96%
	✔			34%	10%	52%	76%	**86%**	52%
		✔		**98%**	46%	**98%**	**98%**	**98%**	88%
			✔	96%	52%	**100%**	98%	98%	88%
	✔	✔	✔	**72%**	8%	7%	66%	84%	47%

Table 7.B.2: Detection rates of 5 object detectors on the **horse** images in the NDDA dataset generated by the 3 diffusion models. **Bold** and underline denote highest and lowest scores in each row.

	Removed Robust Features			Object Detectors					
	Shape	Color	Pattern	YOLOv3	YOLOv5	DETR	Faster	RTMDet	Avg.
DALL-E 2				76%	48%	78%	84%	**86%**	74%
	✔			90%	60%	92%	**96%**	94%	86%
		✔		**48%**	4%	40%	32%	**48%**	34%
			✔	8%	0%	18%	16%	**24%**	13%
	✔	✔	✔	10%	0%	14%	2%	**18%**	9%
Stable Diffusion 2				86%	60%	90%	88%	**94%**	84%
	✔			**100%**	92%	98%	**100%**	**100%**	98%
		✔		**82%**	18%	72%	54%	68%	59%
			✔	46%	10%	64%	58%	**74%**	50%
	✔	✔	✔	68%	12%	50%	60%	**74%**	53%
Deepfloyd IF				96%	54%	**98%**	94%	100%	88%
	✔			90%	76%	92%	**96%**	**96%**	90%
		✔		82%	32%	72%	68%	**88%**	68%
			✔	60%	2%	64%	62%	**70%**	52%
	✔	✔	✔	**44%**	2%	28%	14%	**44%**	26%

7.B.1 Additional Evaluation with YOLOv8

We evaluate the natural attack capability against YOLOv8 [216], which is one of the current state-of-the-art object detectors. Fig. 7.B.1 illustrates the comparison with YOLOv5 in the stop sign case. As shown, YOLOv8 is generally more vulnerable to the NDD attack than YOLOv5, particularly for the images of DALL-E2. It thus indicates that the current state-of-the-art model still has vulnerability against the NDD attack.

7.B.2 Additional Evaluation on More Class Categories

Fig. 7.B.2 shows box plots of the detection rates of the 15 object categories (stop sign, car, dog, hot dog, traffic light, zebra, fire hydrant, frog, horse, bird, boat, air plane, bicycle, cat,

Figure 7.B.1: Detection rates of YOLOv5 and YOLOv8 on the stop sign images generated by the 3 diffusion models.

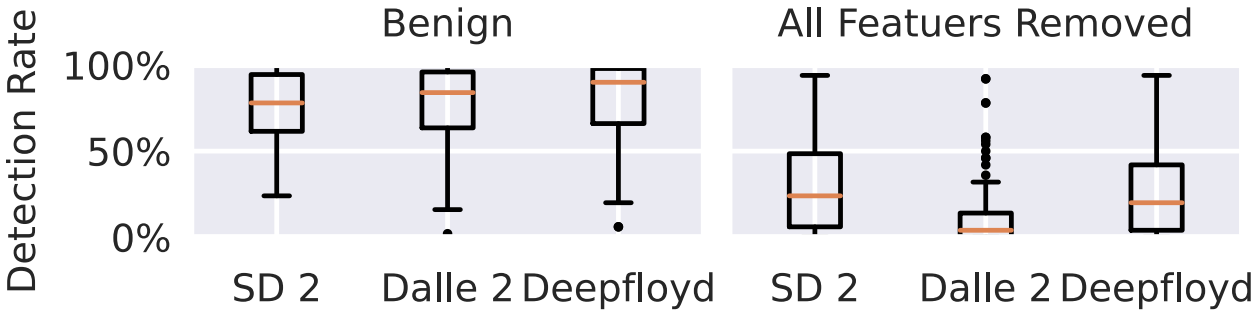


Figure 7.B.2: Box plots of detection rates for 15 object category images generated by the 3 diffusion models.

and carrot). We evaluate 6 object detectors: YOLOv3, YOLOv5, DETR, Faster R-CNN, RTMDet, and YOLOv8. Thus, there are 90 (15×6) data points for each plot. As shown, there are always some levels of vulnerability against the NDD attack because the median values are above zero. Our paper covered the general classes of the object: a stop sign for an artificial sign, a fire hydrant for an artificial object, and a horse for a natural object. We thus leave a deep analysis of other categories for the following works.

7.C Detailed Results of Natural Attack Capability on Image Classification Models

Experimental setup. We train 4 state-of-the-art image classification models, ResNet50 [194], DenseNet121 [203], EfficientNet b0 [319], and ResNeXt [350], on the training datasets derived

from the COCO dataset [243]. We convert the COCO dataset into a dataset for the multiclass classification task in the same way as we did in RQ4: We crop the images in the COCO dataset with their bounding box annotations and randomly select 500 images for each class.

Results. Table 7.C.1, 7.C.2, and 7.C.3 show the classification accuracy for the 3 classes: stop signs, fire hydrants, and horses. As shown, the large number of the generated images are still classified as the targeted object class even though we remove a robust feature. For stop sign, ≥47% of the generated images are still classified as stop signs even though all 4 robust features are removed. For fire hydrant and horse, ≥38% and ≥16% of the generated images are classified as the original class, respectively. In summary, the NDD attack shows quite high attack effectiveness against not only object detectors but also image classifiers.

Table 7.C.1: Classification accuracy of 4 classifiers on the **stop sign** images in the NDDA dataset generated by the 3 diffusion models. **Bold** and underline denote highest and lowest scores in each row.

	Removed Robust Features				Image Classifiers				
	Shape	Color	Text	Pattern	Resnet50	Dense121	Effic.Net	Resnext	Avg.
DALL-E 2					**100%**	**100%**	**100%**	**100%**	100%
	✔				**98%**	**98%**	**98%**	**98%**	98%
		✔			**100%**	**100%**	**100%**	**100%**	100%
			✔		94%	94%	94%	96%	95%
				✔	84%	88%	86%	**90%**	87%
	✔	✔	✔	✔	42%	58%	40%	**64%**	51%
Stable Diffusion 2					72%	**76%**	60%	66%	69%
	✔				50%	56%	40%	**80%**	57%
		✔			**68%**	58%	54%	**68%**	62%
			✔		72%	66%	38%	**76%**	63%
				✔	**58%**	56%	44%	**58%**	54%
	✔	✔	✔	✔	**56%**	44%	34%	54%	47%
Deepfloyd IF					**100%**	96%	**100%**	**100%**	99%
	✔				**96%**	70%	84%	92%	86%
		✔			**100%**	86%	**100%**	**100%**	97%
			✔		92%	86%	74%	**96%**	87%
				✔	88%	86%	78%	**92%**	86%
	✔	✔	✔	✔	**90%**	86%	52%	**90%**	78%

Table 7.C.2: Classification accuracy of 4 classifiers on the **fire hydrant** images in the NDDA dataset generated by the 3 diffusion models. **Bold** and underline denote highest and lowest scores in each row.

	Removed Robust Features			Image Classifiers				
	Shape	Color	Pattern	Resnet50	Dense121	Effic.Net	Resnext	Avg.
DALL-E 2				96%	**98%**	84%	96%	94%
	✔			**94%**	86%	78%	76%	84%
		✔		**94%**	88%	90%	90%	91%
			✔	**76%**	62%	36%	52%	57%
	✔	✔	✔	**66%**	62%	64%	46%	60%
Stable Diffusion 2				**90%**	**90%**	78%	84%	86%
	✔			50%	46%	40%	**52%**	47%
		✔		**94%**	86%	88%	86%	89%
			✔	68%	64%	62%	**72%**	67%
	✔	✔	✔	**46%**	26%	34%	**46%**	38%
Deepfloyd IF				98%	98%	**100%**	**100%**	99%
	✔			**94%**	90%	**94%**	**94%**	93%
		✔		94%	92%	96%	**100%**	96%
			✔	**96%**	88%	82%	90%	89%
	✔	✔	✔	86%	82%	74%	**98%**	85%

7.D Detailed Results of GAN-based Attacks (RQ1)

Fig. 7.D.1 shows the attacks generated by BigSleep that successfully trick object detectors with a confidence score ≥ 0.5. As shown, GAN-based NDD attacks can still generate several successful attacks which have high stealthiness as confirmed by the user study. While the diffusion models have much higher attack effectiveness as discussed in RQ1, this natural attack capability in the GAN model can be also a serious attack threat. Considering such a high attack capability has not been reported even though many GAN-based adversarial attacks [112] are proposed, we think this is mainly due to the high-quality text guide by OpenAI CLIP [276] in BigSleep [101] rather than due to a unique characteristic of the GAN model. However, it is not trivial to investigate the relationship between the impact of non-robust features and the quality of the text guide. We hope that a future study can handle

Table 7.C.3: Classification accuracy of 4 classifiers on the **horse** images in the NDDA dataset generated by the 3 diffusion models. **Bold** and underline denote highest and lowest scores in each row.

	Removed Robust Features			Image Classifiers				
	Shape	Color	Pattern	Resnet50	Dense121	Effic.Net	Resnext	Avg.
DALL-E 2				64%	60%	48%	**66%**	60%
	✔			80%	**84%**	74%	82%	80%
		✔		34%	32%	18%	**38%**	31%
			✔	**18%**	6%	0%	6%	8%
	✔	✔	✔	**26%**	14%	10%	12%	16%
Stable Diffusion 2				**90%**	78%	74%	82%	81%
	✔			92%	**98%**	86%	94%	93%
		✔		**80%**	54%	40%	54%	57%
			✔	**32%**	24%	14%	28%	25%
	✔	✔	✔	46%	**62%**	48%	54%	53%
Deepfloyd IF				**94%**	92%	82%	**94%**	61%
	✔			90%	**92%**	86%	82%	88%
		✔		**94%**	90%	80%	86%	88%
			✔	60%	**64%**	44%	50%	55%
	✔	✔	✔	52%	**62%**	34%	34%	46%

this. So far, the DDA should be more preferable to the attackers because BigSleep's image generation takes much longer ($\geq$20 minutes) than the diffusion model (a few seconds), even though BigSleep needs to generate more images due to its low attack success rate.

7.E Detailed Setup of the User Study (RQ2)

To conduct an ethical user study, we have gone through the IRB process in our institution. The entire questionnaire form of this user study is attached as a part of the supplementary materials. We recruited 82 human subjects on Prolific [56], a crowdsourcing platform specialized for research purposes. All human subjects are English speakers in the United States to follow the IRB instruction. We did not collect any other demographic or privacy-sensitive

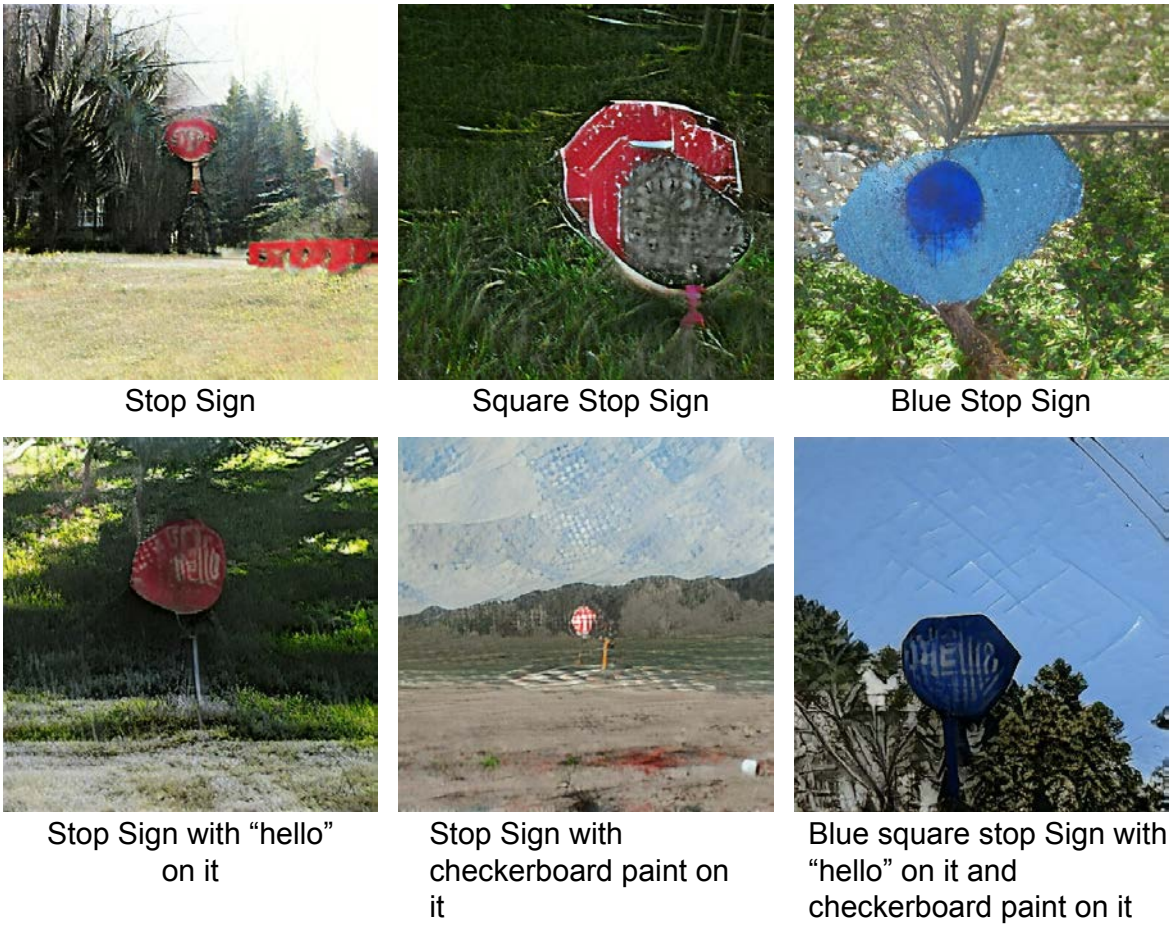


Figure 7.D.1: Examples of stop sign images generated by BigSleep that successfully trick at least one object detector, i.e., a stop sign is detected with a confidence score ≥ 0.5. The user survey includes these images and more.

information from the participants.

We provide 3 types of images: benign, diffusion-generated, and GAN-generated. 3 benign images are provided first in order to have a set of baseline results. Then, we show diffusion-generated images to the users; for every text prompt used, 3 images are generated. A text prompt may either produce a benign image, an image with 1 robust feature removed, or an image with all robust features removed. This step is also repeated using the BigSleep [101] model. Fig. 7.D.1 shows examples of stop sign images generated by BigSleep that successfully trick at least one object detector, i.e., a stop sign is detected with a confidence score $\geq$0.5. The user survey includes these images and more.

7.F Additional Results of the Experiment on Non-Robust Features (RQ4)

Table 7.F.1 and 7.F.2 show the accuracy of the robust and normal classifiers on the fire hydrant and horse object classes, respectively. For the fire hydrant images, the robustified classifier drops the accuracy when the robust features are removed, as observed in RQ4. For the horse images, this analysis does not show meaningful results because the training of robustified classifier fails as the accuracy is <40% even for the benign images. We manually check the robustified horse images and find that the robustified images do not look like legitimate horses. Since this methodology [206] is originally evaluated on the CIFAR-10 [226], it may not be fully compatible with the dataset derived from the COCO dataset. Nevertheless, our finding is generally observed in other cases when the robustified classifiers can achieve similar accuracy to the normal classifier in the benign images.

Table 7.F.1: Accuracy of the robust and normal classifiers on the **fire hydrant** images in the NDDA dataset. The benign means the images generated with the benign prompts; The NDD means the NDD attack that removes all robust features.

	Robustified classifier			Normal classifier		
	Benign	NDD	Diff.	Benign	NDD	Diff.
DALL-E 2	0.66	0.1	0.56	0.94	0.36	0.58
Stable Diffusion 2	0.96	0.42	0.54	0.96	0.38	0.58
DeepFloyd IF	0.82	0.2	0.62	1.00	0.9	0.1
Avg.	0.81	0.24	**0.57**	0.97	0.55	**0.39**

Table 7.F.2: Accuracy of the robust and normal classifiers on the **horse** images in the NDDA dataset. The benign means the images generated with the benign prompts; The NDD means the attack that removes all robust features. Robustified classifier fails to train on the robustified data as its detection rate is ≤40% on the benign images

	Robustified classifier			Normal classifier		
	Benign	NDD	Diff.	Benign	NDD	Diff.
DALL-E 2	0.10	0.06	0.04	0.54	0.04	0.50
Stable Diffusion 2	0.28	0.28	0.00	0.64	0.24	0.40
DeepFloyd IF	0.40	0.28	0.12	0.9	0.28	0.62
Avg.	0.26	0.21	**0.05**	0.69	0.19	**0.51**

7.G Additional Results of Tesla Experiments (RQ6)

Fig. 7.G.1 and 7.G.2 show the stop signs that successfully and unsuccessfully deceived Tesla's vision system respectively. We not only demonstrate that 73% of these generated images are successful but also show 3/4 of the diffusion models exhibit the natural attack capability in the real world. Out of the models that were successful, most of them required a carefully designed text prompt in order to achieve a successful attack, as illustrated by the last 5 images of Fig. 7.G.1 for Adobe Firefly and Stable Diffusion 2. Other diffusion models, such as Google Duet, require less effort; a simple text prompt, "stop sign", is enough to generate images that do not look like stop signs but have enough non-robust features to fool object detectors. We plan to inform this vulnerability to Tesla if this paper is accepted. We hope that our study can facilitate further research to train more robust DNN models.

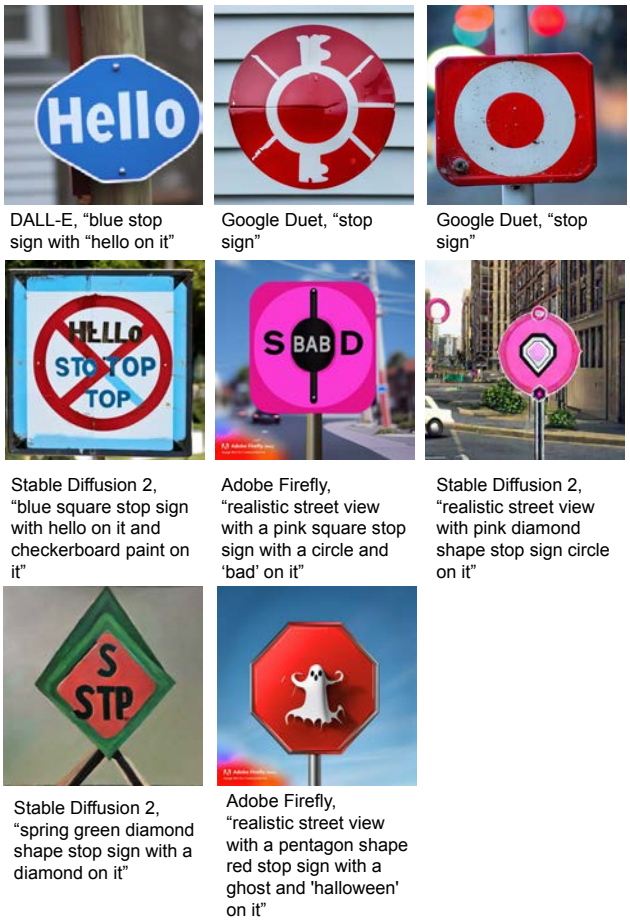


Figure 7.G.1: Successful NDD Attacks on Tesla Model 3. The caption of each image shows the used diffusion model and the text prompt.

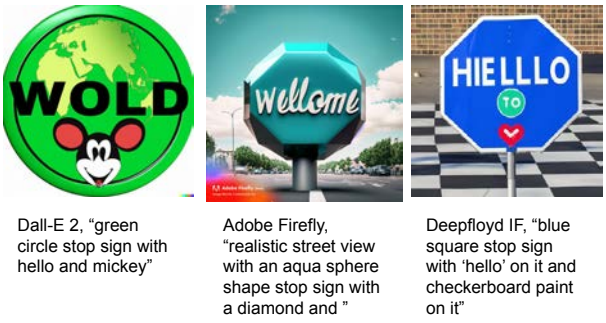


Figure 7.G.2: Unsuccessful NDD Attacks on Tesla Model 3. The caption of each image shows the used diffusion model and the text prompt.

Figure 7.G.3: Overview of the NDDA dataset. 6 popular text-to-image diffusion models are used to generate 15 object classes from the COCO dataset [243]. The left grid consists of benign images while the right grid shows NDD attacked images where diffusion models are instructed to remove all robust features from the image.

Chapter 8

LiDAR Spoofing Meets the New-Gen: Capability Improvements, Broken Assumptions, and New Attack Strategies

8.1 Introduction

LiDAR (Light Detection And Ranging) is one of the most innovative sensors in the past decade. By shooting a laser pulse and measuring its reflection, LiDAR can provide a detailed 3D understanding of the surrounding environment. Autonomous Driving (AD) is one of the most benefited applications of the high-speed and high-precision sensing of LiDARs. After LiDAR showed its effectiveness in the 2007 DARPA Urban Challenge[335], it has been widely recognized as an essential sensor for Level-4 AD and has been adopted in almost all recent robotaxi services (Waymo One [69], Cruise [9]) and AD vehicles operating in the US [26, 28].

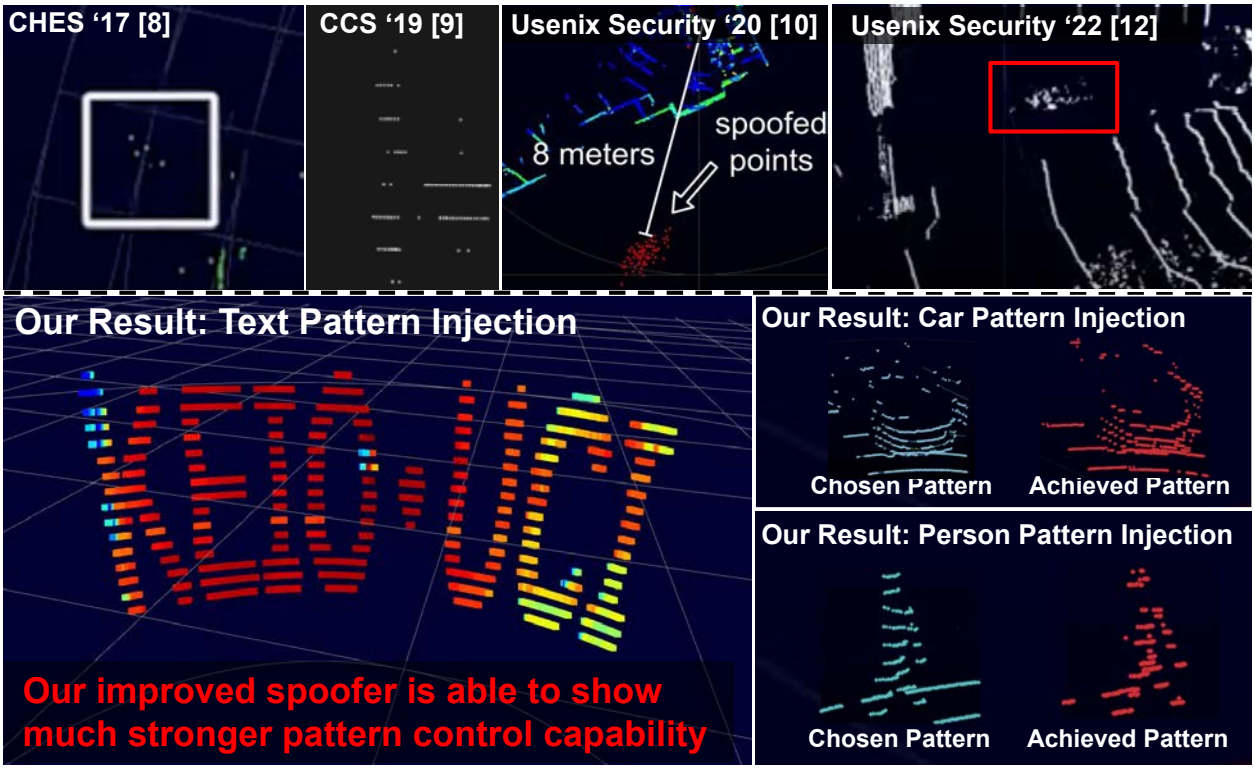


Figure 8.1.1: Demonstration of the Chosen Pattern Injection (CPI) attack capability with >6,000 spoofed points with our improved LiDAR spoofer. This significantly improves the spoofing attack capability from prior works: [306] (∼10 points), [130] (∼20 points) [312] (∼200 points), and [185] (∼200 points).

While highly beneficial to our everyday life and society, AD is also highly security-critical as even a small operational error can cause fatal consequences [27]. To address this, numerous researchers have been conducting security analyses on LiDARs [270, 306, 130, 312, 129, 185, 125] due to their critical role in AD perception. The major security concern of LiDARs is the fundamental vulnerability against malicious laser shooting, or *LiDAR spoofing attacks*. The recent research along this line [312, 190, 185] found that such attacks can cause both false positives (injecting a non-existing fake object) and false negatives (removing an existing object). However, we find that there are 3 critical research gaps in these prior efforts:

Considering only one specific LiDAR: Velodyne VLP-16 [48] has been dominantly used in the prior works since it is viewed as a *de facto* choice for LiDAR spoofing evaluation after the first practical spoofing attack was proposed in 2017 [306]. The following works thus all evaluate their attacks only on VLP-16 [130, 312, 185, 125] or use the attack capability on

VLP-16 to justify the validity of their threat model [190, 191]. Although these results are valid on VLP-16, there is no guarantee that these results are still valid in more recent LiDARs, known as next-generation or new-generation (or *new-gen*) LiDARs [363], as opposed to the first-generation (or *first-gen*) ones such as VLP-16. The new-gen LiDARs have more advanced security-related features, such as laser timing randomization and pulse fingerprinting. Prior works [130, 125, 306] partially discussed some of them as potential defenses, but none of them experimentally studied their impact and effectiveness against LiDAR spoofing.

Assuming unvalidated attack capabilities: So far, all prior works on fake object injection [130, 312, 185] assume a *Chosen Pattern Injection (CPI)* attack capability, i.e., an attacker can spoof a specifically chosen point cloud pattern that was carefully optimized/identified offline beforehand. For example, the Adv-LiDAR attack [130] is a white-box adversarial attack that needs a specific pattern to trigger a vulnerability in deep neural networks (DNNs). The black-box attacks in [312, 185] also require a specific pattern in the shape of a vehicle surface. However, none of them have systematically studied how achievable this is in practice. The top area of Fig. 8.1.1 shows the demonstrated spoofing capabilities so far from prior works [306, 130, 312, 185]. As shown, none of them were able to show strong pattern control capabilities, not to mention to provide a systematic quantification of such a capability to ensure valid security analysis on the object detector side.

Evaluating object detectors with limited spoofing capability modeling and setup diversity: To study the spoofing attack impacts on the object detector side, prior works typically leverage a mathematical modeling of the spoofing capability to achieve high analysis scalability and flexibility [130, 312, 185]. However, the modelings used so far not only fail to correctly capture real-world attack characteristics such as spoofing inaccuracies (detailed in §8.4.1, which can significantly bias the analysis results as shown in §8.4.2), but also lack the coverage of more recent LiDAR features that can significantly change the spoofing capabilities (§8.4.1). Moreover, the object detector model setup is also limited, e.g., no prior

works have evaluated the same model design trained on different training datasets, which we found can significantly change the model robustness analysis results (§8.4.2 and §8.5.2).

To fill these critical research gaps, we conduct the first large-scale measurement study on LiDAR spoofing attack capabilities against object detection. Specifically, our study is driven by 3 key novel research questions (RQ):

RQ1: *Are the common design-level assumptions made in prior LiDAR spoofing attacks actually realizable? If so, can they also hold for the more recent new-gen LiDARs?*

RQ2: *Do different types of LiDARs, especially the new-gen ones with security-related features, have different vulnerability characteristics to LiDAR spoofing attacks?*

RQ3: *Does the vulnerability status of popular object detectors to spoofing attacks significantly change due to the new-gen LiDAR features and different model setups?*

To comprehensively address these RQs, we devote significant engineering efforts to reproduce and/or improve the current state-of-the-art attacks: For RQ1, to properly explore the potential to achieve the CPI attack capability, we improve the spoofing device on their optical and electronics parts (§8.3.4), which enables us to be the first to demonstrate and quantify the CPI attack capability, which is commonly assumed but never clearly demonstrated in prior works [130, 312, 125, 185, 190] and shown in Fig. 8.1.1. For RQ2, to adequately measure the vulnerability status of new-gen LiDARs, we explore *asynchronized* (§8.2.3) spoofing attacks since synchronized ones are directly foiled by their common security-related features such as timing randomization and pulse fingerprinting (§8.4.1). However, since all existing works only consider first-gen LiDARs, their designs predominately focus on synchronized attacks [306, 130, 312, 185, 125], leaving the asynchronized attack design space under-explored. For RQ3, to enable large-scale evaluation of LiDAR spoofing attack capabilities against different object detectors, we perform novel mathematical modeling of the spoofing attack capabilities on different types of LiDARs based on our measurements for RQ1 and RQ2, which is not only the first to perform such modelling for first- and new-gen LiDARs, but

also the first to model for object removal attacks.

In the measurement study, we cover major types of LiDAR spoofing attacks against (1) 9 popular LiDAR models in total, covering both the classic first-gen ones (e.g., VLP-16) and the new-gen ones with new security-related features; and (2) 3 major types of object detectors trained on 5 different datasets. To the best of our knowledge, this is the first large-scale measurement study on this topic in terms of *LiDAR numbers* (none of the prior works studied over 1 LiDAR model, while we study 9), *LiDAR types* (all prior works concentrate on first-gen ones, and we are the first to study the new-gen ones), and also *training datasets* (no prior works studied the same model design with over 1 dataset, while we study 5).

Through this study, we are able to identify a total of *15 novel findings*, which include not only completely new ones due to the measurement angle novelty (e.g., measurements on new-gen LiDARs), but also many that directly challenge the latest understandings in this problem space. For example:

- We find that with more careful spoofer optics and electronics implementations, an attacker actually does not really need to exploit object detector-level vulnerabilities to achieve a near-front road object injection, which is a common assumption made in prior LiDAR spoofing works [130, 312, 185];
- We find that VLP-16 is actually the *only* LiDAR model for which the CPI attack capability assumption is feasible, which is another key design assumption made in latest prior works [130, 312, 185].
- We find that the new-gen LiDAR features previously expected to be capable of largely mitigating or even preventing spoofing (e.g., timing randomization [130, 125], pulse fingerprinting [306]) may not be effective as expected (§8.4.1, §8.4.2);
- We find that the latest synchronized object removal attack can no longer be applied to the

new-gen LiDARs, but there exist asynchronized object removal attacks that can overcome such a limitation and can lead to a similar level of practical attack capabilities (§8.5).

In summary, our study has the following contributions:

- We conduct the first large-scale measurement study on LiDAR spoofing attack capabilities on object detectors with 9 LiDARs, covering both first- and new-generation ones, and 3 major types of object detectors. We are also the first to investigate the impacts of new security-related features in more recent LiDARs from the security perspective.
- To facilitate the measurements, we not only identify spoofer improvements that significantly improve the latest spoofing capability, but also identify a new asynchronized object removal attack that overcomes the applicability limitation of the latest method to new-gen LiDARs while having a similar level of practical attack capability.
- To facilitate the object detector-level measurements, we perform novel mathematical modeling of the spoofing capabilities for both object injection and removal attacks based on our measurement results, which is the first to model for both first- and new-gen LiDARs and also to model for object removal attacks.
- Our study is able to uncover a total of 15 novel findings, including not only completely new ones due to the measurement angle novelty, but also many that can directly challenge the latest understandings in this problem space.

Data/code release. All data, code, and hardware designs (e.g., for the new spoofer) are released at **https://sites.google.com/view/cav-sec/new-gen-lidar-sec**.

8.2 Background and Related Works

8.2.1 LiDAR Basics and Recent Trend

LiDAR is an active sensor that can obtain high-resolution 3D point cloud data of the surrounding environment. The most common type of LiDAR today is direct time-of-flight (dToF) LiDAR, which fires laser pulses to the environment and uses their reflections to actively measure (or "scan") the surrounding objects. Specifically, for each laser pulse, it uses the time difference between the firing and reflection to obtain the 3D position of a "point" on the object surface. By performing laser firing to different horizontal angles (*azimuth*) and vertical angles (*altitude*), the point measurements thus form a "point cloud" (Fig. 8.1.1). Such LiDARs typically use laser peak power (e.g., ∼10 W) significantly higher than sunlight even after a long flight, which allows them to detect objects more than 200 meters away even under high ambient lights.

Recent Trend: New-Gen LiDARs. Early Velodyne LiDARs such as VLP-16 [48] and VLP-32c [46] are commonly known as the first-generation (*first-gen*) LiDARs [363], which naively integrate the classic point-wise laser ranging systems. The advent of first-gen LiDARs has significantly boosted critical real-world applications such as autonomous driving (AD), but its complex mechanical design increases costs and limits scalability. To overcome the limitations, the new-generation (*new-gen*) LiDARs [363] mount all components, such as the photodetector and the readout circuitry, on a single chip (called a system-on-chip (SoC) approach). This not only reduces the cost and improves the scalability of the system, but also allows new designs of laser firing and receiving. For example, microelectromechanical systems (MEMS) LiDAR [337] move a mirror to scan a wide azimuth range instead of mechanical rotation. Flash LiDARs [286, 33] fire a broad laser that covers the entire field of view (FOV) and calculates the distance by compensating its weak return laser by accumulation. Moreover, the SoC approach allows more complex signal processing, such as a large number

of simultaneous laser firings [10], laser timing randomization [10, 37], and fingerprinting [49], to be robust against challenging environments, e.g., multiple LiDARs operating very closely.

All in all, the recent new-gen LiDARs substantially improved the electronics and the scanning mechanisms even though they still have the same basic design: laser firing and receiving. However, none of the prior works on LiDAR spoofing attacks have studied the security property of such new-gen LiDARs; *all* of them are performed on only the first-gen rotational ones, with a predominate focus on in fact *only 1 specific LiDAR model*: VLP-16 [306, 130, 312, 185, 125]. The major design differences between first- and new-gen LiDARs are likely to cause significant differences in their security characteristics, which thus motivates this study.

8.2.2 Object Detection on Point Cloud

In real-world applications such as autonomous driving (AD), object detection is no doubt one of the most critical roles for LiDAR. As in other computer vision areas, 3D object detection on point cloud is substantially benefited by the recent progress of DNN. However, traditional DNN architectures such as CNN cannot be directly applied due to the irregular and unordered structure of point clouds [239]. To handle such data complexity, 3 major types of 3D object detection methods are widely adopted: *voxel-based*, *point-based*, and *point voxel-based* methods [273]. More details are in Appendix 8.E.

8.2.3 LiDAR Spoofing Attacks against Object Detection

Due to the basic sensing mechanism of laser firing and receiving, LiDARs are fundamentally vulnerable to malicious laser shooting, or *LiDAR spoofing attacks*. Specifically, such attacks work by using an external attack device ("spoofer") to fire laser pulses back to the victim LiDAR in order to manipulate the time measurements of the laser-receiving events and thus

Table 8.2.1. Taxonomy of existing LiDAR spoofing attacks against 3D object detection. Sync. and Async. refer to synchronized and asynchronized spoofing techniques (§8.2.3).
* Cannot inject fake objects (1) closer than the spoofer itself, and (2) in a different angle towards the victim than the spoofer itself.
† Only show removal of a 41×42 cm^2 metal plate in short duration (4 sec)

Spoofing Technique	Object Detection Model-Level Attack Effect	
	Object Injection	Object Removal
Sync. (White-box)	Adv-LiDAR [130], Occlusion [312], Frustum [185]	PRA [125], ORA [270]
Async. (Black-box)	Relay* [270]	Saturating† [306], HFR (§8.3.3)

the corresponding 3D position measurements (§8.2.1). Such point position manipulations can thus be strategically used to achieve attack effects on the 3D object detector side, more specifically (1) *Object injection*, e.g., by moving the positions of a set of points to form a 3D pattern to cause a false-positive detection of a non-existing object; and (2) *Object removal*, e.g., by moving the points originally on an object to elsewhere to cause false-negative detection of such an object.

Attack Taxonomy

Table 8.2.1 shows a taxonomy of existing LiDAR spoofing attacks on object detection based on the LiDAR spoofing technique (specifically, the requirement of *synchronization*) and the targeted object detector-level attack effect. *Synchronization* means to synchronize the malicious laser firing timing with the victim LiDAR scanning (i.e., laser firing) timing, which can enable precise control of the attack laser-receiving timing and thus the corresponding positions of the spoofed points. Fig. 8.2.1 illustrates the synchronized and asynchronized spoofing techniques with the latest attack capabilities. As shown, to achieve synchronization, the attacker needs to use an extra device (photodetector, or PD, in Fig. 8.2.1) to first learn the current state of the victim LiDAR scanning in the real time, and then use the victim LiDAR's scanning pattern to derive future victim laser-firing timings for synchronized attack laser-firing. More detailed explanations are in Appendix 8.B. This process requires precise

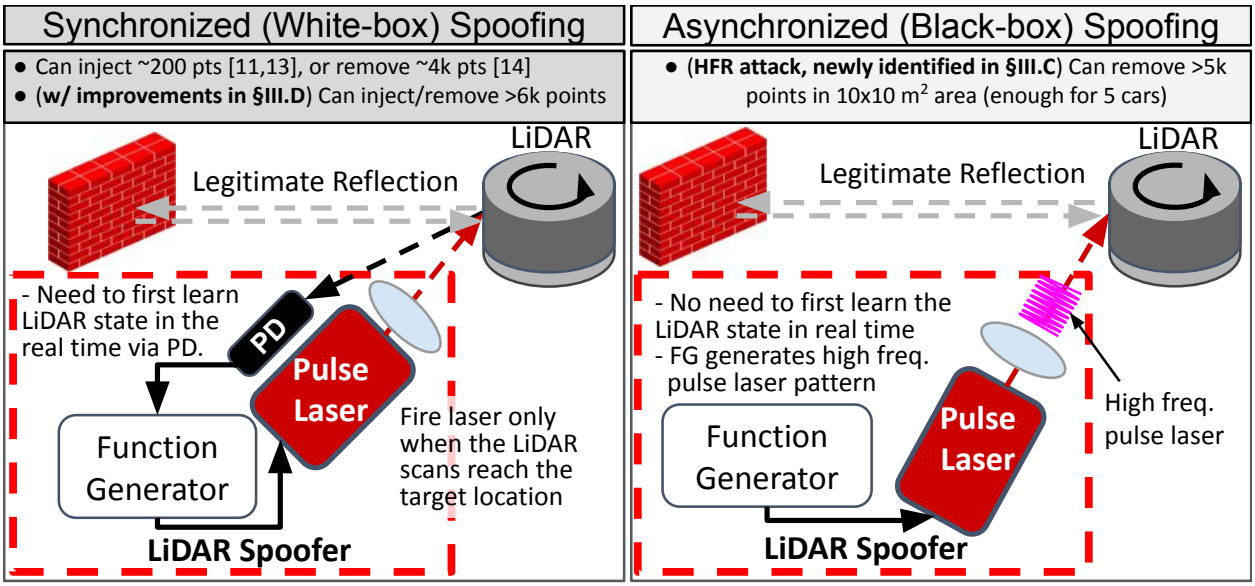


Figure 8.2.1: Illustration of the synchronized and asynchronized LiDAR spoofing techniques with the latest attack capabilities. Synchronized spoofing needs white-box knowledge of the victim LiDAR scanning patterns and an extra device (Photodetector, or PD) for synchronization (§8.2.3), while asynchronized spoofing does not need these (i.e., black-box attack).

and predictable knowledge of the victim LiDAR's scanning pattern beforehand, which thus can be viewed as a *white-box* LiDAR attack. Those that do not assume synchronization (i.e., *asynchronized* attacks) can be applied without knowing anything about the LiDAR internal scanning logic, which can be viewed as *black-box* attacks.

Spoofing Attacks for Object Injection. The very first LiDAR spoofing attack was discovered in 2015 [270], which found that shooting lasers to a victim LiDAR can inject ≥200 points and achieve fake object injection at the object detector side. This attack works by relaying the laser received from the victim LiDAR, which thus does not require synchronization. However, this also makes it impossible to inject objects (1) in closer positions than the spoofer itself, and (2) in a different angle towards the victim than the spoofer. These limitations make it not very useful in real-world attack scenarios since at the time of spoofing, the attackers would hope that the most threatening object to the victim vehicle is the induced fake objects instead of the attacker herself [306].

To address such limitations, synchronized spoofing [306] is proposed to use the *synchroniza-*

tion process described above and also illustrated in Fig. 8.2.1. The early-stage attack designs can inject only 10 spoofed points [306], but the following works progressively increase the number of spoofed points to 60 [130] and 200 [312], and show that the 60 and 200 spoofed points are enough to cause fake object injection on latest object detectors. After the success, the attack capability of injecting 200 points under the Chosen-Pattern Injection (CPI) assumption (detailed later in §8.2.3) becomes the *de facto* threat model in the following works [190, 185, 191]. However, none of them have (1) demonstrated and quantified the CPI attack capability, and (2) studied their attack effect on new-gen LiDARs, which have new features such as laser timing randomization and pulse fingerprinting that can directly challenge their basic design assumption — the requirement of synchronization. In this paper, we fill both of these critical research gaps.

Spoofing Attacks for Object Removal. With the success to achieve object injection, more recent works start to explore using LiDAR spoofing for object removal. Specifically, saturating attack [306] is the first to show the object removal effect. In this attack, instead of firing pulsed lasers like the prior works above, it fires a strong *continuous* laser to the victim LiDAR to indirectly cause measurement errors of laser-receiving events. However, due to the requirement of maintaining continuous high-power laser, it is physically difficult for the attack laser beam to (1) achieve a large receiver area coverage at the victim LiDAR side with high intensity, and (2) maintain the attack effect for a long time. For example, the demonstrated object removal effect is only about removing a 41×42 cm^2 metal plate, lasting $<$4 seconds, which thus makes it highly limited in real-world attack scenarios such as when attacking AD. For example, the saturating attack (0.8 W average power) needs roughly 10 times higher average power than the HFR attack (80 W peak power) (§8.3.3) to achieve similar attack effectiveness, but such a powerful diode is not publically available so far.

To address such limitations, similar to the object injection side, more recent works start to leverage synchronized spoofing techniques. Specifically, the ORA attack (object removal

attack) [190] found that using the synchronized spoofing capability of injecting 200 points under the CPI assumption as mentioned above, attackers can fool 3D object detectors by strategically injecting spoofed points inside the target object's bounding box, since the point cloud of legitimate objects should have points mostly on the object surface instead of inside. However, such removal effect still depends on the object detector-side vulnerabilities. Most recently, the physical removal attack (PRA) [125] found that the object removal effect can be more generally achieved by moving all points on a victim object to within the minimum operational threshold (MOT) of the victim LiDAR, which is common filtering mechanism to automatically discard points below a certain distance. This filtering is implemented in most LiDARs because the LiDAR measurement at a very close distance is generally inaccurate since the ToF can be too short to distinguish from errors due to inadequate calibration. This work shows the capability of removing ∼4k points and thus can remove critical road objects such as pedestrians, which is thus the current state-of-the-art in object removal attack capability.

However, similar to the object injection attack side, none of these prior works have considered the new-gen LiDARs that may come with features that can break some of their fundamental design assumptions. In this paper, we find that (1) PRA can no longer be applied to new-gen LiDARs due to the requirement of synchronization (§8.5.1), and (2) the assumption of non-zero MOT for PRA may not necessarily hold (e.g., for XT32 [49], §8.4.1). We further identify a new object removal attack adapted from the saturation attack, which does not require synchronization but can still achieve practical object removal effects (can remove >5k points in 10×10 m^2 area, which is enough for 5 cars as shown later in §8.5.2).

Chosen-Pattern Injection (CPI) Attack Capability

So far, all prior works using synchronized spoofing for object injection explicitly or implicitly assume that at the actual attack time the attacker is able to accurately inject a specific

spoofed point cloud pattern chosen by the attacker beforehand, e.g., via various kinds of offline optimization/identification processes [130, 312, 185]. In this paper, we thus explicitly define this as the *Chosen Pattern Injection (CPI)* attack capability.

Such an attack capability is theoretically achievable since the synchronized LiDAR spoofing technique should by design be capable of precisely controlling the position of each spoofed point. However, so far no prior works have systematically studied how achievable this is in practice, not to mention to provide a systematic quantification of the pattern control capability, which is necessary for achieving valid security analysis on the object detector side. In this paper, we thus perform the first measurement study to fill this critical gap, with the coverage of both first- and new-gen LiDARs.

8.2.4 Threat Model

We follow the same threat model as in prior works [130, 312, 185], i.e., the attacker fires malicious lasers from their spoofer to the victim LiDAR (§8.2.3, Fig. 8.2.1). As described in [185], the spoofer device can be at a front vehicle, vehicle in the next lane, or a roadside in AD scenarios.

8.3 Measurement Study Setup

8.3.1 Targeted LiDARs

Table 8.3.1 summarizes the LiDARs involved in this measurement study. As shown, we are able to cover 9 popular LiDARs in total, including 3 first-gen ones, 6 new-gen ones, an operating range from 9 m to 300 m, 3 scanning types (Rotating, MEMS, and Flash), and 3 security-related features (simultaneous laser firing, timing randomization, and fingerprint-

Table 8.3.1. The 9 LiDARs involved in our measurement study. 1stG and NewG: First- and new-generation. FOV: Field of View.

		Velodyne			Leddar	Ouster	Intel	Livox	Hesai	Robosense
										
		VLP-16 [48]	VLP-32c [46]	VLS-128 [4]	Pixell [18]	OS1-32 [10]	Realsense L515 [15]	Horizon [22]	XT32 [49]	Helios 5515 [37]
	Gen. (year)	1stG (2016)	1stG (2017)	1stG (2017)	NewG (2019)	NewG (2019)	NewG (2019)	NewG (2020)	NewG (2020)	NewG (2021)
	Scanning Type	Rotating	Rotating	Rotating	Flash	Rotating	MEMS	MEMS	Rotating	Rotating
	Wavelength	905 nm	905 nm	905 nm	905 nm	865 nm	860 nm	905 nm	905 nm	905 nm
General Specs	Vertical FOV	30°	40°	40°	16°	45°	55°	25.1°	31°	70°
	Horizontal FOV	360°	360°	360°	180°	360°	70°	81.7°	360°	360°
	Max. Range [m]	100	200	300	56	120	9	260	120	150
	Min. Range [m]	1	1	0.5	0.1	0.3	0.25	0.5	0	0.2
	Vertical Channel	16	32	128	8	32	-	-	32	32
Security	Simul. Firing	1	2	8	3	32	1	1	1	1
	Timing Random.				✔	✔	✔	✔		✔
	Fingerprinting								✔	

ing). This makes this work the largest-scale measurement study on this topic so far in terms of both LiDAR numbers (none of the prior works studied over 1 LiDAR model, while we study 9) and LiDAR types (all prior works concentrate on first-gen ones, and we are the first to study the new-gen ones).

8.3.2 Targeted Object Detection Models and Datasets

Our study includes 5 popular DNN-based 3D object detectors in total covering all the 3 major model types (§8.2.2): voxel-based (PointPillars [229], SECOND [355], and Part-A^2 [305]), point-based (3DSSD [357]), and point voxel-based (PV-RCNN [303]). In order to study the impact of the training dataset, we also prepare 5 different versions of PointPillars [229] trained on 5 different datasets (KITTI [175], Waymo [313], nuScenes [124], Lyft [221] datasets), and the dataset used in Apollo 6.0 [7]. This thus makes our measurement also the largest in terms of the coverage of object detection model architectures and training datasets. We directly use pre-training ones from MMdetection3D [150], OpenPCDet [84], or the Apollo repository [7].

8.3.3 Targeted LiDAR Spoofing Attacks

Our study targets spoofing attacks with the latest attack capabilities on both the object injection and removal sides (§8.2.3). Specifically, for object injection, we reproduce the latest synchronized spoofing technique [130, 312, 125], and for object removal, we reproduce the latest physical removal attack (PRA) [125]. In addition, since the latest removal attack cannot be applied when synchronization is not possible (which we found to be the case for new-gen LiDARs in general, detailed in §8.4.1), we further identify and include in our measurement a new asynchronized spoofing technique for object removal, which is adapted from the saturation attack (§8.2.3) but with much more practical object removal capabilities:

High-Frequency Removal (HFR): New Asynchronized Spoofing Technique for Object Removal. As described in §8.2.3, due to the requirement of maintaining continuous high-power laser, the saturation attack is fundamentally limited in the removal area size and duration (e.g., 41×42 cm^2 in <4 seconds [306]), making it highly limited in real-world attack scenarios. Later works address this by using synchronization, but this also makes them inapplicable to new-gen LiDARs (§8.4.1). To address this, we identify a new adaptation of the saturation attack, which still does not require synchronization, but instead of using high-power continuous laser, it uses *high-frequency pulsed laser*. This new attack is thus called *high-frequency removal (HFR) attack*. The key idea is to fire a large number of attack laser pulses to the victim LiDAR at a frequency that is higher than the laser-firing frequency of the victim LiDAR. This allows the attack laser to hit every laser-firing event of the victim LiDAR in the scanning range hit by the spoofer, which can thus achieve the spoofing effect for every points in that range without any knowledge of the victim scanning pattern (i.e., without requiring synchronization). Fig. 8.2.1 illustrates the spoofing mechanism, and Fig. 8.A.2 in Appendix illustrates the difference to the saturation attack. Due to the lack of synchronization, the receiving timing of the attack laser is random, and thus the spoofing effect will be moving each legitimate surface point of the targeted objects to a random

position or undetectable area of the victim LiDAR (e.g., within MOT). As shown later in §8.5.2, this can completely destruct the point cloud patterns at the original object position and thus cause the object removal effects.

8.3.4 LiDAR Spoofer Setup

Fig. 8.3.1 shows an overview of the LiDAR spoofer, its optics design, and the setups of the indoor and outdoor experiments to support our measurement study of the targeted spoofing attacks (§8.3.3). We generally follow the setup adopted in the latest works [130, 312, 185, 125], so we leave the details to Appendix 8.C.

Spoofer Improvements. When reproducing the prior works, we also implemented several improvements on top of the latest spoofer setup to more properly study the spoofing attack capabilities. For instance, we find that inadequate optical design can significantly affect the number and angle coverage of spoofable points due to undesired diffusion and convergence of the attack laser beam. Specifically, if the laser beam is expanding, the intensity rapidly decays with distance, which thus limits the number of spoofable points; if the laser beam is converging, the diameter of the beam decreases with the distance, which thus makes it difficult to aim and cover a wide receiver angle. Ideally, the laser beam should be collimated without diffusion and convergence to achieve the target LiDAR with a minimum loss. To achieve this, we use a 25.4 mm focal length plano-convex lens with 1 inch (25.4 mm) diameter to cover the laser emitted by SPL90_3 [12], which has a maximum beam divergence angle of 25°. Since the diameter of the laser at the focal point is $\tan 25^\circ \times 25.4 \times 2 = 23.7$ mm, we can cover it with the 1 inch (25.4 mm) lens. To precisely calibrate the lens setup, we develop a device that can use a hollow screw to adjust the distance between the laser diode (LD in Fig. 8.3.1) and the lens. Technically, this allows the spoofer to maintain a stably large number and angle coverage of spoofable points from hundreds of meters away.

Besides the optics side, we also notice that inadequate electronic implementations in the spoofer setup can introduce inaccurate laser detection timing (from LiDAR to PD in Fig. 8.3.1) and also long and unstable delays in the signal processing of the spoofer devices, which can thus significantly affect the attack capability in precisely controlling the placement of a spoofed point at a chosen 3D position. To address this, we improve the amplifier for the PD to increase the laser detection accuracy, and also improve the function generator (FG) setup to allow more precise nanosecond-level configuration and calibration. As shown in Fig. 8.1.1 and detailed later in §8.4.1, these improvements significantly increase the latest spoofing attack capabilities in terms of the spoofed point number, angle coverage, pattern control, and robustness over distance. Following the best practices advocated in [300], we release the raw hardware design file (CAD file) and the bill of materials on our website [32] to make it easier for future works to reproduce and build upon (most recent works did not do this [130, 312, 125]).

8.4 Object Injection Attack Measurements

With the measurement setup above, in this paper we perform the first large-scale measurement study for both classes of state-of-the-art LiDAR spoofing attacks: object injection attack and object removal attack, which will be the focus of this and the next section respectively. For each attack class, we first perform the attack capability measurements at the LiDAR point cloud level, and then model such attack capabilities for the subsequent measurements at the object detector level.

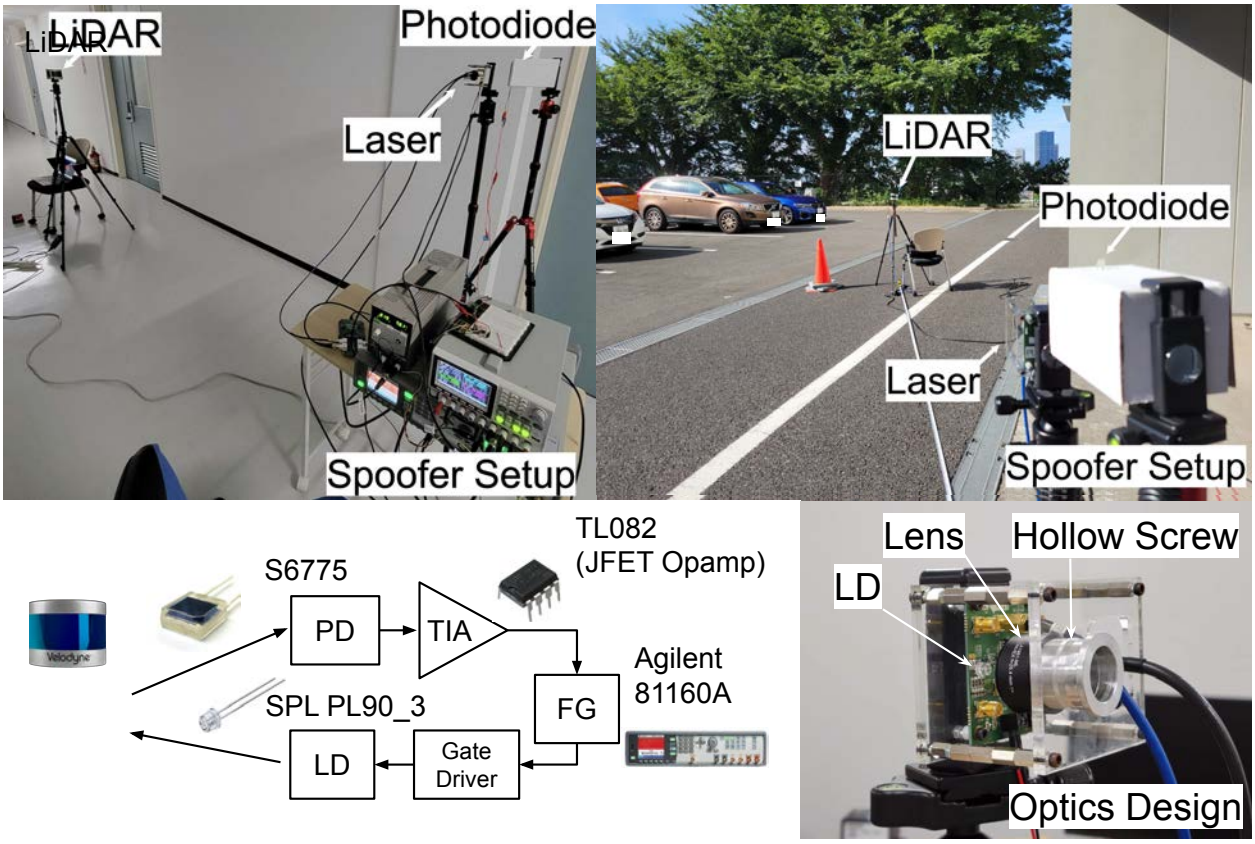


Figure 8.3.1: Overview of our LiDAR spoofer setup, the optics design, and the setup of the indoor and outdoor experiments. PD: Photodetector. TIA: Transimpedance amplifier. FG: Function generator. LD: Laser diode. More details in Appendix 8.C.

8.4.1 LiDAR-Level Measurements (RQ1, RQ2)

In this section, we measure the object injection attack capabilities at the LiDAR point cloud level, which is measured by the spoofing attack's capability in injecting spoofed points. We perform this measurement using an attack laser with sufficiently high intensity so that the attack-induced spoofed points can be easily differentiated from the benign ones. To answer RQ1, we first re-visit the existing state-of-the-art spoofing attacks for point injection on the VLP-16 LiDAR [48], which is predominantly used as the *only* evaluation target in prior works [306, 130, 312, 125, 185]. After that, we then conduct the measurement study on new-gen LiDARs to answer RQ2.

Re-Visiting Design Assumptions Made in Prior Works with VLP-16 (RQ1)

Prior works have shown that the number of spoofed points in VLP-16 increases as the quality of the spoofer device improves, ranging from 60 [130] to ∼4k [125]. As shown in Table 8.4.1, with our spoofer improvements (§8.3.4), such attack capabilities are further improved significantly, which can now inject >6,131 points indoors and >6,514 points outdoors. This is at least 50% more than those in the latest prior work [125], which capped at ∼4k. We noticed a concurrent work that may have also made contributions in improving the spoofing capability [213]. The full paper is not available at the time of our writing, but from the abstract it seems that (1) the spoofing capability is still capped at 4200 points, which is still at least 50% fewer than ours; and (2) the study scale is much smaller than ours in terms of LiDAR number (only 2, v.s. 9 for us) and likely also LiDAR types and generation coverage.

Meanwhile, we also observe that the number of spoofed points and angles in the outdoor setup is generally larger than those in the indoor setup. We consider this reasonable since the legitimate laser reflections are fewer in the outdoor environment (e.g., no wall reflections), leaving more room for spoofing. However, this actually contradicts those reported results in the latest prior work [125]: as shown in Table 8.4.1, the spoofed points outdoors are significantly fewer than those indoors (1.8k versus 4k) and they attribute this to the lighting condition differences (e.g., they also report that the number of spoofed points decreases from ∼2,500 at night to ∼1,600 at day.) We consider that our results are achieved by our more careful optical setup as described in §8.3.4 since the spoofing laser intensity is much stronger than the sunlight and should not be decayed in a short 10 m flight. We suspect that the optical setup of the previous work is not well calibrated and the laser beam diverges.

Finding 1 (RQ1): *With electronic and optical setups, the LiDAR spoofing attack can inject >6,000 points ranging >80°, which is a significantly higher number of points and wider range than in previous studies. Furthermore, previous observations that spoofing distance and lighting conditions can greatly affect the LiDAR spoofing capability [125] no*

longer hold if implemented using more careful optics.

CPI Attack Capability Measurements

As shown in Fig. 8.1.1, our spoofing improvements achieve strong spoofed point pattern control capability for the first time. Considering that the CPI attack capability is a common design-level assumption made in prior works (§8.2.3), we thus perform the first systematic quantification of it to allow more rigorous attack capability analysis on the object detector side. Note that some prior works tried to include a modeling of such CPI attack capability in their object detector-side analysis [185], but their modeling is based on their intuitions (e.g., simply add vertical and horizontal noises [185]) instead of the LiDAR sensing mechanisms and the spoofer designs. As shown later in §8.4.2, such erroneous spoofing accuracy modeling causes significant differences in the object detector-level attack evaluation.

Based on our attack reproduction and spoofing inaccuracy cause investigations, we find that there are two types of errors in controlling the position control of each spoofed point x_{ij} at i-th altitude and j-th azimuth (illustrated in Fig 8.4.2): (1) *inner-frame error* $\delta_{ij}^{\text{inner}}$: the inaccuracy in spoofing a point at a chosen 3D position within an individual frame; and (2) *inter-frame error* δ^{inter}: the inaccuracy in maintaining the spoofed position of the same point in the chosen pattern across consecutive frames. Rooted in the spoofing design (Fig. 8.3.1), the inner-frame error is due to the inaccurate signal bursting of FG to the gate driver, while the inter-frame one is due to inaccurate timing of triggering the FG from TIA. With our experimental setup, we measure both and find that $\delta_{ij}^{\text{inner}}$ and δ^{inter} are $\sim$10 cm as in Fig 8.4.1 and $\sim$35 cm respectively.

Finding 2 (RQ1): *The CPI attack capability, which is commonly assumed but never demonstrated in prior works [130, 312, 125, 185, 190], can be achievable in practice with a well-calibrated spoofer, while by design such pattern control capability is subject to inner- and inter-frame errors.*

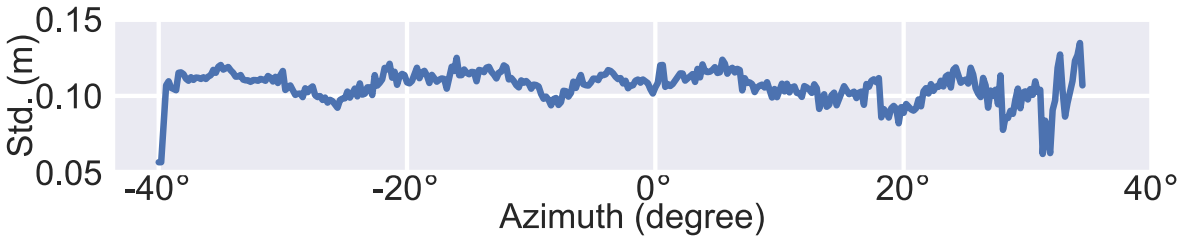


Figure 8.4.1: Standard deviations of inner-frame error on VLP-16.

Assumption that object detector-level vulnerability exploitation is needed: Combining these spoofing capability advances above (significantly improved spoofable points, angle, and CPI attack capability), this actually means that *an attacker does not always need to exploit object detector-level vulnerabilities to achieve a near-front road object injection*, which is actually the major design assumption for all attack works so far on the object injection attack side [130, 312, 185]. Specifically, as clearly stated in prior works [130, 312], a valid point cloud for a front-near vehicle needs ∼2,000 points and 15° azimuth (θ) coverage, but their spoofers can only achieve <200 points and <8°, which is why object detector-level vulnerability exploitation (and thus those novel spoofing pattern optimization/identification) are needed. These results indicate that the exploitation of the object detector-level vulnerability is not the necessary condition although it may help to design more advanced attacks.

Finding 3 (RQ1): *Now with the CPI attack capability demonstrated with sufficiently larger spoofable point number and angle, an attacker actually has sufficient capabilities to directly inject a near-front vehicle pattern. This finding has significant implications for the current line of research, since this (1) may not need to exploit object detector-level vulnerabilities on the attack sides [130, 312, 185]; and (2) may suggest that model-level defenses that try to leverage the inability of spoofers in directly injecting indistinguishable patterns as benign cases (e.g., [312]) can be ineffective.*

Table 8.4.1. Evaluation results of the synchronized injection attack on VLP-16. $\mathcal{N}$: Number of injected points by spoofing. θ: Azimuthal range of the point injection. $\mathcal{R}$: Point injection success rate within θ. The distance d is between the spoofer and the LiDAR. Number in parenthesis: $\mathcal{N}$ from latest prior work [125]. "-": Data not available.

	Indoor			Outdoor (Daytime: 70 lux)		
d	$\mathcal{N}$	$\mathcal{R}$	θ	$\mathcal{N}$	$\mathcal{R}$	θ
2 m	6,523 (-)	98.5%	82.7°	7,705 (-)	94.9%	100.5°
(2.5 m)	(<4k)			(-)		
4 m	6,386 (-)	96.9%	82.5°	7,950 (<1.8k)	96.9%	101.5°
6 m	6,575 (-)	98.6%	83.4°	7,357 (<1.5k)	87.2%	99.6°
8 m	6,213 (-)	93.8%	82.8°	6,702 (<1k)	97.7%	83.4°
10 m	6,131 (-)	93.2%	82.1°	6,514 (<1k)	93.3%	84.2°

Measurements on New-Gen LiDARs

Table 8.4.2 shows the measurement results of the point injection capabilities on not only first-gen LiDARs such as VLP-16 but also new-gen ones. The first thing we find is that the latest spoofing attack for object injection, synchronized spoofing (§8.2.3), can only be applied to the first-gen ones; for the new-gen ones, since they all have either timing randomization or pulse fingerprinting, synchronization (and thus CPI) becomes virtually impossible since timing randomization by design prevents the prediction of the future laser firing timing from the victim LiDAR, while pulse fingerprinting by design prevents the prediction of the future laser pulse pattern of each measurement. To still measure their vulnerabilities to spoofing attacks, we try to inject as many spoofed points as possible with a random attack with high-frequency pulses. The random attack is similar to the newly-identified HFR attack (§8.3.3), but the frequency is tuned to achieve the largest number of injected points.

As shown, compared to the first-gen LiDARs, the security-related features in the new-gen LiDARs result in huge differences in the point injection capabilities: >19k injected points for Horizon, ∼3.2k for Helios, 100-350 on Realsense L515 and XT32, and only 28 on OS1-32. We will closely investigate such differences in the next section. Despite these variances, one observation is in common: VLP-16 is actually the *only* LiDAR among all these 7 ones that can achieve the CPI attack capability — it is not possible by design for the 6 new-gen ones due to timing randomization and pulse fingerprinting, while it is also not achievable to the

Table 8.4.2. Measurements of the point injection capabilities on different LiDARs. For the first-gen LiDARs, the attack is synchronized spoofing (§8.2.3). For the new-gen ones, we inject as many points as possible with random firing (1 MHz). Symbols are the same as in Table 8.4.1.

	First-Gen		New-Gen				
			w/ Timing Randomization				w/ Fingerprint
	VLP-16	VLP-32c	OS1-32	Helios	Horizon	L515	XT32
$\mathcal{N}$	6,523	9,711	28	3,203	19,182	321	113
$\mathcal{R}$	98.50%	82.90%	43.80%	19.4%	79.90%	0.1%	2.10%
θ	82.7°	73.2°	0.72°	34.2°	103.4°	81.7°	70°

other first-gen LiDAR (VLP-32c) due to simultaneous firing (§8.4.1).

Finding 4 (RQ1): *VLP-16 is actually the only LiDAR for which the CPI attack capability is feasible, which is the key design assumption made in all prior works on object injection attack side [130, 312, 185]. This directly challenges the validity of all these existing designs against the more general and recent set of LiDARs.*

Impacts from Security-Related Features

Simultaneous Laser Firing. As listed in Table 8.3.1, many LiDARs fire multiple laser pulses simultaneously. Velodyne LiDAR is likely adopting a modular approach that doubles units when increasing altitudes. Hence, the number of simultaneous laser firings is also doubled: VLP-16 [48] fires 1 laser during each measurement, and VLS-32 [46] fires 2 lasers simultaneously. This design makes the CPI attack infeasible because we cannot selectively return the laser to each simultaneous laser due to the large diameter of the spoofing laser. For example, on VLP-32c we can always inject spoofed points pair by pair, and the injected pair will always have the same distance to LiDAR. Although simultaneous laser firing may help attackers since it can affect more points by a single attack laser pulse, the CPI becomes no longer feasible.

Timing Randomization. New-gen LiDARs typically have a feature that randomizes their laser shooting timing at each firing to mitigate interference from other LiDARs as listed in Table 8.3.1. The timing randomization makes the CPI attack capability virtually impossi-

Table 8.4.3. Distribution of laser firing intervals for LiDAR with timing randomization. Std. σ: Standard deviation of the error caused by the timing randomization in meters. Max. Δ: Maximum error caused by the timing randomization in meters.

	OS1-32 [10]	Horizon [22]	L515 [15]	Pixell [18]	Helios [37]
					
Dist. [μs]	$\mathcal{U}_{1.4,1.8}$	$\mathcal{U}_{4.0,4.3}$	$\mathcal{N}_{51,0.025}$	$\mathcal{U}_{4.5,5.8}$	$\mathcal{N}_{1.6,0.005}$
Std. σ	33.3 m	26.0 m	7.5 m	110.4 m	1.5 m
Max. Δ	57.7 m	45.0 m	20.1 m	191.3 m	5.3 m

$\mathcal{U}_{\text{min,max}}$ - Uniform distribution, $\mathcal{N}_{\text{mean,std}}$ - Gaussian distribution

ble because the attacker can no longer synchronize with the laser firing pattern. However, interestingly, we find that the attacker may still be able to inject some points if the randomization is not strong enough. To quantify the impacts of the timing randomization, we measure the laser firing interval (not available from data sheets) with a PD and an oscilloscope. Table 8.4.3 illustrates the histogram, standard deviation, and maximum value of laser firing intervals of each LiDAR with timing randomization. We also categorize and fit the observed firing intervals into the uniform or Gaussian distribution based on the shape of its histogram. We convert time difference to distance with the following formula: $\frac{\Delta t \times c}{2}$, where Δt is the timing difference and c is the speed of light. Fig. 8.A.3 in Appendix shows the point clouds of Horizon [22], which can illustrate the impacts of timing randomization on spoofing capability. As shown, when under attack, the whole point cloud becomes highly randomized, and the object shapes (e.g., our lab room as shown in the benign case) completely disappear. We further evaluate the impact of the errors on object detectors in §8.4.2.

Pulse Fingerprinting. We identify that Hesai XT32 [49] can foil synchronization (and CPI attack capability) even without timing randomization due to its pulse fingerprinting. While we cannot be entirely sure due to the lack of official documentation, XT32 likely encodes its fingerprinting in the interval of the pair: As shown in Fig. 8.4.3, the XT32 pulse always forms a pair of pulses (two closely-connected spikes) corresponding to a single distance measurement. However, we also find that the fingerprinting itself cannot perfectly defend against spoofing attacks because random spoofing can sometimes coincide with the fingerprinting interval, which can lead to up to 113 spoofed points as in Table 8.4.2. This could

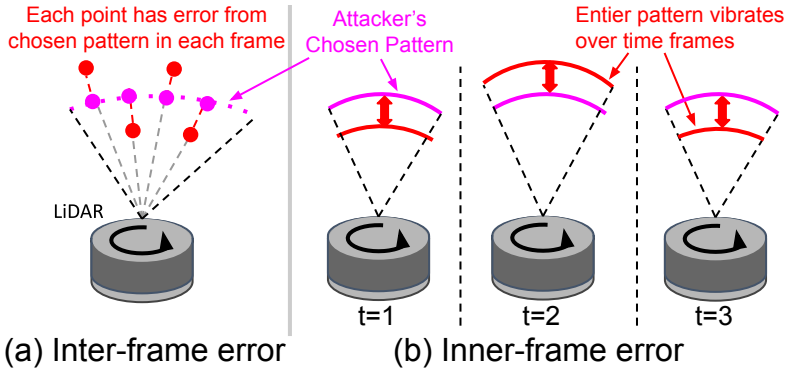


Figure 8.4.2: Illustration of inner- and inter-frame errors. Inner-frame error causes spoofing inaccuracy along with the ray direction within a frame. Inter-frame error causes the entire pattern to vibrate across consecutive frames.

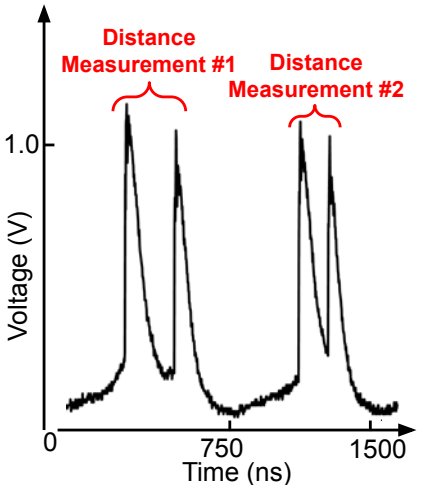


Figure 8.4.3: Examples of the receiving pulse shape of XT32 [49].

be because such pulse fingerprinting is originally developed for anti-interference purposes (e.g., to allow multiple LiDARs to operate at close range) instead of security and thus does not have enough randomness/entropy. Furthermore, it is not trivial to design more complex fingerprinting while ensuring eye safety as we will discuss in §8.6. This means that if in each attack laser-firing event the attacker also fires a pair of pulses with the interval that can most likely coincide the fingerprinting interval (e.g., the one we found that can spoof 113 points), there can still be a random subset (e.g., 113) of the points in the attacker-chosen point cloud pattern that can be spoofed with the CPI attack capability. Thus, later in §8.4.2, we model this effect on the spoofing capability as a random downsampling from a chosen pattern.

Finding 5 (RQ2): *The current pulse fingerprinting is not complex enough to perfectly prevent spoofing attacks, likely because it is currently designed only for anti-interference purposes instead of security. However, it is not trivial to design a complex fingerprinting while ensuring eye safety.*

Case Study on Specific LiDARs

Using our measurement setup, we also found 5 uniquely-interesting characteristics of specific LiDARs: the use of rare wavelength in OS1-32, relay attack on Leddar Pixell, zero-distance sensing of XT32, extra-wide vertical FOV of Helios 5515, and simultaneous laser firing on OS1-32 and VLS-128, which led to 2 new findings that are different from existing understandings in the community. Details are in Appendix 8.D.

8.4.2 Object Detector-Level Measurements (RQ3)

Modeling of the Spoofing Attack Capability in Point Injection

Similar to prior works in this problem space [130, 312, 185], to enable large-scale measurements of the attack capabilities on the object detector side, we need to mathematically model the point injection capabilities. Prior efforts in such modeling did not systematically consider (1) the modeling of the CPI attack capability (§8.2.3), and (2) common new-gen LiDAR features that can fundamentally affect the object injection attack capabilities such as timing randomization and pulse fingerprinting. Leveraging this measurement study, we can thus develop a new modeling that addresses both fronts:

$$\mathcal{P}_I(x_{ij}) = x_{ij} + (\delta_{ij}^{\text{rand}} + \delta_{ij}^{\text{inner}} + \delta^{\text{inter}}) \cdot g(x_{ij}),\ x_{ij} \in \mathcal{C}_n \subset \mathcal{C} \tag{8.1}$$

, where $\mathcal{C}$ is a point cloud (i.e., chosen pattern) that the attacker originally wants to inject (e.g., the point cloud of a vehicle); $\mathcal{C}_n$ is a point cloud randomly downsampled to n points from $\mathcal{C}$ to model the impact from pulse fingerprinting (§8.4.1); $x_{ij} \in \mathbb{R}^3$ is a point injected by the attack at i-th altitude and j-th azimuth; $g : \mathbb{R}^3 \to \mathbb{R}^3$ is a function to obtain a movable

unit direction of point x. Due to the physics of LiDAR, each point can only move along with the laser direction. As the LiDAR typically locates at the origin of the point cloud, $g(x_{ij})$ can be written as $\frac{x}{||x||_2}$ in this case. To model the spoofing inaccuracy and the effect of the timing randomization, we add 3 types of errors: $\delta_{ij}^{\text{rand}}$, $\delta_{ij}^{\text{inner}}$, and δ^{inter}. Randomization error $\delta_{ij}^{\text{rand}}$ is the error introduced by the timing randomization. It follows a certain distribution based on the target LiDAR as measured in Table 8.4.3. We use $\delta_{ij}^{\text{rand}} \sim \mathcal{N}(0, \sigma)$ for the Gaussian distribution and $\delta_{ij}^{\text{rand}} \sim \mathcal{U}(-\frac{\max - \min}{2}, \frac{\max - \min}{2})$ for the uniform distribution. $\delta_{ij}^{\text{inner}}$ and δ^{inter} are the inner-frame and inter-frame errors, respectively. Based on the measurements in §8.4.1, we sample $\delta_{ij}^{\text{inner}}$ from $\mathcal{N}(0, 10 \text{ cm})$ and sample δ^{inter} from $\mathcal{N}(0, 35 \text{ cm})$. Note that $\delta^{\text{inter}} \in \mathbb{R}$ does not depend on each point x_{ij}, i.e., one scalar value is sampled per case and applied to all points in the case. To the best of our knowledge, this is the first modeling of the point injection capability that can cover both first- and new-gen LiDAR features.

Note that this modeling does not cover the impacts from the simultaneous laser firing feature (§8.4.1). This is because although such a feature can make the CPI attack capability infeasible, (1) it cannot prevent the point injection itself; (2) details of the simultaneous firing pattern are not well documented; and (3) the more recent LiDARs (e.g., the ones after 2019 in Table 8.3.1) do not have such a feature. We thus leave its modeling to future work.

Experimental Setups

We perform the experiments based on a scenario in the KITTI dataset. Fig. 8.A.4 in Appendix depicts the generated scenario: we place a 3D vehicle object in front of the victim and move the corresponding points to the object's surface, following the same methodology used in [312, 129]. We generate 15 scenarios varying the distance between the victim to the vehicle object from 0 m to 14 m (0 m means the victim's nose touches the vehicle object's tail). Fig. 8.A.1 in Appendix shows the number of points on the injected vehicle object. In addition to the original point clouds, we evaluate point clouds downsampled from the

original one to 10, 50, 100, and 200 points (i.e., $n = 10, 50, 100$, and 200 in Eq. 8.1) as a modeling of the fingerprinting effect (§8.4.1). We judge the object injection as successful if there exists a detected object overlapping with the ground truth area of the injected object, i.e., the IoU between the ground truth and detected object is more than zero.

Results

Impact of Error Modeling on Spoofing Accuracy: We measure the impacts of 3 types of different modeling: without errors, our modeling with inner- and inter-frame errors (§8.4.1), and the error modeling used in the Frustum attack [185], which is the latest modeling effort in prior works and simply adds Gaussian errors along with the vehicle's Cartesian coordinates (forward, left, and up) following ($\mathcal{N}(1, 0.1)$, $\mathcal{N}(0, 0.5)$, $\mathcal{N}(1, 0.2)$) in meters, respectively. Fig. 8.4.4 shows the attack success rates against our targeted models with different architectures and different training datasets (§8.3.2). As shown, the different error modeling can generally cause significant differences in the attack success rates, e.g., the results without errors and with the naive modeling in [185] can differ 96% and 70% respectively on average to those using our modeling. Specifically, the latest modeling effort in [185] significantly over-estimates the errors along their forward and up coordinates, which causes the model-level success rates to be generally lower than those with our more systematic modeling based on actual experiments (§8.4.1).

Finding 6 (RQ1): *Error modeling on the spoofing accuracy can significantly affect the object detector-level attack results. The latest prior efforts [185] largely overestimate the errors as compared to our systematically modeled and quantified ones via experiments.*

Impact of Pulse Fingerprinting: Fig. 8.4.5 shows the attack success rates under different down-sampling levels n as a modelling of different pulse fingerprinting levels (§8.4.1). To perform control experiments of the fingerprinting impact factor, we do not apply inter- and inner-frame errors in this experiment. Generally, fingerprinting with higher complexity (i.e.,

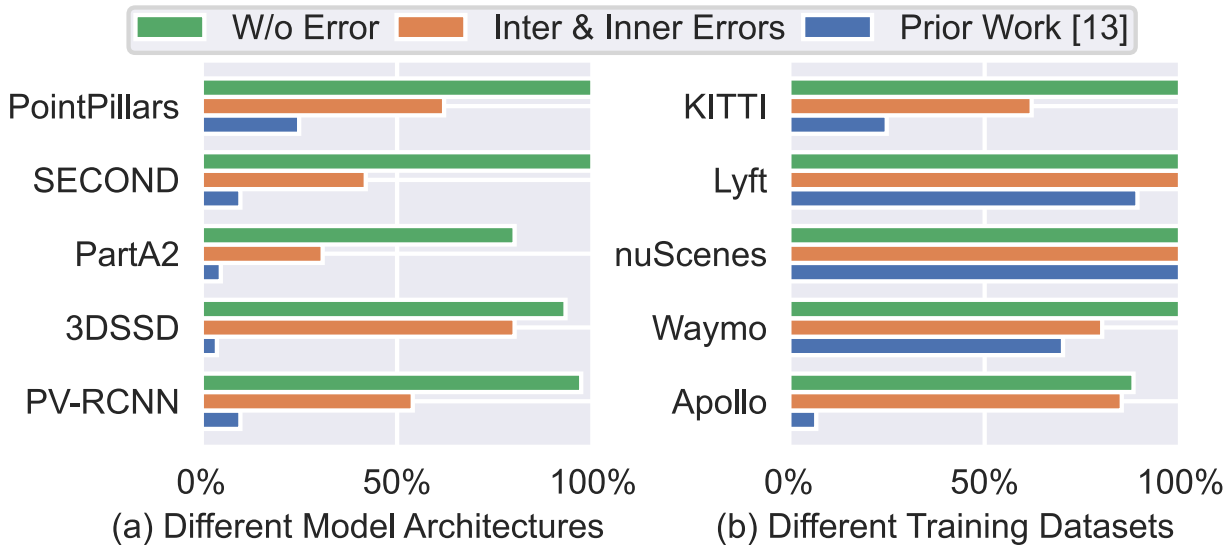


Figure 8.4.4: Object injection attack success rates under 3 different types of noises on (a) models trained on the KITTI dataset and (b) PointPillars trained on different datasets.

lower n) shows higher defense capability. However, when $n = 100$, which can represent the fingerprinting complexity level today (XT32 as measured in §8.4.1), the defense effectiveness is actually still very minimal (reduce the attack success rate only by 3% on average). The defense effectiveness only becomes significant when n reaches as low as 10, while such higher effectiveness is still model architecture and training dataset dependent.

Finding 7 (RQ3): *If with enough complexity, pulse fingerprinting can indeed show high defense capability to object injection attacks (although can be object detector model architecture and training dataset dependent). Unfortunately, the complexity of pulse fingerprinting today may not be enough to achieve such high defense capability.*

Impact of Timing Randomization: Table 8.4.4 and 8.4.5 show the attack success rates under different timing randomization levels based on our measurements in §8.4.1. As shown, timing randomization can in general significantly reduce the attack success rates, e.g., the average attack success rates with randomization are dramatically reduced to at most 34% for most models, while those without randomization are at least 88%. We also measured the attack success rates with both timing randomization and pulse fingerprinting, with the fingerprinting level set at the observed level today ($n = 100$, based on XT32 measurements

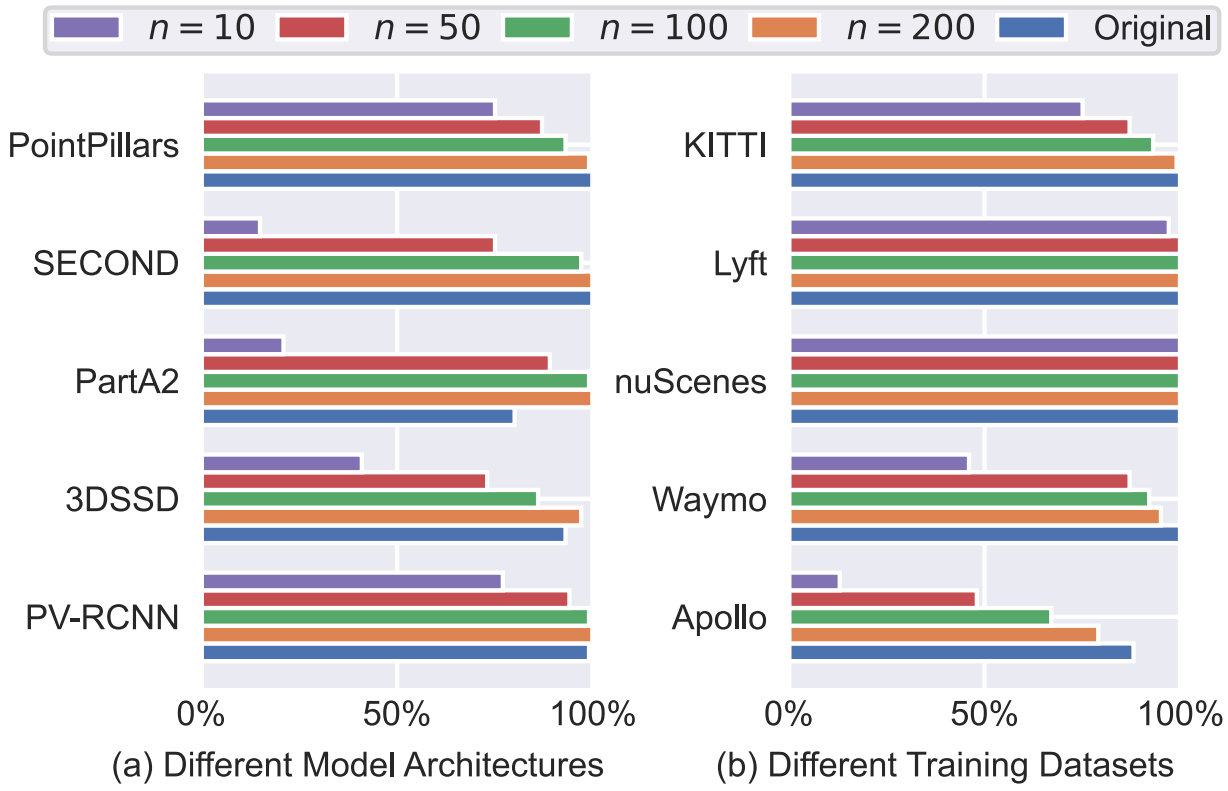


Figure 8.4.5: Object injection attack success rates under different down-sampling levels n on (a) models trained on the KITTI dataset and (b) PointPillars trained on different datasets.

in §8.4.1). However, consistent with Finding 7, we are not able to observe significant changes in the attack success rate reduction.

Finding 8 (RQ3): *Timing randomization, even with those with low randomization entropy that is realized for anti-interference purposes instead of security today, can have significant defense capabilities against object injection attacks. Today's fingerprinting complexity level again may not be able to significantly help boost the defense capabilities when used together with timing randomization.*

Meanwhile, we further notice that the defense effectiveness with timing randomization can differ significantly when applied to models trained on different datasets. For example, as shown in Table 8.4.5, the attack success rates are reduced by only 10% and even 0% on average across all randomization levels for the models trained on Lyft and nuScenes respectively, while the success rates can be reduced by as high as 84% and 88.6% for the same model

Table 8.4.4. Object injection attack success rates under different randomization levels based on our measurements in §8.4.1 on *different model architectures*. ∅: No randomization. Avg.: The average results across all these randomization levels.

LiDAR	Rand. model [m]	PointPillars	SECOND	PartA^2	3DSSD	PV-RCNN
VLP-16	∅	100%	100%	80%	93%	97%
Helios	$\mathcal{N}(0, 1.5)$	2%	54%	41%	7%	24%
L515	$\mathcal{N}(0, 7.5)$	**0%**	24%	14%	7%	**0%**
Horizon	$\mathcal{U}(-45, 45)$	39%	35%	21%	30%	17%
OS1-32	$\mathcal{U}(-58, 58)$	47%	38%	21%	28%	23%
Pixell	$\mathcal{U}(-191, 191)$	60%	21%	20%	8%	43%
	Avg.	30%	34%	23%	16%	21%
With fingerprinting effect $n = 100$:						
	Avg.	38%	20%	16%	18%	41%

Table 8.4.5. Object injection attack success rates under different randomization levels based on our measurements in §8.4.1 on *models trained on different datasets*. ∅: No randomization. Avg.: The average results across all these randomization levels.

LiDAR	Rand. model [m]	KITTI	Lyft	nuScenes	Waymo	Apollo
VLP-16	∅	100%	100%	100%	100%	88%
Helios	$\mathcal{N}(0, 1.5)$	2%	100%	100%	26%	48%
L515	$\mathcal{N}(0, 7.5)$	**0%**	60%	100%	**0%**	**0%**
Horizon	$\mathcal{U}(-45, 45)$	39%	96%	100%	7%	**0%**
OS1-32	$\mathcal{U}(-58, 58)$	47%	99%	100%	12%	**0%**
Pixell	$\mathcal{U}(-191, 191)$	60%	96%	100%	36%	1%
	Avg.	30%	90%	100%	16%	10%
With fingerprinting effect $n = 100$:						
	Avg.	38%	85%	92%	33%	5%

design trained on Waymo dataset and the private dataset from Apollo. In particular, the models trained on Lyft and nuScenes are particularly vulnerable to object injection attacks since as shown in Fig. 8.4.5, these models can detect the vehicle even when the vehicle point cloud is randomly downsampled to as few as only 10 points. This suggests that the training dataset choice can play a significant role in determining the model vulnerability to spoofing attacks. Later in §8.5.2, we will continue investigating this aspect together with the object removal attack measurement results.

8.5 Object Removal Attack Measurements

After the measurements on object injection attacks, in this section we measure the other important class of LiDAR spoofing attacks: object removal attacks (§8.2.3).

8.5.1 LiDAR-Level Measurements (RQ1, RQ2)

Table 8.5.1 shows the LiDAR-level attack capability measurement results for the PRA attack and the newly-identified HFR attack (§8.3.3) on different LiDARs. To count the removed points, starting from the attacked point cloud, we first identify the attack-induced spoofed points using the same method as in §8.4.1 and remove them. Next, we subtract the remaining points in the attacked point cloud from the benign point cloud; now the remaining points in the benign point cloud are thus the removed benign points by the attack (detailed in Appendix 8.G). We select 1 MHz as the pulse frequency and 80V as the laser drive voltage for HFR (detailed in Appendix 8.F).

As shown in Table 8.5.1, with our spoofer improvements in §8.3.4, PRA can now achieve over 6,600 removed points on VLP-16, which is 65.5% more than the original paper [125]. However, such strong attack capability is limited to the first-gen LiDARs; for the new-gen ones, PRA is not applicable to any of them since its requirement of synchronization is directly foiled by common next-gen LiDAR features such as timing randomization and pulse fingerprinting (§8.4.1).

Finding 9 (RQ1): *Due to the requirement of synchronization, the basic design assumption of the latest object removal attack is generally broken for next-gen LiDARs due to common features such as timing randomization and pulse fingerprinting.*

Since there is no need of synchronization, as shown in Table 8.5.1, the newly-identified HFR attack can be generally applied to all LiDARs in the table, no matter if the targeted LiDAR is first- or new-gen. Specifically, for the first-gen ones, HFR is able to achieve comparable point removal capability to PRA, e.g., removing >5,300 points in >85°. Fig. 8.5.1 shows an example of the HFR attack effect on VLP-32c. As shown, the point cloud patterns of a person and the majority of the room wall are completely removed, with only some points with random noise patterns left. Compared to PRA, the point removal success rate of HFR

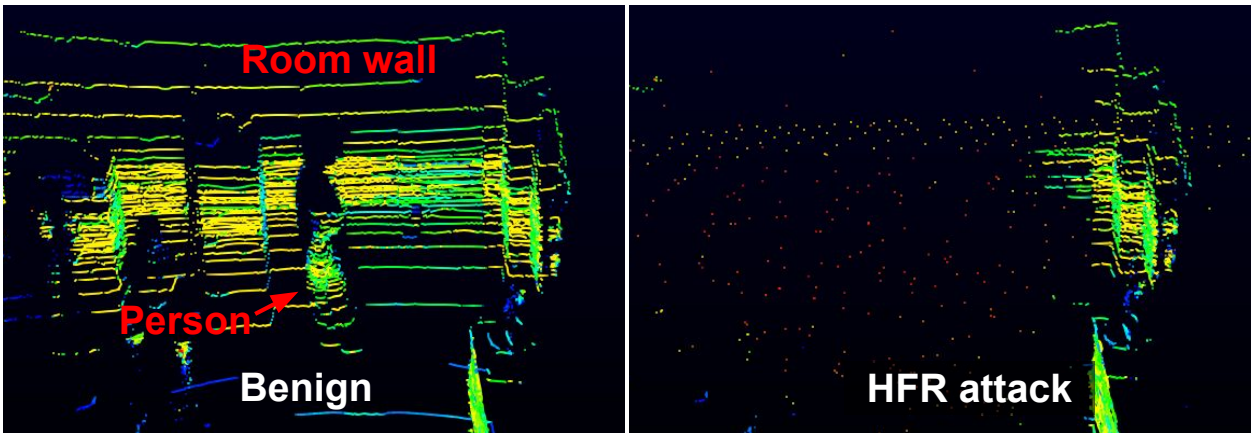


Figure 8.5.1: HFR attack effect on VLP-32c. Patterns of a person and the majority of the room wall are completely removed.

is slightly slower (72-78% versus 80-97% in PRA), which is expected since PRA has more precise control of points with synchronization.

For next-gen LiDARs with timing randomization, HFR is still highly-effective since the high-frequency lasers can still hit all legitimate measurements within the affected azimuthal range despite the randomized timing of legitimate laser firing, showing the capability of removing 4k-206k points (for OS1-32 it is only 28 due to the lack of laser diodes with matching wavelength, explained in §8.4.1). For Helios, the point removal success rate is lower mainly due to its extra-wide vertical FOV (Appendix 8.D.4); in the vertical range critical to real-world attack scenarios (e.g., 30-40° in the center), the success rate is similar to others (Fig. 8.5.2). For next-gen LiDAR with pulse fingerprinting (XT32), the point removal capability is significantly reduced due to the difficulty for the attack pulses to coincide with the fingerprinting interval. However, as we found in §8.4.1, there are still chances to randomly remove ∼100 points by using a pulse frequency that is most likely to coincide with the fingerprinting interval.

Finding 10 (RQ2): *For new-gen LiDARs, although they are no longer generally vulnerable to the latest practical attacks such as PRA, they unfortunately still remain generally vulnerable to object removal attacks, with a similar level of practical attack capabilities as PRA, due to the possibility of new asynchronized removal attack designs such as HFR.*

Table 8.5.1. LiDAR-level attack evaluation for object removal attacks. PRA [125] is only applicable to the first-gen LiDARs. Symbols are the same as in Table 8.4.1.

		First-Gen		New-Gen				
				w/ Timing Randomization				w/ Fingerprint
		VLP-16	VLP-32c	OS1-32	Helios	Horizon	L515	XT32
PRA [125]	$\mathcal{N}$	6,621	9,711	N/A	N/A	N/A	N/A	N/A
	$\mathcal{R}$	96.9%	82.9%	N/A	N/A	N/A	N/A	N/A
	θ	85.4°	73.2°	N/A	N/A	N/A	N/A	N/A
HFR (§8.3.3)	$\mathcal{N}$	5,358	8,778	28	4,108	19.2k	206k	113
	$\mathcal{R}$	78.1%	72.2%	43.8%	24.8%	79.9%	91.3%	2.1%
	θ	85.8°	76.0°	0.72°	103.4°	81.7°	70.0°	34.2°

* N/A: Attack is not applicable to the LiDAR

8.5.2 Object Detector-Level Measurements (RQ3)

Modeling of the Spoofing Attack Capability in Point Removal

Similar to the object injection attack side, we first perform a mathematical modeling of the spoofing attack capabilities in point removal in order to enable large-scale object detection-level attack capability measurements. For removal attacks, the point removal goal is the same for all point measurements that can be hit by the attack laser (e.g., move to within MOT for PRA, or place the point to a random position for HFR). Thus, the main factor that can affect the removal capability is whether the point is at an azimuth angle that can be effectively hit by the attack laser. For example, Fig. 8.5.2 shows the point removal percentage for the azimuth angles under attack. As shown, for VLP-16 under both PRA and HFR, the points in the center of the attack laser-hit azimuth range are almost 100% removed regardless of altitudes and distances, while the removal percentages symmetrically decrease from the center to the side.

Based on these observations, we model the attack capability for point removal $\mathcal{P}_R$ as follows:

$$\mathcal{P}_R(x_{ij}) := \begin{cases} \xi \cdot g(x_{ij}) & \text{if Bernoulli}(p_j) = 1 \\ x_{ij} & \text{othewise} \end{cases} \tag{8.2}$$

ξ is the error of the HFR attack, which distributes the original points to random locations

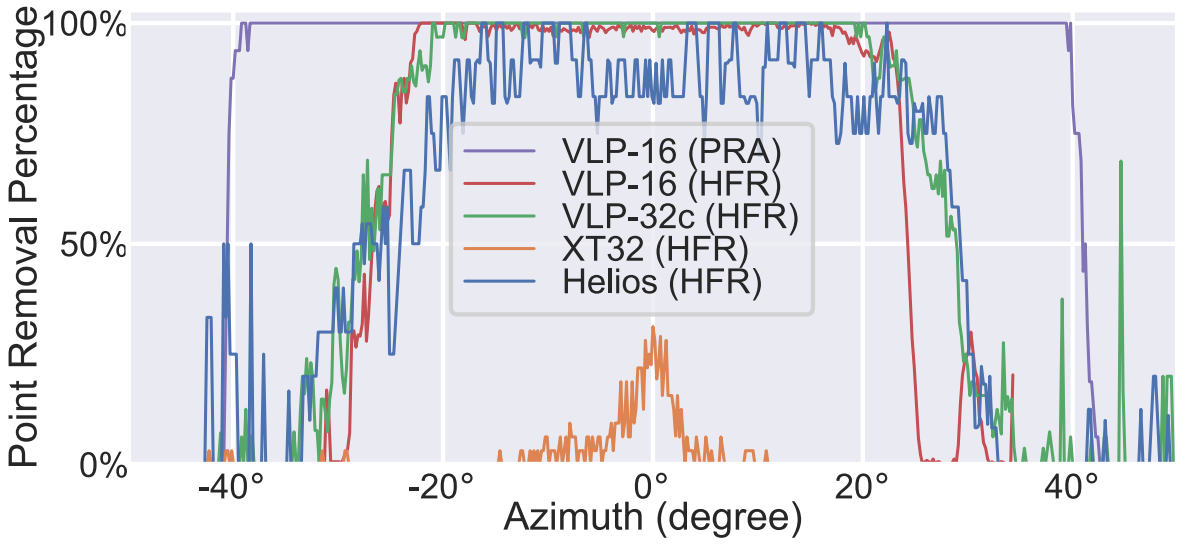


Figure 8.5.2: Point removal percentage of PRA [125] and the HFR attack (§8.3.3) for the azimuth angles under attack.

along with the laser direction $g(x_{ij})$. Thus, ξ can be written as $\xi := \mathcal{U}\left(0, \frac{c}{2} \cdot \frac{1}{f}\right)$, where $\mathcal{U}(a, b)$ is the uniform distribution with the minimum a and maximum b values, c is the speed of light, f is the frequency of HFR attack pulses. This is derived from the fact that the maximum ToF is capped by the time interval of the high-frequency attack pulses in HFR. For PRA, ξ is set to 0 since the point removal principle of PRA is to move the points toward the origin as much as possible to locate them within MOT (§8.2.3). After applying Eq. 8.2, if $\mathcal{P}_R(x_{ij})$ is within the MOT or outside of the maximum detection range (Table 8.3.1), point x_{ij} is removed from the point cloud. To the best of our knowledge, this is the first mathematical modeling of the point removal capability for LiDAR spoofing attacks.

Model-Level Attack Capability Measurement

Experimental Setup. We use the same evaluation scenarios as in §8.4.2 (i.e., 15 scenarios varying the distance between the victim to the vehicle object from 0m to 14m). We consider

the object removal as successful if the IoU between any detected objects and the ground truth object is 0. When applying $\mathcal{P}_R(.)$, we set p_j using the measured point removal percentages for different LiDARs as in Fig. 8.5.2. Note that for Helios, since its vertical FOV is much wider than others (Appendix 8.D.4), we only calculate the point removal percentage for the FOV range related to our attack scenario (i.e., 33°, which is enough to cover the height of the front vehicle to remove).

Results. Fig. 8.5.3 shows the object removal attack success rates for object detectors with different architectures and datasets. As shown, for the majority of the cases, HFR can reach similar success rates as PRA, which thus further extends our Finding 10 to the object detector level. As a validation of such attack effectiveness in the physical world, Fig. 8.5.4 shows the object removal attack effect against real vehicles using the HFR attack. As shown, the HFR attack is found to cause 5 front vehicles to become undetected by the Apollo model [7], with 100% success rate for over 10 seconds. In the figure, we can see the spatial features of the vehicles were completely destroyed (with some random points left) and thus no objects were detectable in the attacked region.

For new-gen LiDAR features, similar to the object injection side, timing randomization (Helios in Fig. 8.5.3) can significantly reduce the attack success rate in the majority of the cases (by 35% on average from VLP-16). However, pulse fingerprinting shows much higher defense effectiveness compared to the object injection side: with the pulse fingerprinting strength level of XT32 (i.e., ∼100 randomly removed points), the average attack success rate is significantly reduced by 63% on average from VLP-16, while such a reduction is 3% on average on the object injection side (Fig. 8.4.5). This is due to the trade-off of the object detector's sensitivity to object injection and removal attacks. Since fingerprinting only allows a random subset of 100 points to be injected, the object detectors that are more vulnerable to object injection are those that are very sensitive to even a heavily-occluded point cloud pattern with a very small number of object points.

Finding 11 (RQ3): *Both timing randomization and pulse fingerprinting show high defense capabilities against object removal attacks in general; the defense capability from pulse fingerprinting is especially strong when compared with that against object injection attacks.*

In addition, similar to our findings on the object injection attack side, the model robustness to object removal attacks shows a high diversity across different model architectures and different training datasets, which is especially prominent across training datasets. Interestingly, the model robustness properties are generally the *opposite* across object injection and removal attacks: for example, the models trained on Waymo and by Apollo are found to be much more robust against object injection attacks when compared to those trained on Lyft and nuScenes (§8.4.2), while the former ones become much less robust than the latter ones for object removal attacks. This might be because different dataset has different balances of the trade-off between false positives and more false negatives for the corner cases. As shown in Fig. 8.4.5, models trained on Lyft and nuScenes can have 100% detection rate of a vehicle even when the point cloud is randomly downsampled to as few as 10 points. This can allow the trained models to have a very low false negative rate even for highly-occluded legitimate vehicle point cloud patterns, which can thus make the models highly robust to object removal attacks, but this also makes them highly vulnerable to object injection attacks.

Finding 12 (RQ3): *The model robustness to object injection and removal attacks can be highly diverse when trained on different datasets. The choice of training datasets can incur large trade-offs between the model robustness to object injection attacks and that to object removal attacks.*

System-Level Attack Capability Measurement

Due to the downstream components such as object tracking in real-world applications such as autonomous driving (AD), object detector-level object removal effect may not directly

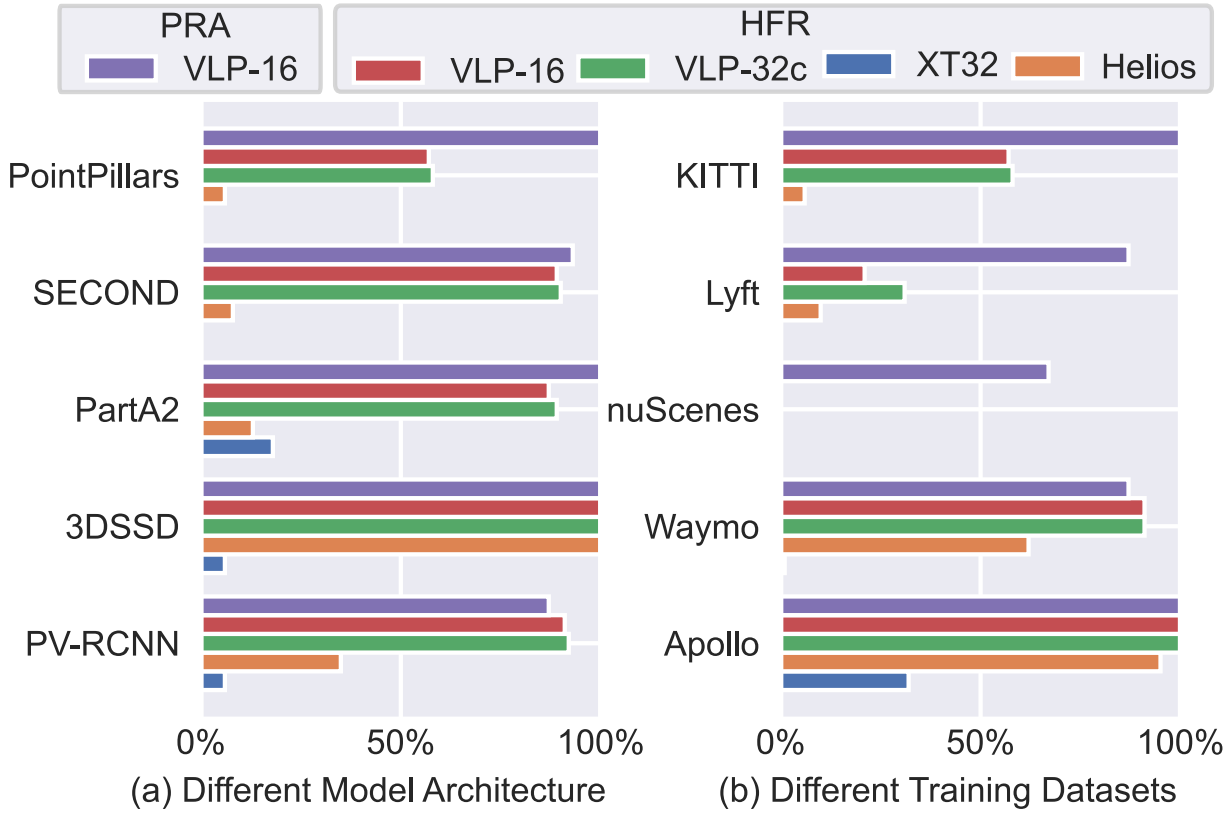


Figure 8.5.3: Object removal attack success rates of HFR and PRA for (a) 5 different models trained on the KITTI dataset and (b) PointPillars trained on 5 different datasets.

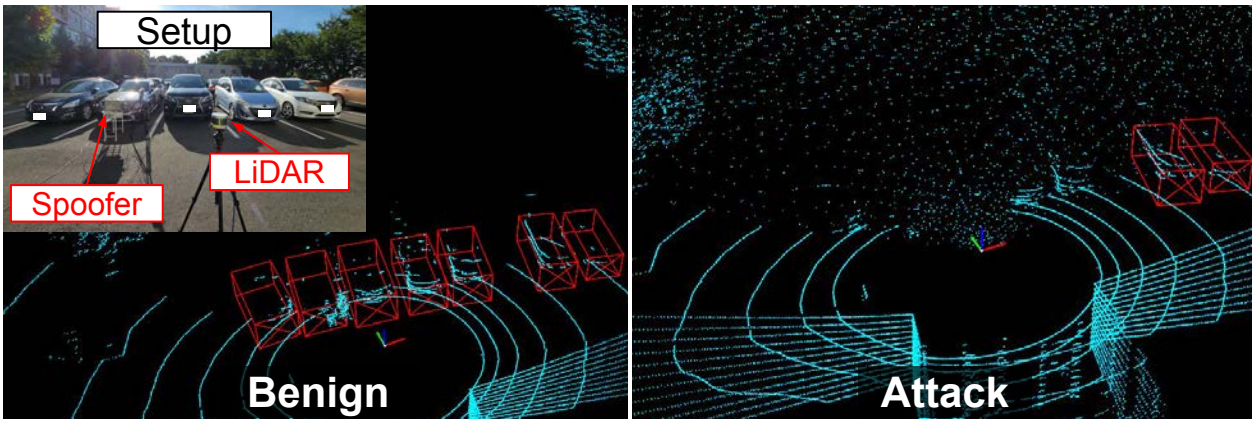


Figure 8.5.4: Object removal attack effect against real vehicles using the HFR attack. The 5 front vehicles become undetected with a 100% success rate for over 10 seconds (100 frames in total) by PointPillars [229] in Apollo [7].

imply system-level attack effect such as vehicle crashes [210, 300]. Thus, in this section we further measure the attack capabilities of different object removal attacks at the system level, with the focus on the representative application: AD.

Experimental Setup. We use a common setup widely used in previous work [130, 312, 185, 125]. For the AD system, we use Baidu Apollo 7.0 [7]. For the driving simulator, we use LGSVL [19]. Both systems are known as industry-grade software. Fig. 8.A.5 in Appendix illustrates the experiment scenario. We place a sedan vehicle as a target object 200 m away from the victim AD and make the victim AD drive toward the target. We add up to 1 m random perturbation both laterally and longitudinally to (1) the starting point of the victim and (2) the target object. Before reaching the target object, the AD vehicle reaches and keeps 40 km/h. We set an attack start distance $\mathcal{D}$ and only start to apply the simulated removal attack effect when the distance from the victim AD vehicle to the front vehicle is $\leq \mathcal{D}$. We use the collision rate over 10 trials as the evaluation metric, i.e., how many times the victim vehicle collides with the sedan out of 10 trials.

Results. Table 8.5.2 shows the collision rate over 10 trials for different LiDARs. As shown, although HFR has relatively weaker attack capabilities than PRA at raw point cloud (§8.5.1) and object detector (§8.5.2) levels due to the lack of synchronization, their attack capabilities at the system level are almost the same at every attack start distances $\mathcal{D}$, and reach 100% collision rate when $\mathcal{D}$ is over 18 m. However, due to the requirement of synchronization, PRA is only applicable to first-gen LiDARs such as VLP-16, while HRA can be generally applied to new-gen ones such as Helios with similar attack effectiveness. Fig. 8.5.5 shows an example attack trial under the HFR attack with $\mathcal{D} = 15$ m on Helios. The target sedan can be momentarily detected before the collision, but the detection disappears soon and the sedan becomes undetected until and also after the collision.

For next-gen LiDAR with pulse fingerprinting (XT32), although HFR can still have $\leq$32% model-level attack success rate (§8.5.2), it cannot achieve any system-level attack success at all (0%), which is likely because the object removal rate is not high enough to fool object tracking [210]. This suggests that pulse fingerprinting can be quite effective in defending against object removal attacks at the system level. videos of these experiments can be found

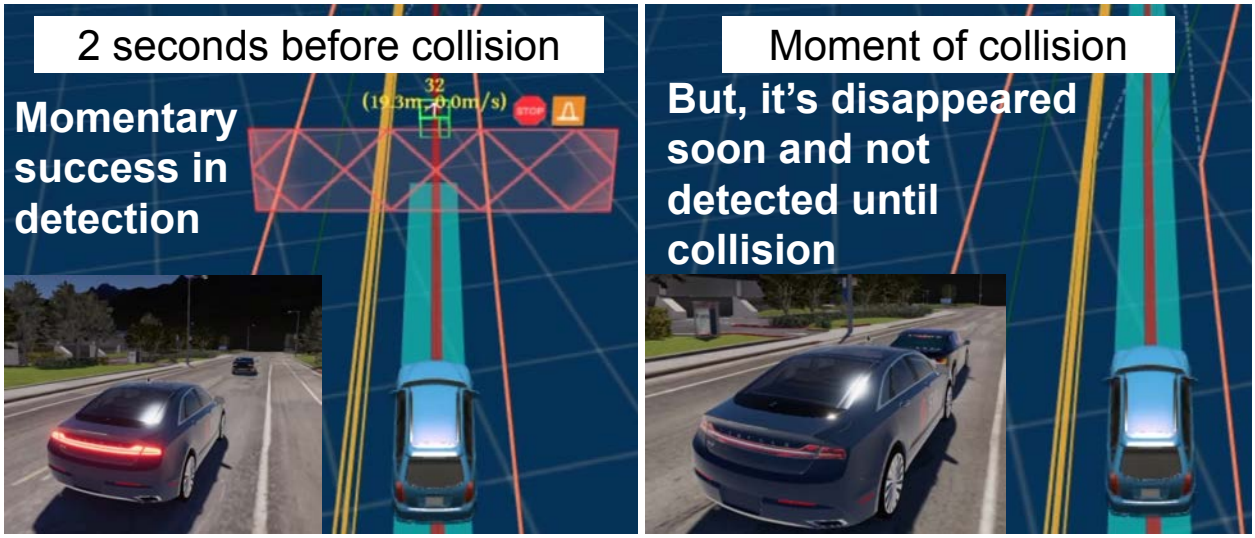


Figure 8.5.5: An example HFR attack trial with $\mathcal{D} = 15$ m on Helios. The remaining points are occasionally detected as an object but are not sufficient to avoid a collision.

Table 8.5.2. Vehicle collision rates over 10 trials using PRA and HFR for different LiDARs with varying attack start distances. **Bold** and underline highlight 100% and 0% collision rates.

		Benign	10m	15m	16m	17m	18m	19m	20m
PRA	VLP-16	0/10	0/10	5/10	8/10	9/10	**10/10**	**10/10**	**10/10**
HFR	VLP-16	0/10	0/10	6/10	7/10	8/10	**10/10**	**10/10**	**10/10**
	VLP-32c	0/10	1/10	9/10	8/10	**10/10**	**10/10**	**10/10**	**10/10**
	XT32	0/10	0/10	0/10	0/10	0/10	0/10	0/10	0/10
	Helios	0/10	0/10	6/10	5/10	**10/10**	**10/10**	**10/10**	**10/10**

at our website [32].

Finding 13 (RQ3): *Although the HFR attack has weaker attack capabilities than PRA at point cloud and object detector levels, the system-level attack effect is almost the same. Since the HFR attack does not require synchronization, this means that real-world LiDAR applications such as autonomous driving, no matter if using first- or new-gen LiDARs, are facing practical threats from object removal attacks. Meanwhile, pulse fingerprinting can be quite effective in defending against the removal attack effect at the system level.*

Table 8.6.1. Defense effectiveness and limitations of security-related features in the new-gen LiDARs against object injection and removal attacks. **Bold** means desired properties; Underline means undesire ones.

	Effectiveness		Limitations		
Features	Injection	Removal	Eye safety	Latency	Range↓*
Timing Random.	**High**	**High**	**No risk**	**Low impact**	**None**
Pulse Fingerprint	Mid	**High**	High risk	Mid impact	High
Simul. Firing	Low	None	**Low risk**	**Low impact**	**Low**

* Range↓: Degradation of the effective sensing range of LiDAR

8.6 Discussions

8.6.1 Defense Discussions

Sensor-Level Defenses

Through large-scale measurements, we identify the defense capability of popular security-related features in new-gen LiDARs on the spoofing attack capabilities. Table 8.6.1 summarizes our observations regarding their defense effectiveness and limitations against object injection and removal attacks.

Defenses against Object Injection Attacks. As shown in §8.4.2, the timing randomization feature has high mitigation capabilities, particularly on several models such as the industry-grade Apollo model. Even with low randomization entropy, the timing randomization shows high defense capabilities. Meanwhile, higher randomization entropy has a higher defense capability. We suggest that the magnitude of the randomization be set as high as possible, e.g., std. dev. $\sigma \geq 0.75$. Pulse fingerprinting also has mitigation capabilities but the complexity of fingerprinting in new-gen LiDARs today is not enough to effectively defend against injection attacks. Finally, simultaneous firing also has a defense effect as it can prevent the CPI attack capability (§8.4.1). However, this alone is unlikely to affect the object injection attack goal in general since the attacker is still able to inject an object point cloud with at least the same number of points as the chosen pattern (but just cannot control

the pattern very precisely).

Defenses against Object Removal Attacks. As discussed in §8.5.2, both timing randomization and pulse fingerprinting show high mitigation capability against removal attacks, especially pulse fingerprinting in XT32 shows high defense capability even against the HFR attack. The object detectors are typically tuned to avoid false negatives rather than false positives, especially for AD scenarios. This trade-off benefits pulse fingerprinting for defending against object removal attacks because the majority of object points are not compromised. However, this tuning is a backfire to defend against object injection attacks because it allows a hundred injected points to be detected as an object. However, the current XT32-level pulse fingerprinting is still not enough to prevent all attacks as the HFR attack still has 32% attack success rate on the Apollo model (§8.5.2).

To improve this further, more complex fingerprint encoding is needed. However, this is fairly non-trivial due to the dilemma between eye safety and detection range. More specifically, the most direct way to increase the fingerprint coding complexity is to increase the number of pulses for each distance measurement. However, the laser power per unit time is capped to ensure eye safety. For example, if we shoot N pulses for each point measurement, the power for each pulse should be $1/N$, and it will roughly degrade the effective sensing range by $1/\sqrt{N}$. To address the dilemma, future work can explore: (1) possible fingerprint coding designs both with high complexity and fewer pulses, and (2) use a wavelength to which the human eye is highly resistant, such as 1550 nm wavelength. For the simultaneous firing feature, for object removal it does not have any defense capability since it just makes the attack effects on some sub-group of the points in the chosen pattern equal instead of having a removal effect of them.

Summary. Overall, the timing randomization and pulse fingerprinting features show promising defense effectiveness on at least one of the object detector-side attack goals. Particularly, timing randomization has high effectiveness on both attack goals while suffering from

the least limitations in eye safety, latency, and sensing range, despite the simplicity of the method. Thus, we strongly recommend implementing it in future LiDARs as a highly cost-effective measure to improve their resiliency against LiDAR spoofing attacks.

Software-Level Defenses

Recently, several software-level defenses have been proposed against injection attacks. CARLO [312], SVF [312], and Shadow-Catcher[191] leverage an assumption that the spoofer cannot directly spoof a point cloud pattern with sufficient number of points and precision to make it indistinguishable from a benign one. However, we find that this assumption does not hold since our improved spoofer can achieve the CPI attack capability with a much larger number of points that is sufficient to directly spoof the complete point cloud pattern of a near-front vehicle (§8.4.1). Fortunately, VLP-16 [48] is the only LiDAR for which such CPI attack capability is feasible (§8.4.1). Thus, future research can focus more on more recent LiDARs, especially the new-gen ones, to explore software-level defense design possibilities. For object removal attacks, we still do not have effective defenses since PRA [125] is discovered very recently. A potential software-level defense for the HFR attack is to detect a unique characteristic of the HFR attack that will cause a randomized point distribution pattern (like salt-and-pepper noise) in the area under attack, which is caused by the attack design (§8.3.3) and can rarely occur in benign cases.

8.6.2 Limitations of Our Study

Aiming at Driving AD Vehicles

In this study, we do not discuss the deployability of LiDAR spoofing attacks against real-world applications such as AD, especially when the victim AD vehicle is moving in high

speed. We focus more on investigating the security properties of different types of LiDARs, rather than the system-level security analysis of autonomous driving in the real world. In a future study, we plan to demonstrate attacks against driving vehicles. A recent study [125] has demonstrated a system capable of victim LiDAR tracking and spoofer aiming to address this problem. Therefore, we expect that targeting a driving vehicle is feasible.

LiDAR Model Coverage

While we cover popular LiDARs as many as possible with our best efforts, it is infeasible to cover all public LiDARs due to our budget and their supply capacity. For example, we cannot cover 1550 nm LiDARs such as AEye [2] and Luminar [24] which utilize unique technology, e.g. adaptive scanning and FMCW ToF, and thus potentially have different spoofing capabilities. We also cannot cover private LiDARs such as those in Waymo's robotaxi [69] since they are not publicly available on the market.

8.6.3 Considerations for Safe Experiments

All experiments were conducted in a controlled environment and we wore safety goggles for extra eye safety. Note that the unit-area peak power of our laser is actually weaker than prior works [125] as we use a 50% larger aperture lens.

8.7 Conclusion

In this work, we conduct the first large-scale measurement study on LiDAR spoofing attack capabilities on object detectors with 9 popular LiDARs, covering both first- and new-gen LiDARs, and 3 major types of object detectors trained on 5 different datasets. To facilitate

the measurements, we (1) identify spoofer improvements that significantly improve the latest spoofing capability, (2) identify a new synchronized object removal attack (HFR)), and (3) perform novel mathematical modeling for both object injection and removal attacks. Through this study, we are able to uncover a total of 15 novel findings, including not only completely new ones due to the measurement angle novelty, but also many that can directly challenge the latest understandings in this problem space. We discuss defenses based on the findings. We hope that our findings can inspire and facilitate future security research on LiDAR spoofing, especially those targeting safety-critical application domains such as autonomous driving.

Appendix

8.A Additional Figures

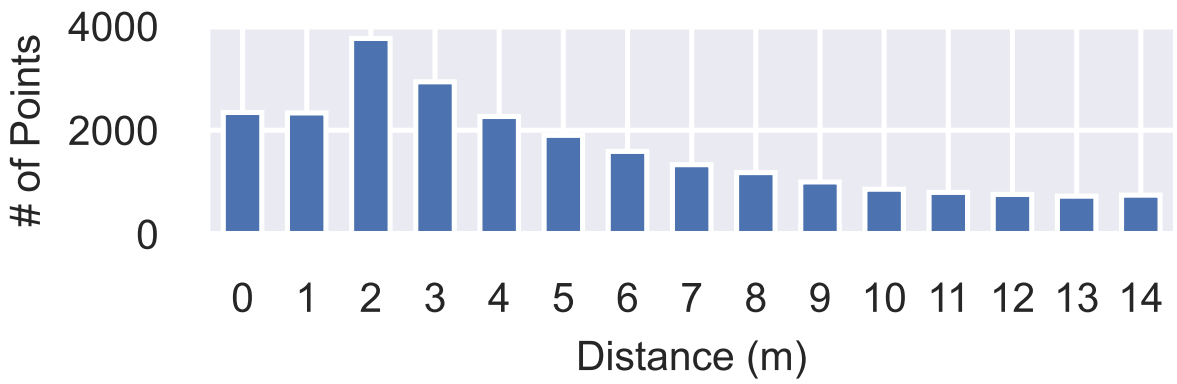


Figure 8.A.1: Number of points for the targeted vehicle at each distance to the victim ego vehicle.

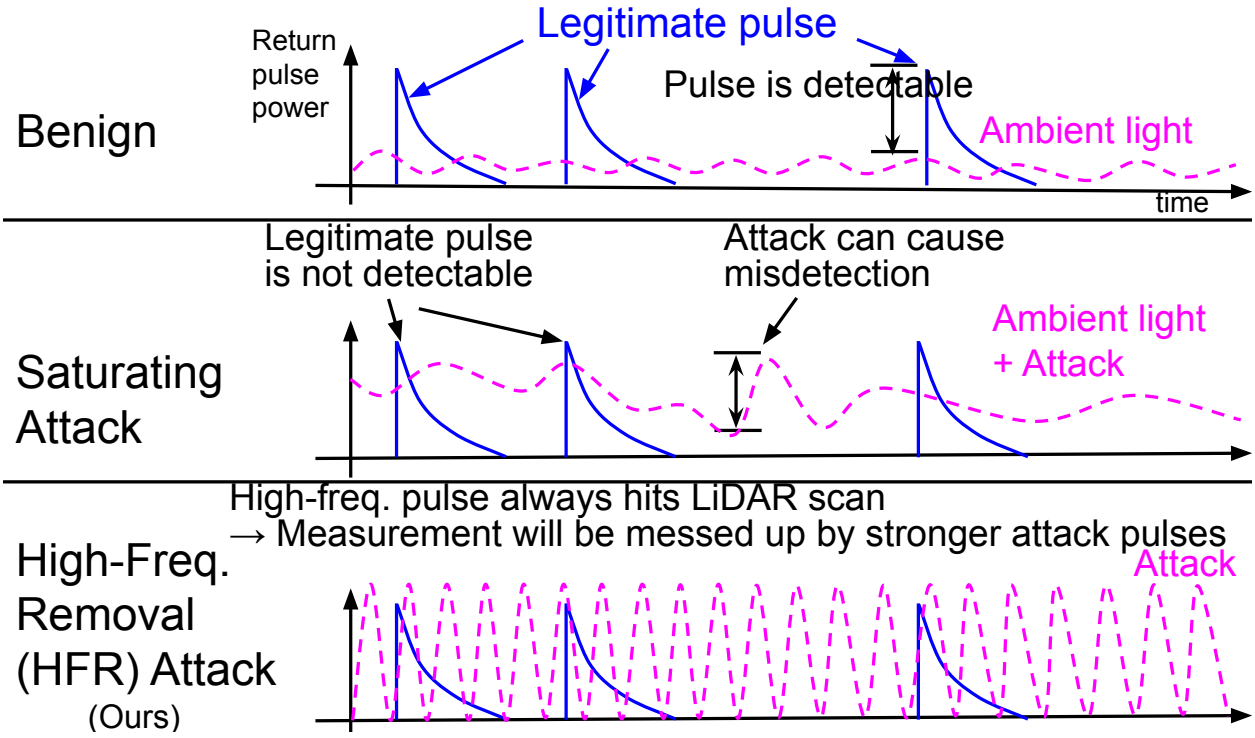


Figure 8.A.2: Attack mechanism difference between the saturating attack and the newly-identified HFR attack (§8.3.3).

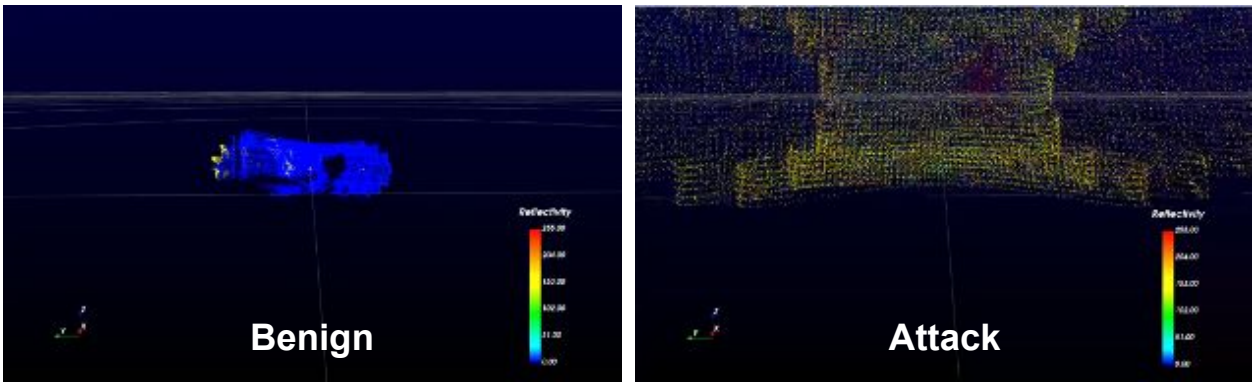


Figure 8.A.3: Point cloud of Livox Horizon in benign and attack scenarios. The point cloud is totally randomized by the attack and the pattern of the rectangle area (our lab room) in the left figure completely disappears as shown in the right figure.

8.B Detailed Explanations on Synchronized LiDAR Spoofing Attacks

Fig. 8.B.1 illustrates the synchronized attacks on VLP-16. The attack mechanism is common on both the injection attacks [306, 130, 312, 185] and the removal attacks [190, 125] since their

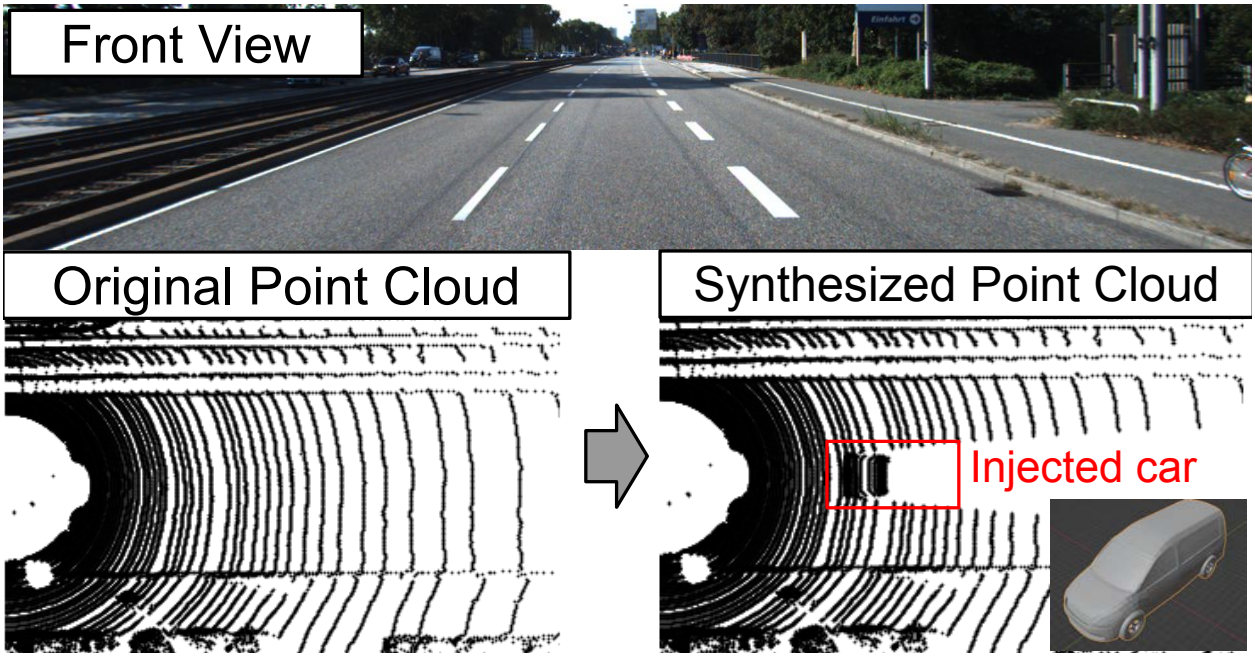


Figure 8.A.4: Targeted experiment scenario from KITTI [175].

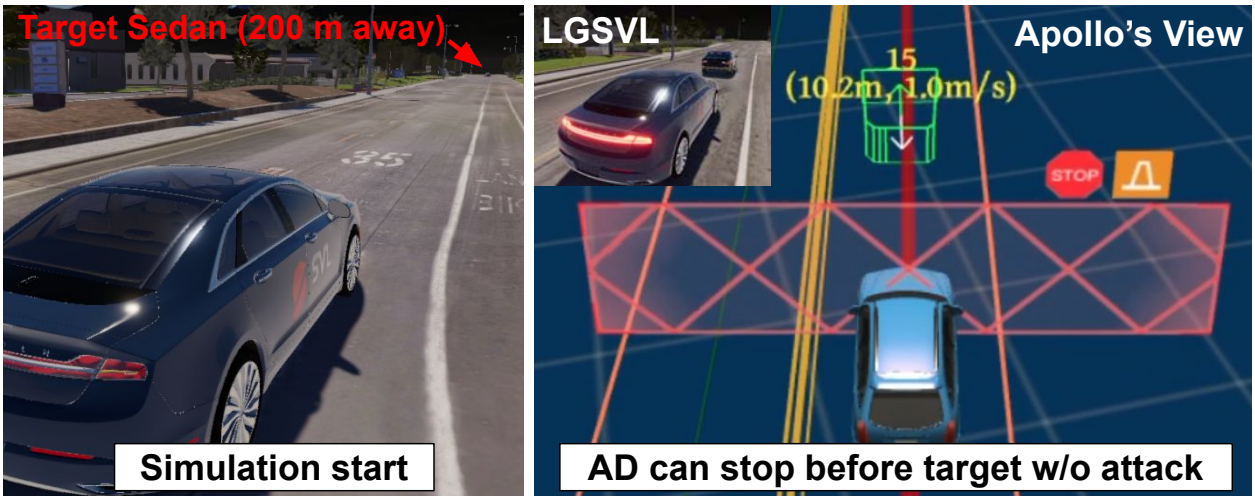


Figure 8.A.5: Illustration of the evaluation scenario. AD starts driving from 200 m away and it can always successfully stop before the target sedan in the scenarios without attack.

difference is whether they move points at target locations or move points into undetectable area. The attack procedure consists of 3 steps: ① PD first receive the legitimate laser from the target LiDAR to know when the LiDAR will scan the point that the attacker want to change; ② FG plans when to fire lasers based on the information from PD and the pre-defined scan pattern of the target LiDAR; ③ the laser is fired based on the plan from FG through the gate driver and LD. Therefore, to achieve the CPI attack capabilities, the attacker must know exactly where LiDAR is scanning and the scan schedule must be

predefined or predictable. The timing randomization breaks the assumption by randomizing the scan schedule.

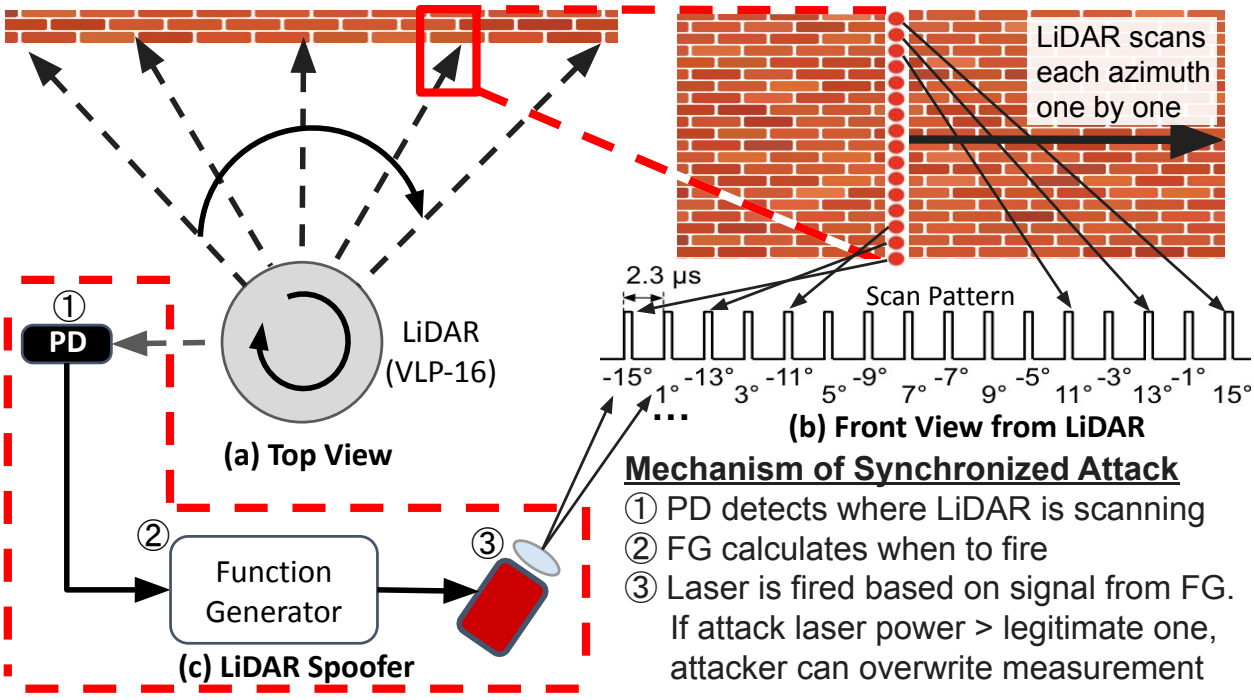


Figure 8.B.1: Illustration of synchronized attacks on VLP-16. VLP-16 scans each azimuth (every 0.1°) one by one. At an azimuth, it fires 16 lasers vertically based on the pre-defined scan pattern. Once attackers can identify its state by PD, they can know *when* to fire a malicious laser based on the pattern.

8.C Details of LiDAR Spoofer Used in Experiments

Our spoofer operates in 4 steps (shown in Fig. 8.3.1): ① The photodetector (PD) receives a laser pulse from LiDAR; ② the transimpedance amplifier (TIA) amplifies the pulse to 100 mV to feed it into the function generator (FG); ③ FG synchronizes with the LiDAR spanning pattern based on the received pulse and generates laser pulses according to the points that the attacker wants to inject; ④ The gate driver (GD) drives the laser diode (LD) which transmits a laser pulse to the LiDAR.

We use the same PD (S6775 [38]) and TIA (TL082 [44]) as the prior work [125]. For other components, we used the equivalent or enhanced devices: FG is Agilent 81160A [1], GD is

EPC9126HC [13], and LD is SPL PL90_3 [12]. In particular, we utilize a more functional FG (Agilent 81160A) than the one used in previous studies (Tektronix AFG320 [3]) to better facilitate the exploration of the CPI attack capability.

Furthermore, the improved optical design significantly affects the number and angle coverage of spoofable points as discussed in §8.3.4. Fig. 8.C.1 illustrates the 3 types of laser beam shapes: converged, diverged, and collimated. Among them, the collimated beam is the most ideal for LiDAR spoofing attacks because it can deliver the laser to the target with a minimum loss. To precisely calibrate the lens setup, we develop a device that can adjust the distance between the LD and the lens as designed. As shown in Fig. 8.3.1, the lens is connected to the frame with a hollow screw so that we can adjust it precisely.

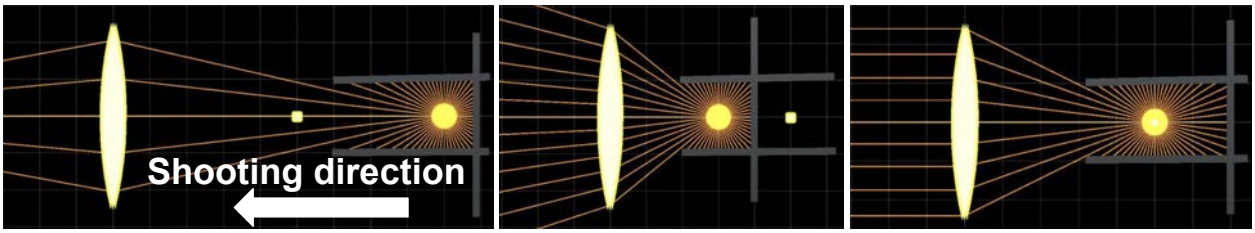


Figure 8.C.1: Illustration of the 3 types of laser beam shapes: converged, diverged, and collimated.

8.D Case Study Results on Specific LiDARs

8.D.1 The Use of Rare Laser Wavelength in OS1-32

For OS1-32 [10], we find that we are not able to inject many points since we only have a 905 nm wavelength (SPL PL90_3), which is also the setup for all prior works [306, 130, 312, 185, 125]. The spoofing with 905 nm wavelength is not effective on OS1-32 that uses an 865 nm wavelength laser. We tried our best to purchase 865 nm laser diodes, but we find that high-power 865 nm laser diodes are actually not generally publicly available for personal

use. We also tried low-power 860 nm laser diodes such as OPV380 and RLD85PZJ4, but they cannot work more than 4 W, which is too low to inject any points (our 905 nm diodes work at 90W to achieve successful injections). From some perspective, this may imply that developing/using a LiDAR with a relatively rare wavelength may have a "immunization" effect against LiDAR spoofing attacks in practice since it may make it harder for attackers to acquire the corresponding attack laser diodes. At this point, only a very small portion of LiDARs has this property; a recent survey [20] reports that 905 nm wavelength laser is the most commonly used and 865 nm wavelength laser does not even appear in their graph due to its rarity.

Nevertheless, such a potential "immunization" effect would lose effectiveness if more LiDARs adopt 865 nm wavelength as this may boost the availability of 865 nm laser diodes on market. Also we find that not all LiDARs using a rare wavelength can directly have such an "immunization" effect. For example, although Realsense L515 is using 860 nm instead of 905 nm, we find that it is still directly attackable with the 905 nm attack laser since it does not have a band-pass filter to exclude light other than its wavelength.

8.D.2 Relay Attack on Leddar Pixell

Flash LiDAR fires a wide diverging laser beam and needs to receive them at the same time. This mechanism is exploitable by the relay attack (§8.2.3), which is not effective on the scanning LiDARs because PD can only receive a very limited azimuthal range, and the spoofed points will always be located far from the spoofer as discussed in [306]. However, we find that the relay attack can be effective on the flash LiDAR as a removal attack. As its laser covers a wide range, the attacker can also disturb a wider range. If the attacker's laser is strong enough, they can remove the original points by moving them to farther points than the spoofer. Fig. 8.D.1 shows the results of the relay attack on Leddar Pixell [18]. The

entire row of detection is moved to very far position. Note that Leddar Pixell has a special design that consists of multiple rows of scanning. If it is a typical flash LiDAR, the relay attack can cause an effect on more rows.

Finding 14 (RQ2): *Compared to traditional scanning LiDARs, relay attack may be more effective on flash LiDARs as an object removal attack.*

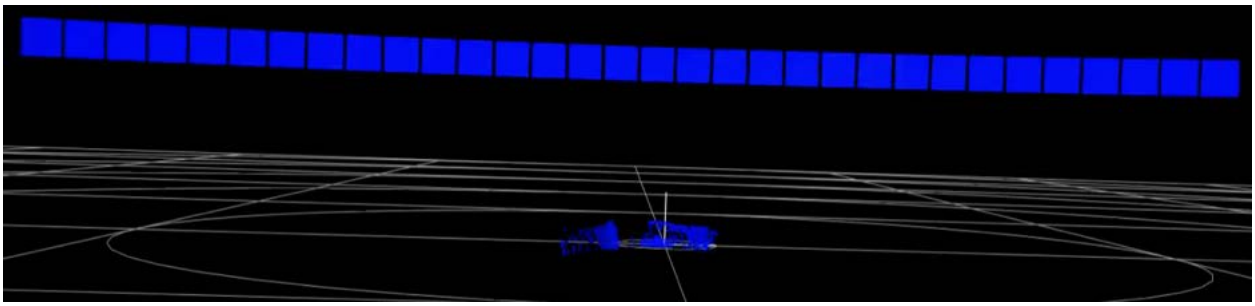


Figure 8.D.1: Relay Attack on Leddar Pixell [18]. The attack moves the detected area to a much farther location.

8.D.3 Zero-Distance Sensing of XT32

As listed in Table 8.3.1, XT32 [49] is capable to measure the distance to LiDAR down to 0 m. The zero-distance sensing is not available in most LiDARs due to its technical challenges: detecting a very short time of laser flight is difficult to distinguish from noise due to unideal hardware calibration as discussed in [125]. However, it is technically realizable, which thus directly breaks the design assumption of the latest PRA attack [125] (i.e., needs enough MOT, §8.2.3).

Finding 15 (RQ1): *LiDARs with zero-distance sensing do exist, which directly breaks the design assumption of the latest PRA attack [125].*

8.D.4 Extra Wide Vertical FOV of Helios 5515

As listed in Table 8.3.1, Helios [37] has an extra wide vertical field-of-view (FOV), 70°, which is far wider than the normal range of 30-40° (Table 8.3.1). It results in the lower

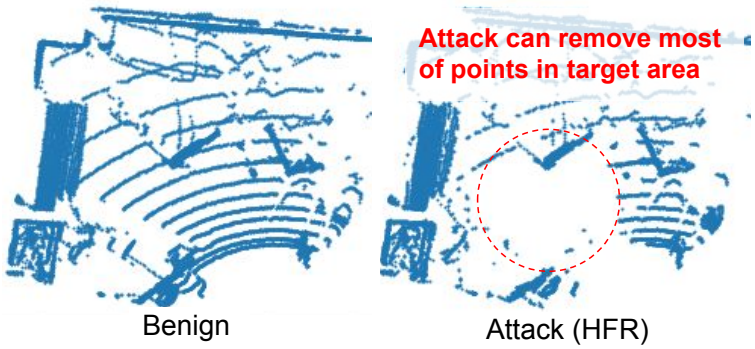


Figure 8.D.2: HFR attack effect on Helios [37]. Our spoofer cannot cover the entire FOV, but can still be effective in the center area.

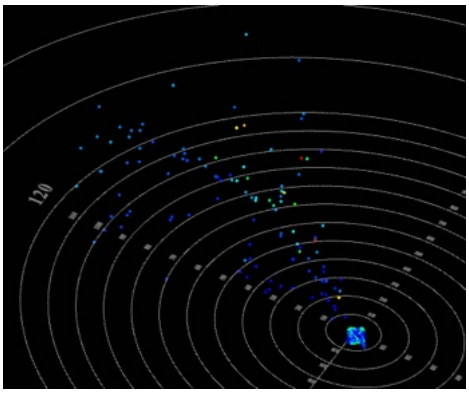


Figure 8.D.3: Spoofed points on XT32 [49].

attack success rate $\mathcal{R} = 19.4\%$ even though the number of points is large, 3,203 points. This is because our spoofer cannot cover all altitudes of Helios. However, it does not mean that Helios is more robust against attacks because the attacker does not need to attack the entire vertical FOV. Fig. 8.D.2 shows an example of the HFR attack effect on Helios. As shown, the HFR attack is successful in the middle of the vertical FOV, which can allow attackers to hide objects there.

8.D.5 Simultaneous Firing on OS1-32 and VLS-128

Fig. 8.D.4 shows the spoofed points on OS1-32 [10] and VLS-128 [4].As shown, OS1-32 fires and scans 32 lasers vertically, and thus we can only move the depth of each vertical line with 32 points simultaneously. VLS-128 shoots 8 lasers based on a predefined pattern, and thus we can only simultaneously change the depth of each group consisting of 8 points. As we do not have the capability to selectively return a laser to each simultaneous laser, we are not able to achieve the CPI attack capability as discussed in §8.4.1.

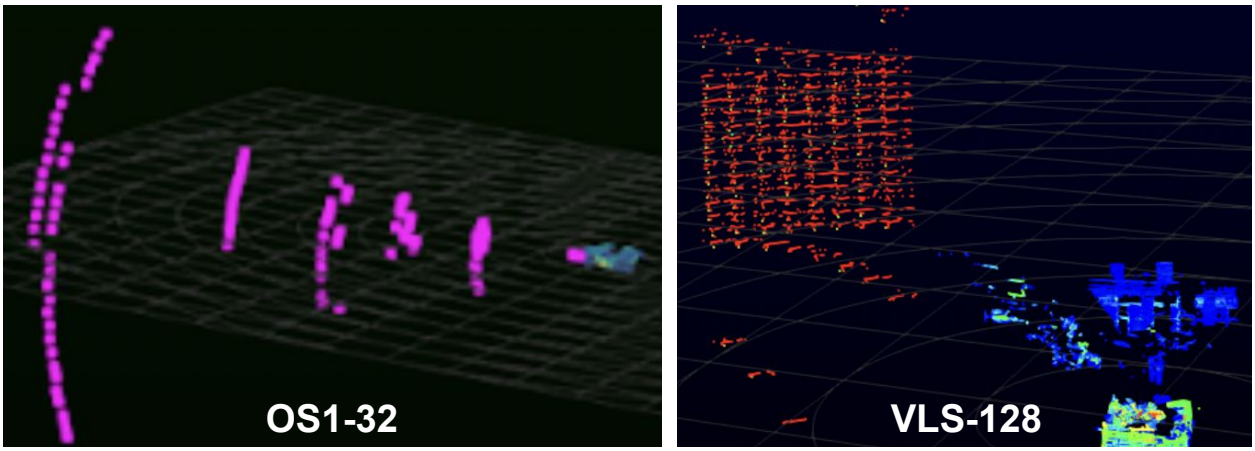


Figure 8.D.4: Spoofing results for OS1-32 and VLS-128.

8.E Taxonomy of 3D Object Detectors

Voxel-based Methods. Voxel-based method is a very early, but still dominant approach [229, 379, 305, 355]. To deal with the irregular structure of point clouds, this approach aggregates points into 3D voxels to make the CNN work effectively. PointPillars [229] is the most widely used in autonomous driving systems such as Baidu Apollo [7] and Autoware [219] because it can achieve higher throughput by constructing voxels as pillars perpendicular to the grounds although the z-axis resolution becomes coarse. In §8.4.2, we find that the voxel-based method is slightly less robust than the other methods to the injection attacks as it cannot recognize the detailed geometry of points. This design allows to achieve of higher throughput and is widely used in autonomous driving systems such as Baidu Apollo [7] and Autoware [219] since the z-axis of an object is typically unimportant in ground vehicles. Part-A^2 [305] achieves a higher attack success rate than the above methods by leveraging multi-scale voxelization and anchor-based strategies.

Point-based Methods. Point-based methods directly handle point cloud without voxelization. To efficiently handle point clouds, this approach utilizes permutation-invariant operators. PointRCNN [304] is a two-stage method inspired by FastRCNN [176] in 2D object detection. PointRCNN generate 3D proposals and point features with PointNet backborns [272]. 3DSSD [357] is a single stage method inspired by SSD [247] in 2D object

detection. By leaving only representative points in the downsampling, 3DSSD removes the feature propagation layer and achieves higher throughput than two-stage method.

Point Voxel-based Methods. Point voxel-based method [303, 143] is a hybrid approach of the voxel-based and point-based methods. PV-RCNN [303] is a two-stage method that uses the voxel-based method in the first stage (3D proposal generation) and the point-based method in the second stage (regional refinement). In §8.4.2, PV-RCNN [303] shows the highest robustness to the injection attacks. The second stage with PointNet backbone, which is not in 3DSSD, enables to obtain more detailed point geometry.

8.F Impact of Pulse Frequency and Laser Drive Voltage on HFR Attack

For HFR attack, the higher the attack pulse frequency is, the more effective the point removal attack capability should be since this can make it more likely for the attack pulse to (1) hit the laser receiving window on the victim LiDAR side to affect the legitimate point measurement; and (2) push the attack-induced random position measurements to be within MOT (and thus become undetectable). However, highly-frequent laser firing makes the temperature of LD high and thus results in the degradation of laser intensity, which affects the spoofing capability. We thus experimentally study this trade-off. As shown in Fig. 8.F.1, when we increase the frequency, the number of removed points peaks at $\sim$1 MHz and decreases after that. Thus, we use 1 MHz as the default attack laser frequency in our other experiments.

Meanwhile, the attack laser intensity at the firing time also practically affects the attack effectiveness, since a lower one is not able to ensure that the attack laser received at the victim is stronger than the legitimate one. To understand this, we also vary the attack laser

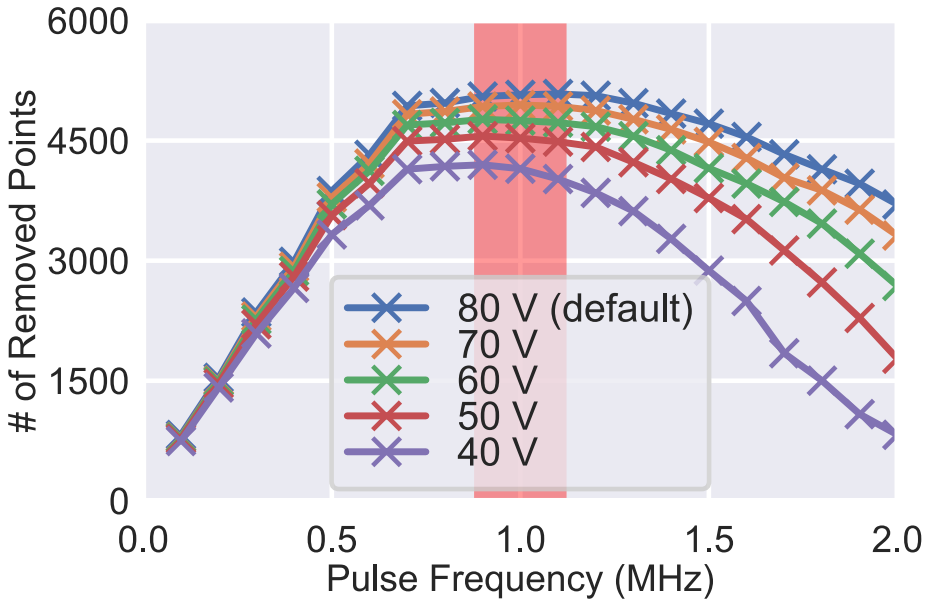


Figure 8.F.1: The relationship between the attack pulse frequency and the removed points by HFR attack.

drive voltage in our experiments. Since VLP-16 has ∼30V laser drive voltage, we vary the attack laser drive voltage from 40 to 80 V (80V is the maximum possible one in our setup). As shown in Fig. 8.F.1, the number of removed points monotonically decreases with the voltage value. Thus, we use 80V as the default voltage in our experiments.

8.G Criteria to Count the Number of Removed and Injected Points

To quantitatively evaluate the attack performance, we design a systematic method to count the number of removed and injected points by attacks. We conduct all indoor experiments 5 m away facing the wall of the room, i.e., the legitimate reflection should be from 5 m away. For point injection attacks, we judge whether a point is spoofed or not based on its intensity. When the intensity scale is 0 to 255, The intensity of reflections from the walls of the room is typically below 70. On the other hand, the attack laser has more than 200

as we directly shoot lasers at the LiDAR without any reflections. We thus used 80 as the threshold for indoor experiments to differentiate whether legitimate or injected points. For the outdoor experiments, we selected an adequate threshold from 80 to 150 based on the maximum intensity in a benign point cloud. For point removal attacks, we also utilize the intensity of the point. We first identify the attack-induced spoofed points with the same threshold of intensity. Next, we subtract the remaining points in the attacked point cloud from the benign point cloud; now the remaining points in the benign point cloud are thus the removed benign points by the attack. To calculate the point removal percentage of each azimuth, we split the area into pies corresponding to each azimuth and apply the same calculation. Based on our measurement, our method can capture 96% of injected or removed points, i.e., 4% of injected or removed points could be missed.

Chapter 9

On the Realism of LiDAR Spoofing Attacks against Autonomous Driving Vehicle at High Speed and Long Distance

9.1 Introduction

The rapid implementation of Autonomous Driving (AD) on public roads poses emerging challenges for our society. AD vehicles with roof-mounted sensor stacks are also a common sight [89, 65] in many cities around the world. Waymo's robotaxi services are available in several cities, such as Phoenix, San Francisco, and Los Angeles [69]. LiDAR (Light Detection and Ranging) plays a significant role in enabling such a rapid implementation of AD, especially for driverless Level 4 AD [90], which heavily relies on accurate 3D environmental information obtained via LiDAR to achieve production-level object de-

tection and localization. Meanwhile, the essential roles of LiDAR in AD have motivated extensive research efforts to ensure its security due to the potentially fatal safety implications. One of the major security concerns is the robustness against LiDAR spoofing attacks [270, 306, 130, 312, 185, 213, 290, 125, 129], which project malicious lasers against target LiDARs to overwrite their legitimate distance measurements.

In the context of object detection, LiDAR spoofing attacks have been demonstrated to have two attack effects: object injection attacks [306, 130, 312, 185, 213, 290] and object removal attacks [306, 213, 290, 125]. These attacks have demonstrated successful results at the object detection model level and in low-speed, short-distance lab environments. However, none of the prior works have successfully proven their effectiveness in practical high-speed, long-range autonomous driving scenarios, despite suggesting potential safety and security impacts on AD vehicles. Table 9.1.1 provides an overview of existing LiDAR spoofing attacks demonstrated in the physical world. As listed, prior work predominantly focuses on stationary lab-level setups or dynamic but impractical low-speed setups (e.g., at most 5 km/h), and none of prior work has evaluated LiDAR spoofing attacks against a vehicle actually controlled by an AD software stack. Based on our survey, we identify the following three research limitations that potentially prevent demonstrating LiDAR spoofing attacks in practical AD scenarios:

Lack of practical detection and tracking system capable of high speeds and long distances: Precise and long-range detection and tracking system is essential to keep LiDAR spoof attacks effective on moving AD vehicles, especially for removal attacks, which need to be effective for at least several consecutive frames. The only previous attempt is camera-based detection and tracking with a pan-tilt system PTX-ATX18 [125, 127]. However, this system has not been evaluated in practical scenarios at long distances, in which we found that camera-based detection cannot be effective as in §9.5.1.

Lack of practical spoofing devices capable on public roads: No prior work has demonstrated their attacks in realistic environments. Most of them have only been evaluated

in indoor lab-level or outdoor setups where AD vehicles are driving slowly (<5km/h), such as parking lots. To deploy LiDAR spoofing attacks on real roads, the hardware requirements will be significantly higher to enable high-speed and long-range attacks. For example, when spoofing at long distances such as 100m, even small distortions in optics, errors in detection results, and delays in motor control could lead to an error of a few meters in aiming.

Lack of practical spoofing attacks against recent LiDARs: As also pointed out in [290], recent LiDARs, so-called New-Gen LiDARs, equip security-related features, such as timing randomization and pulse fingerprinting, which can directly foil essential assumptions in prior attacks [130, 312, 125]. The HFR attack [290] is the only attack that demonstrates the removal attack capability even against New-Gen LiDARs. However, the HFR attack has not shown high attack effectiveness against LiDARs with pulse fingerprinting.

Motivated by these limitations in prior work, we pursue the following research question:

Can LiDAR spoofing attacks have end-to-end safety impacts in practical AD scenarios?

To answer this question, we design a novel Moving Vehicle Spoofing system (MVS system) that enables us to deploy LiDAR spoofing attacks in practical AD scenarios. Our MVS system consists of 3 subsystems: infrared (IR) camera-based detection and tracking system, auto-aiming system, and LiDAR spoofing system with significant improvements over prior work. Furthermore, we design a new attack named adaptive high-frequency removal (A-HFR) attack that can be effective against LiDARs with pulse fingerprinting. The A-HFR attack leverages gray-box knowledge of the target LiDARs to avoid overheating the laser diode in the spoofer; it allows the emission of laser pulses at a frequency more than 25 times higher than that of the HFR attack while using the same laser.

This paper is structured as follows: In §9.2, we summarize previous efforts in LiDAR spoofing attacks and formulate our threat model to attack AD vehicles driving at high speeds from long distances. In §9.3, we describe the design details of our MVS system consisting of

3 subsystems. In §9.4, we introduce the methodology of our A-HFR attack and describe how the A-HFR attack can be effective against LiDARs with pulse fingerprinting. In §9.5, we evaluate the attack capability of our MVS system and the effectiveness of our A-HFR attack. We find that the prior vision-based detection can only detect the LiDAR at up to 5 meters. On the other hand, our IR camera-based detection can detect the target LiDAR even at distances $\geq$100 meters regardless of the LiDAR types. We also explore multiple tracking designs including the Kalman filter. We evaluate the attack effectiveness on three commercial LiDARs and find that for all LiDARs, the A-HFR attack can successfully remove over 96% of the point cloud within a 20° horizontal and a 16° vertical angle.

In §9.6, we demonstrate LiDAR spoofing attacks against the victim vehicle driving at high speed, up to 60 km/h. We find that the HFR attack achieves $\geq$96% attack success rates until the braking distance (20 meters) at 60 km/h. The A-HFR attack also achieves 100% attack success rates until 40 meters away. The injection attacks are also successful with $\geq$1k injected points at 60 km/h. In §9.7, we perform an end-to-end closed-loop attack evaluation on an AD vehicle operated by Autoware.ai [219]. We demonstrate that both object injection and removal attacks with our MVS system can cause serious safety consequences. Particularly for the object removal attack, the victim fails to detect an SUV car-sized object in front of it. In §9.8, we finally discuss the findings of this study, the limitations of this study, and potential countermeasures.

In summary, our study has the following contributions:

- We design a novel MVS system that can conduct LiDAR spoofing attacks in practical AD scenarios. The MVS system consists of 3 subsystems: IR camera-based detection and tracking system, auto-aiming system, and LiDAR spoofing systems. Our MVS systems can aim at a target vehicle driving at 60 km/h from 110 meters away.
- We design a new object removal attack, A-HFR, which can be effective against LiDARs

Table 9.1.1. Literature survey on prior works that evaluate LiDAR spoofing attacks in the physical world. We are the first to demonstrate the LiDAR spoofing attack in practical high-speed and long-range AD scenarios and the first to perform the attacks against a vehicle controlled by an AD software stack.

	Attack on Moving Target	Attack on AD Vehicle	Maximum Speed	Attack from Roadside	Attack Goals		Maximum Attack Range
					Injection	Removal	
Ours	✓	✓	**60 km/h**	✓	✓	✓	**110 m**
Cao et al. [125]	✓	-	5 km/h	-	-	✓	10 m
Cao et al. [127]	✓	-	0.4 km/h	-	✓	-	4 m
Jin et al. [213]	✓	-	0 km/h (running parallel)	-	✓	✓	15 m
Petit et al. [270]	-	-	-	-	✓	-	1 m
Shin et al. [306]	-	-	-	-	-	✓	5 m
Sun. et al. [312]	-	-	-	-	✓	-	5 m
Hallyburton et al. [185]	-	-	-	-	✓	-	5 m
Sato et al. [290]	-	-	-	-	✓	✓	10 m

✓ : Covered, - : Not Covered

with pulse fingerprinting by utilizing a gray-box knowledge of the target LiDAR to avoid overheating the laser diode. The A-HFR attack can remove ≥96% of points within attack angles even under pulse fingerprinting.

- We are the first to demonstrate LiDAR spoofing attacks against practical AD scenarios where the victim vehicle is driving at high speeds and the attack is launched from long distances. Our object removal attack achieves ≥96% attack success rate against vehicles driving at 60 km/h up to their braking distance (20 meters).

- We are the first to deploy LiDAR spoofing attacks against a vehicle actually controlled by a popular AD stack (Autoware.ai [219]) and demonstrate the end-to-end serious safety consequences (e.g., hard crash into a parking mock car).

Project Website: Demo videos and more detailed hardware configurations are released on **https://sites.google.com/view/cav-sec/new-gen-lidar-sec**.

9.2 Background

9.2.1 LiDAR Spoofing Attacks

LiDAR spoofing attacks [270, 306, 130, 312, 185, 213, 290, 125, 129] compromise the distance measurements of LiDAR sensors by overwriting legitimate laser signals with higher-power

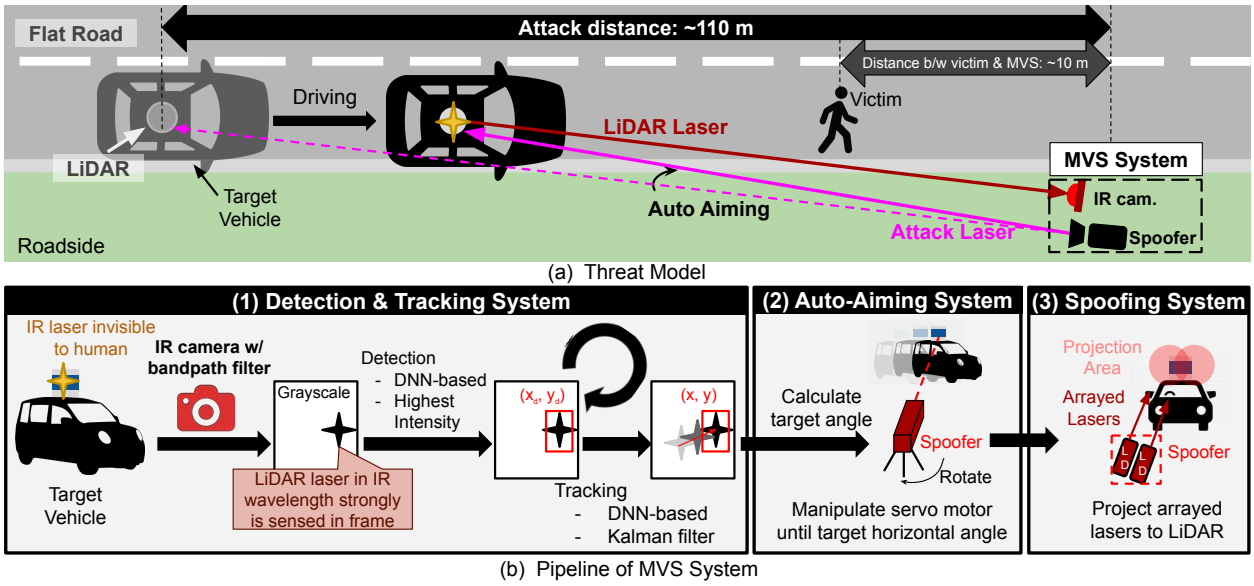


Figure 9.1.1: Overview of our threat model and pipeline of our MVS system consisted of 3 subsystems: (1) detection and tracking system with IR camera, (2) auto-aiming system with high-precision servo motor, and (3) spoofing system with arrayed lasers to achieve a wider attack projection area. Our MVS system can attack a driving AD from ≥110 meters away.

malicious lasers. Table 9.1.1 lists an overview of the existing LiDAR spoofing attacks demonstrated in the physical world. We can taxonomize LiDAR spoofing attacks into two types based on the attack goals.

Object Injection Attacks

This type of attack is designed to inject ghost objects that do not actually exist. The relay attack [270] sends back recorded laser signals to the victim LiDAR. However, the impact of this attack in AD contexts is limited because it cannot inject objects closer than the attacker [290]. To address this limitation, synchronized injection attacks [306, 312, 185] have been designed to more effectively compromise AD vehicles in practical scenarios. This attack requires a *white-box* knowledge of the target LiDAR's deterministic scan pattern and its current state, i.e., where the LiDAR is currently scanning. With this *white-box* knowledge, the attacker can emit malicious lasers to overwrite legitimate scans based on the obtained pattern. However, synchronized injection attacks heavily rely on a deterministic LiDAR

scanning pattern; therefore, they can be easily foiled by laser scan timing randomization, which is a common feature in recent New-Gen LiDARs [290]. Considering the worst scenario in which some AD vehicles may use old-generation LiDARs, we evaluate the effectiveness of synchronized injection attacks in driving scenarios in §9.6 and §9.7.

Object Removal Attacks

These attacks aim to prevent object detectors from identifying actual objects. Synchronized removal attacks [125, 213] remove objects by moving all points of the object to a distance far away or within the area below the minimum distance threshold. However, as discussed in §9.2.1, these *white-box* attacks, which rely on synchronization, are ineffective against New-Gen LiDARs. Asynchronized (or *black-box*) removal attacks [270, 306, 290] do not depend on these assumptions. Particularly, the high-frequency removal (HFR) attack [290] has shown even higher effectiveness than synchronized attacks. The HFR attack made 5 sedan cars undetected and showed potential effectiveness against AD in a driving simulation. However, the HFR attack is not highly effective against LiDARs with pulse fingerprinting. This fact motivates us to design a new attack, A-HFR attack, in §9.4.

9.2.2 Pulse Fingerprinting for LiDAR

Pulse fingerprinting [366] is a technique that authenticates whether the reflected laser pulses are emitted by the LiDAR itself or by other LiDARs. While pulse fingerprinting is installed originally for anti-interference purposes to allow multiple LiDARs to operate at close distances, it also demonstrates a high defense capability against LiDAR spoofing attacks [290]. Pulse fingerprinting can be found in several New-Gen LiDARs such as Livox Mid-360 [23], Hesai XT32 [49], and AT128 [6]. While the detailed implementation of pulse fingerprinting is not publicly released, it is considered to be encoded in the interval between two consecutive

pulses; therefore, high-frequency pulses can accidentally match the interval and then bypass the authentication [290]. Naturally, higher frequency leads to higher chances of hitting the interval. However, it is not trivial to increase the frequency because a higher pulse frequency causes overheating of the laser diode and degrades the peak power of the laser, which must be higher than the peak power of the legitimate laser. To overcome this trade-off, our A-HFR attack utilizes gray-box knowledge of the LiDAR scan pattern to know when to cool down the diode (§9.4). We note that such a naive authentication design in New-Gen LiDARs should be an inevitable design rather than a random choice because more complex authentication requires higher laser power per time, which may harm human eyes. For example, 2 times more pulses for complex authentication will double the laser power per time. Moreover, the detection range of LiDAR will also be degraded if LiDAR uses more power for authentication.

9.2.3 Prior Attempts to Attack Moving Vehicles

So far, there has been no successful demonstration of LiDAR spoofing attacks on AD vehicles driving in realistic AD scenarios. As shown in Table 9.1.1, the majority of prior works do not consider attacks on moving targets. Several prior attempts [127, 125, 213] have targeted moving vehicles, but these works have the following three critical limitations to effectively attack driving AD vehicles: First, the LiDAR detection systems in prior work are not capable of operating at long ranges. Jin et al. [213] just manually aimed at the target LiDAR, and thus their approach cannot be applied to long-range scenarios since humans cannot even see a LiDAR from a far point, such as 100 meters away. Cao et al. [127, 125] detect the target LiDAR with a vision-based approach with YOLOv3 [282] trained to directly detect the target LiDAR. While this is a systematic approach, we find that the vision-based approach cannot handle long-range LiDAR detection even at 10 meters away since the LiDAR appears too small to detect in the captured image frame, as detailed in §9.5.1.

Second, we also find that prior work does not have mature spoofing attack devices that can automatically and precisely aim at a far-away target with attack lasers with sufficiently high power. The only prior attempts [127, 125] used a generic pan-tilt system [35] to control the laser emitter direction. However, these attack devices are unlikely to be effective against fast-moving vehicles at long distances since such generic pan-tilt systems are not originally designed for precise targeting of far and small objects (e.g., LiDAR), but only for coarse vision tracking of photo subjects. Finally, the majority of prior attacks that require synchronization with the target LiDAR cannot be effective against the recent New-Gen LiDARs since timing randomization makes synchronization virtually impossible [290]. HFR attack [290] remains effective even under timing randomization but is highly mitigated by pulse fingerprinting, and thus its end-to-end attack impact on AD vehicles has not been fully validated. To address these limitations, we design our MVS system that can practically deploy effective LiDAR spoofing attacks against driving AD vehicles, as detailed in §9.3.

9.2.4 Threat Model

We generally follow the same threat model adopted in prior work [130, 312, 185]. We particularly target the threat model called "Spoofer placed in environment" threat model [185], in which the attacker deploys LiDAR spoofing attacks against a driving AD vehicle from a roadside. We consider that this threat model is the most practical but challenging among the ones discussed in [185] since the other threat models assume that the attack is deployed from a front or side vehicle driving along with the victim vehicle to justify manual or ill-quality aiming. Fig. 9.1.1 illustrates the bird's-eye-view of our threat model. The victim vehicle driving at high speeds (e.g., 60 km/h) is under attack from $\geq$110 meters away. We assume that the attack site is a straight and flat road, and thus assume that the height of the target LiDAR is preknown. The attack device is placed at the roadside and starts attacking the victim from as far away as possible (e.g., $\geq$110 meters away).

9.2.5 Major Updates from Our Preprint Papers

From our preprint works [192, 314], this full paper has major updates in not only new design contributions enabling the longer-range attacks with the parabolic mirror antenna designs (§9.3.4) to receive and track the lasers from the LiDAR even at 110 m away, but also has more comprehensive evaluation including a quantitative comparison between the vision and IR-based detection methods (§9.5.1), driving evaluation at higher speeds such as 60 km/h (§9.6), injection attack evaluation against the high-speed vehicle (§9.6.4), A-HFR attack evaluation against the high-speed vehicle (§9.6.2), and end-to-end evaluation with the actual AD system (§9.7). Our preprint works focused on the initial testing of our attack devices and attack design validity at lower speeds.

9.3 MVS System Design

9.3.1 Overview

To practically deploy long-range LiDAR spoofing attacks against AD vehicles driving at high speeds, we design a Moving Vehicle Spoofing system (MVS system). Fig. 9.1.1 illustrates the overview of our MVS system consisting of three subsystems: (1) IR camera-based detection and tracking system, (2) auto-aiming system, and (3) LiDAR spoofing system. For the IR camera-based detection and tracking system, we design a novel methodology with an IR camera to accurately localize the target LiDAR even at longer ranges (e.g., 100 m) than the prior vision-based method, which can only detect a LiDAR up to 5 m (§9.5.1). For the auto-aiming system, we design a responsive and accurate auto-aiming system that can precisely aim at any horizontal angle with a high-precision servo motor. For the LiDAR spoofing system, we build a novel spoofer upon the existing LiDAR spoofers [130, 312, 127,

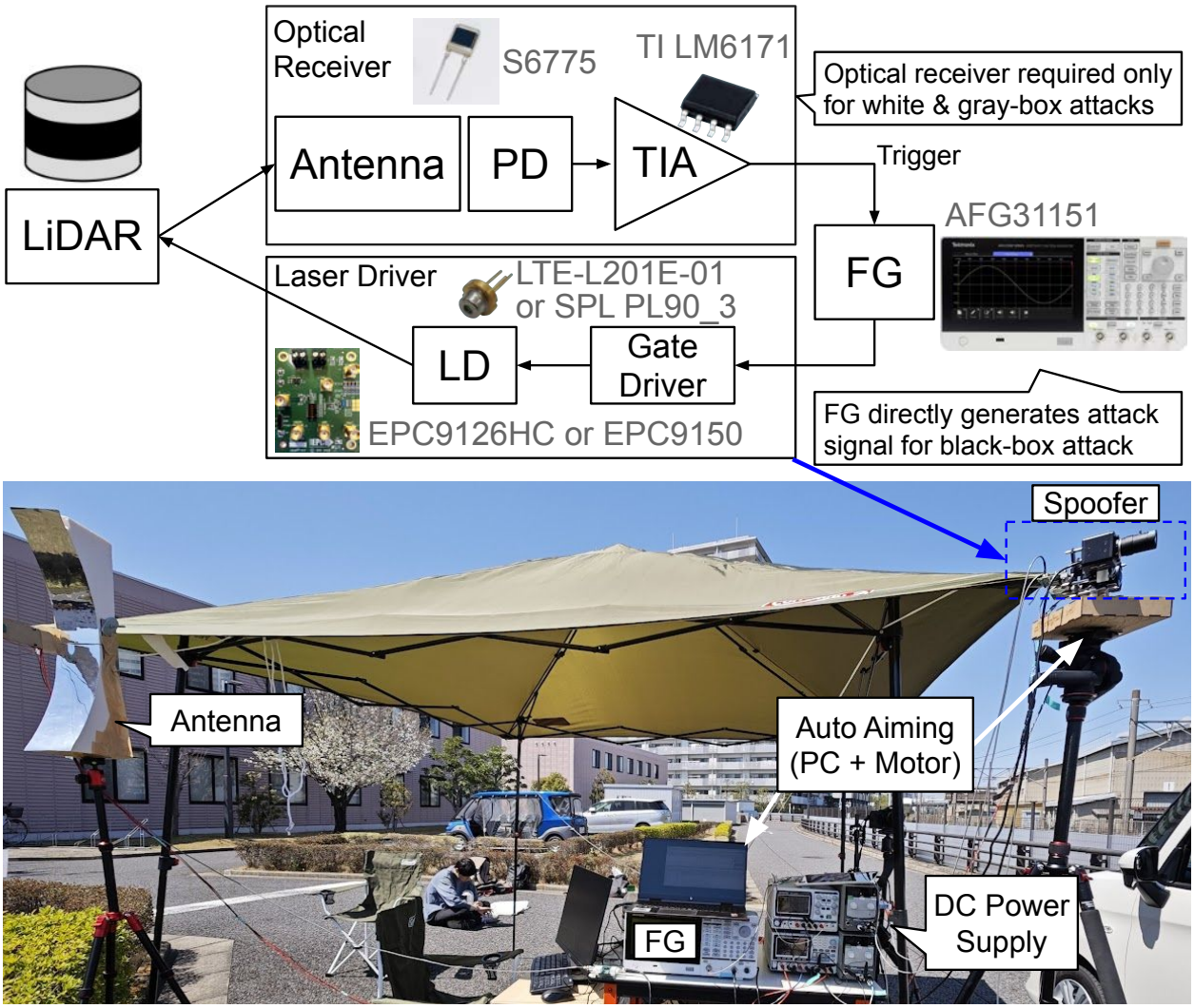


Figure 9.3.1: Overview of the hardware setup of our MVS system. The antenna is only used for white-box and gray-box attacks to obtain the current state of the target LiDAR.

125, 213, 290] to handle long-range attack scenarios. We further improved the electronics and optics as detailed in Fig. 9.3.1. Particularly, we install a parabolic mirror antenna before the (photodiode) PD to receive the laser from LiDAR even at long distances.

9.3.2 Infrared Camera-based Detection and Tracking System

Prior work [127, 125] utilizes a vision-based approach based on YOLOv3 [282] to detect the target LiDAR. However, as shown in Fig. 9.3.2 (left), the LiDAR appears too small in the camera frame for reliable detection at long distances (e.g., 15 m). A naive solution would

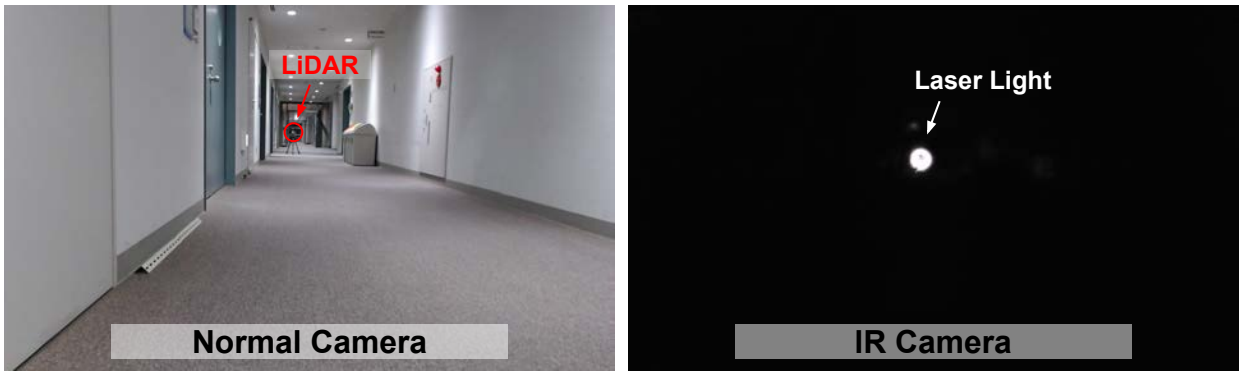


Figure 9.3.2: Comparison of vision and IR-camera frames at the same sight 15 m away from the target LiDAR. The IR camera only senses the light in the IR wavelength and thus enables accurate LiDAR-agnostic detection of the LiDAR location.

be to use an expensive camera and telephoto lens to get a higher resolution image, but this approach not only incurs high costs but will introduce challenges in real-time processing due to the high-resolution image and manipulation of a heavy-weight camera.

Target LiDAR Detection with IR Camera

To address this challenge, we design a simple yet novel detection system inspired by InfraRed Search and Track Systems (IRST Systems) designed for military applications, enabling long-distance detection and tracking of objects like enemy fighter jets [187]. By design, LiDAR must keep emitting IR lasers in all directions to obtain 3D point cloud data. Intuitively, we should be capable of achieving stable detecting and tracking of the IR laser source, similar to how an anti-aircraft missile operates. Fig.9.3.2 (right) shows the IR camera frame at the same location as the left figure. As the shutter time of the IR camera is significantly longer than that of the LiDAR scans, the IR camera can very likely capture the laser trace during the shutter time like the distinct white circle in the figure.

This approach can achieve three key advantages over prior vision-based systems: First, we can *directly* determine the location of the laser trace. As the laser and its reflection fly in a straight line, the origin of the laser emission must coincide with the location where the

reflection was sensed. Therefore, the attacker can directly target the source of the laser, as indicated by the white circle in Fig. 9.3.2 (right). On the contrary, the prior vision-based approach requires an additional step to determine the laser emission location even after detecting the bounding box of the target LiDAR. Furthermore, the detected bounding box itself has localization errors and jittering, especially for small object detection.

Secondly, our approach is *LiDAR model-agnostic* because this approach relies only on the fundamental property inherent to all LiDARs, which inevitably emits lasers into the scanning area. In §9.3.2, we demonstrate that our method can be universally applied to three LiDAR models without requiring additional training or adaptation. On the other hand, the prior vision-based approach requires a large number of various images of the target LiDARs to train a dedicated object detector. If the attacker wants to handle multiple scenarios (e.g., multiple LiDAR models), the number of required images could increase exponentially. Finally, our approach can be robust against different environmental conditions. For example, the prior vision-based approach is not generally capable of nighttime scenarios due to the dependence on passive visible lights from other sources. Our approach can handle such challenging environmental conditions because this approach merely relies on the active IR light emitted from the target LiDAR. The other wavelengths will be filtered out by the bandpass filter.

To eventually localize the LiDAR location in the IR camera frame, we design two approaches: DNN (Deep Neural Network)-based object detection and highest-intensity methods. For the DNN-based object detection, we train an object detector (e.g., YOLOv5 [215]) to detect the LiDAR laser-induced white circle. For the highest-intensity method, we simply select the pixel with the highest intensity in the frame. In §9.5.1, we find that the DNN-based detection shows superior robustness compared to the highest-intensity method. While the highest-intensity method works well in indoor environments, it does not consistently perform well in outdoor environments where the sunlight and its reflection often have the highest intensity. In §9.5.1, we confirm that our DNN-based method achieves significantly higher

detection accuracy than the prior vision-based approach while the vision-based detection can only detect a LiDAR at most 5 meters away. Our IR camera-based detection can LiDAR-agnostically detect different LiDARs from distances exceeding 20 meters with $\geq$90% success rates and can effectively handle the 110-meter outdoor attack scenarios in §9.6.

IR Camera Configurations: As illustrated in Fig. 9.A.1 in Appendix 9.A, we set up our IR camera based on a webcam of ELP-USBFHD08S-MFV [92] with 10 times optical zoom. We removed the pre-installed low-pass filter in front of the CMOS sensor to enable the IR wavelength sensing. We then placed a bandpass filter (FBH905-10 [93]) to selectively receive the IR wavelength (905 $\pm$ 5 nm), which is used in the majority of LiDARs [20].

Robust Tracking with Misdetected Frames

After the LiDAR is localized in each frame, the next step is to track the LiDAR with prior detection results and maintain the LiDAR location for the frame with failed or outlier detection. We note that tracking is particularly beneficial for our IR camera-based detection since we cannot always sense the laser from the target LiDAR. For example, VLP-16 [48] scans each point at around 10 Hz, meaning we can at most detect the LiDAR location every 0.1 seconds. To handle this, we implement two methods: the Kalman filter [218] and DNN-based tracking. Kalman filter is one of the most widely used methods for tracking purposes. Kalman filter estimates the state of a linear dynamic system from a series of noisy measurements. We implement the Kalman filter with outlier elimination based on the Mahalanobis distance [136]. For the DNN-based tracking, we simply feed the current and a few prior frames as the channel of the input image of the DNN-based detection. We denote the number of frames to feed as N_p. In §9.5.1, we find that both methods generally achieve high tracking performance and that the combination of the two methods shows the highest performance. Particularly, the DNN-based tracking with N_p=3 shows high robustness against the ghosting effect [97] caused by strong lasers from the LiDAR.

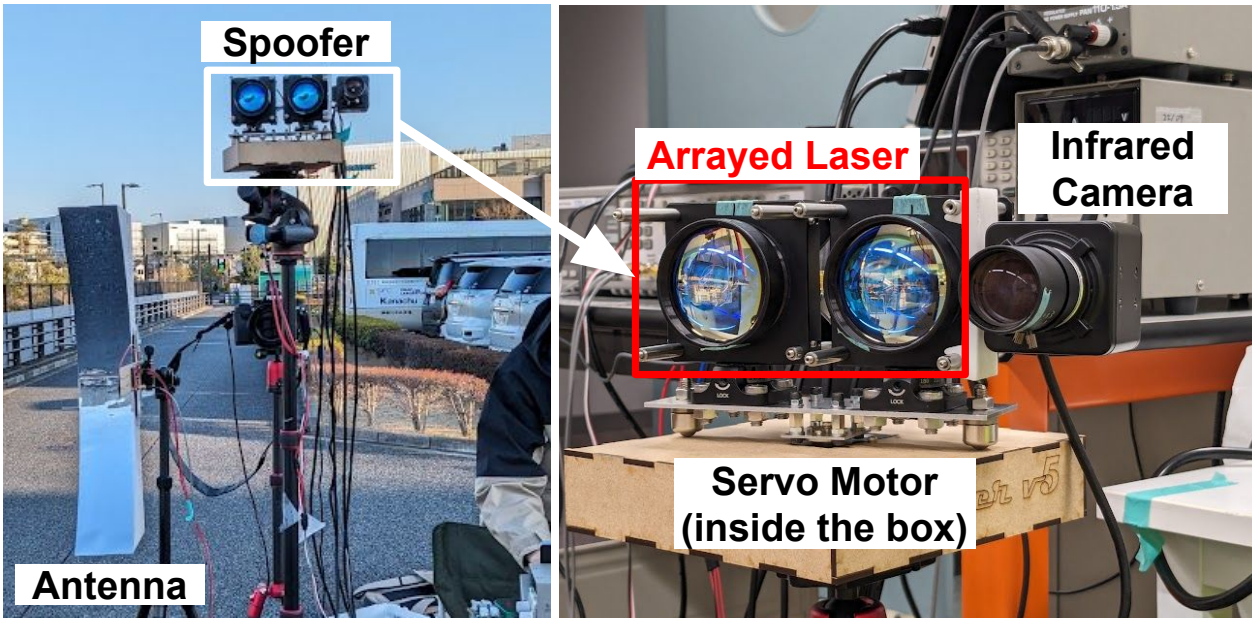


Figure 9.3.3: Overview of our spoofing system. The antenna is only used for white-box and gray-box attacks to know the current state of the target LiDAR.

9.3.3 Auto-Aiming System

After the current position of the target LiDAR is estimated, the auto-aiming system calculates the required rotation angle to aim at the LiDAR. We assume the attack site is on a flat road and that the height of the LiDAR is known and constant, as discussed in our threat model (§9.2.4). Therefore, we first set our spoofer at the same height as the target LiDAR and the auto-aiming system only adjusts the horizontal angle on the fly. To accurately aim at the target horizontal angle, we find that the precision of the servo motor has a major impact on attack success because a 1° error leads to around 1.7 m error at 100 m away. We eventually selected a high-precision servo motor, Dynamixel MX-28, which has 0.088° angle resolution with PID control. To smoothly rotate the entire MVS system, we enclose the motor on a box and the system rotates on the box with two bearings as shown in Fig. 9.3.3.

9.3.4 LiDAR Spoofing System

Table 9.3.1 lists the comparison between our MVS system and prior setups to attack moving targets. Our MVS system has two major improvements over prior works: (1) We adopt two arrayed lasers with 2-inch lenses to cover a significantly larger (100 times larger) area than prior work. The two lenses are horizontally arrayed. While we could place more arrayed lasers to cover a larger beam area, doing so could harm the responsiveness of the auto-aiming system due to the increased weight, and it also introduces additional costs. In §9.6.3, we find that the setup with the two 2-inch lenses can achieve sufficient performance to attack a vehicle driving at 60 km/h. (2) We introduce a vertical parabolic mirror antenna before the photodiode (PD) to reliably receive the laser from the LiDAR. Prior work has used a bare PD, but it cannot work in long-range attack scenarios because the lasers from LiDAR spread radially and sparsely at long distances, and thus the bare PD is less likely to receive them. The parabolic shape can collect signals arriving from any direction at a single point (focal point). Fig. 9.3.4 shows how the vertical parabolic mirror antenna helps the PD receive lasers from LiDAR at long distances. The parabolic mirror surface reflects incoming lasers and concentrates them at the focal point where the PD is located. While we can also gather such sparse lasers by using a convex lens, the parabolic antenna has clear advantages in the lightweight design and cost efficiency as evidenced by their widespread use in space exploration applications [96].

For our threat model, we design a one-dimensional vertical parabolic mirror as shown in Fig. 9.3.3. We use a focal length $f = 200$ mm and an antenna height $h = 600$ mm. For the horizontal dimension, we adopt a thick design, consisting of a stack of parabolic shapes, because the strong directivity of the parabola is not suitable for the horizontal dimension where the target vehicle moves quickly. This design allows us to eliminate the need to move the antenna during the attack. Once the MVS system receives the laser from LiDAR with the antenna, we can launch not only white-box attacks [306, 312, 185] by synchronizing the

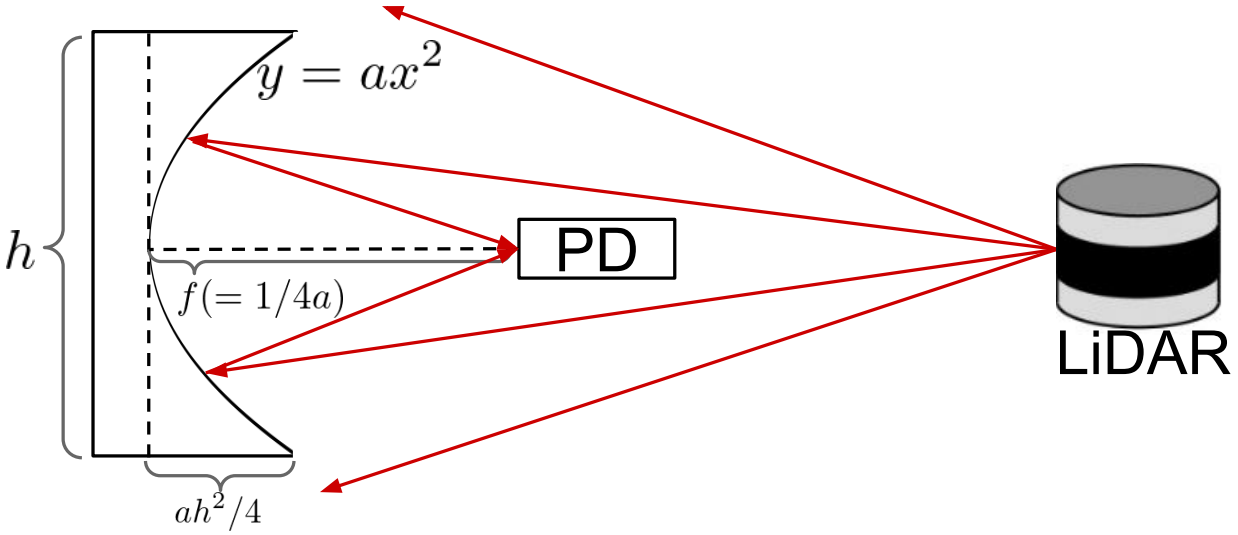


Figure 9.3.4: Illustration of how the vertical parabolic mirror antenna helps the PD receive more lasers from LiDAR.

LiDAR scan pattern, but also our gray-box A-HFR attack, which will be introduced in §9.4. For the LiDAR spoofing methodologies themselves, we follow prior work other than our A-HFR attack since our major contributions of the MVS system are in how to deploy LiDAR spoofing attacks in high-speed and long-range scenarios.

9.3.5 Cost of the MVS system

Fig. 9.3.1 illustrates the entire components of our MVS system, which is built completely using off-the-shelf components. The system has a total cost of $2.3k, including the $1.1k laser driver, $36 optical receiver, $700 detection and auto-aiming systems, and $500 in miscellaneous parts including jigs. Specifically, the laser driver component includes two laser boards ($450 × 2), two 2-inch lenses ($54 × 2), and two laser diodes (LD) ($36 x 2). The optical receiver consists of a PD ($2), a TIA ($4), and an antenna ($30). The detection and auto-aiming systems include the IR camera ($70), the bandpass filter ($165), the servo motor ($270), and the box with a turning table ($200). Additionally, to run the MVS system, we use a function generator (FG) ($6.7k), 4-channel DC power supplies ($3k), and a laptop ($1.7k). We note that the majority of the cost is due to the expensive FG, which

Table 9.3.1. Comparison between our MVS system and prior setups to attack moving targets.

	Number of Lasers	Beam Diameter	Total Beam Area	Maximum Tracking Distance	Tracking	Detection Strategy (target, using device)	LiDAR Agnostic	Vertical Antenna
Ours	2	60 cm	$5654.7\,cm^2$	110 m	Auto	w/ IR Camera	Yes	Yes
Cao et al. [127, 125]	1	2.54 cm	$5.1\,cm^2$	5 m	Auto	w/ RGB Camera	No	No
Jin et al. [213]	1	8cm	$50.3\,cm^2$	15 m	Manual	-	-	-

can be replaced with Analog Discovery ($400) [57] or FPGAs ($200) with a certain level of engineering efforts. As this study is motivated to explore the feasibility of LiDAR spoofing against moving vehicles, we used the FG to flexibly evaluate various parameters and setups with less effort.

9.4 New Attack: A-HFR Attack

To bypass pulse fingerprinting in recent New-Gen LiDARs, we designed a new removal attack, named Adaptive HFR (A-HFR) attack. We discover that the pulse fingerprinting can be bypassed if we can achieve much higher frequency pulses than the ones of the prior HFR attack [290]. We achieve 25 times higher frequency than the HFR attack at the target attack range by utilizing the *gray-box* knowledge of the scan pattern of target LiDAR, to avoid the overheating issue of the laser diode.

9.4.1 Basic Concept to Bypass Pulse Fingerprinting

Prior work [290] points out that the pulse fingerprinting in recent New-Gen LiDARs presumably uses a pair of pulses to measure a point and embeds the fingerprint into the interval of a pair of pulses. The pulse intervals of each pair randomly vary and the LiDAR can authenticate the reflected laser based on whether the interval matches the one the LiDAR emitted or not. The HFR [290] attack demonstrated that it can still bypass pulse fingerprinting LiDARs with 2.1% attack success rate and hypothesizes that their high-frequency pulses can occasionally match the correct interval and bypass the authentication as described in

Fig. 9.4.1. Another possibility is that the overheating may differ the laser wavelength to out of LiDAR's acceptable wavelength. Based on the datasheet [67], the effect is very small (3 nm per 10°C) and thus negligible.

The hypothesis assumes the existence of T_α, which is a tolerance error time used to authenticate the fingerprinting. With T_α, the attack success rate of the HFR attack in an ideal case can be written by $\min\left(1, \frac{T_\alpha}{T_\mathrm{A}}\right)$, where T_A is the interval of attack pulses. In other words, the attack always succeeds if the interval of attack pulses is less than the tolerance error, and stochastically succeeds otherwise.

If the hypothesis is correct, a higher pulse frequency should result in higher attack effectiveness. As shown in Fig. 9.5.3, we confirmed that the hypothesis is correct through the experiment, i.e., a sufficiently high pulse frequency can bypass the fingerprinting in recent New-Gen LiDARs. The remaining challenge lies in achieving such a high pulse frequency. However, we find that it is not trivial to increase the frequency due to a trade-off between pulse frequency and peak power. Specifically, emitting a pulsed laser at a higher frequency causes overheating of the laser driver and the laser itself, significantly degrading the laser's peak power. To overpower the legitimate laser power emitted by the target LiDAR, the peak power of the attack laser must be greater than that of the legitimate laser. The HFR attack achieves the highest attack success rate at around 1 MHz when attacking LiDARs without pulse fingerprinting. Ideally, attacking with higher frequencies should increase the effectiveness of HFR, but overheating causes a reduction in laser power, significantly lowering the attack success rate.

9.4.2 Attack Design

To overcome the trade-off between pulse frequency and peak power, we design the adaptive-HFR (A-HFR) attack. As the name implies, the A-HFR attack adaptively changes its attack

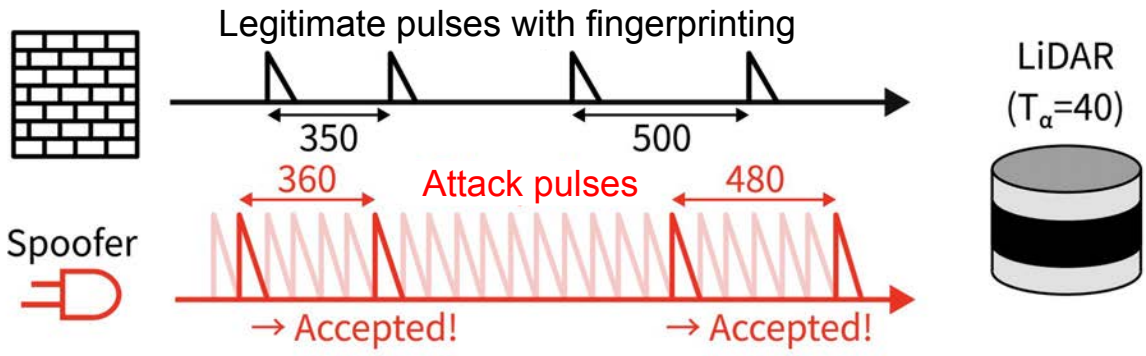


Figure 9.4.1: Requirements to bypass fingerprinting: (1) The interval of the attack pulses (T_A) should be short enough and close to the tolerance error (T_α); (2) the peak power of the attack laser must be higher than the legitimate laser of LiDAR.

laser frequency to avoid overheating. As shown in Fig. 9.4.2, the A-HFR attack strategically boosts the frequency only when the target LiDAR scans the specific object the attacker wants to hide. This design allows the diode to rest most of the time and effectively cool down during the rest. To know when the LiDAR scans the target objects, we utilize a photodiode (PD) similar to the existing *white-box* LiDAR spoofing attacks that require synchronization with the target LiDAR. However, unlike white-box attacks that require precise knowledge of when the LiDAR scans each point and a predictable scan pattern, A-HFR only needs a coarse-grained understanding of the victim LiDAR's state. Due to this reduced information requirement compared to white-box synchronization attacks, we classify the A-HFR attack as a *gray-box* attack and term the coarse-grained requirement as *weak synchronization*.

Obtaining Gray-box Knowledge

We can roughly estimate when the LiDAR scans each horizontal angle based on the laser from the LiDAR with PD. As the LiDAR is rotating, the interval between two consecutive lasers from the LiDAR corresponds to the time required to scan the full 360°. Since LiDAR scans each angle evenly, we can calculate the scan timing of each relative angle between the PD and the target object. Although the relative angle will change as the victim AD vehicle moves, we can account for this by using a wider margin in the attack angle. In §9.5.2, we

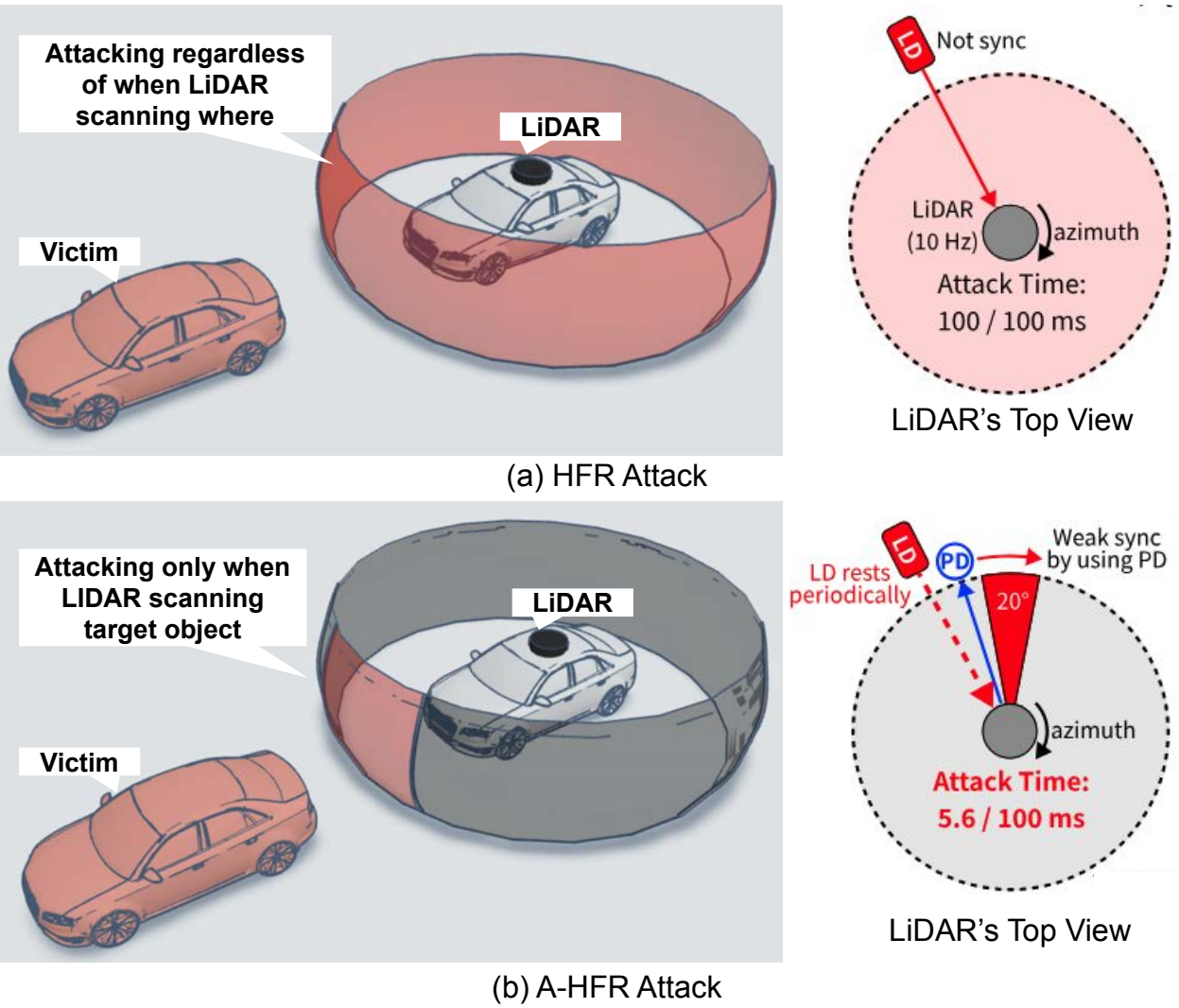


Figure 9.4.2: Comparison between (a) HFR attack and (b) A-HFR attack. A-HFR can keep high peak power at high effective frequencies by limiting the attack angle with weak synchronization to know where to boost the frequencies.

evaluate the maximum attack angle that can maintain attack effectiveness. Similarly, we can also rest the attack when the LiDAR is scanning less important vertical angles. For example, pedestrians and other vehicles generally do not exist at a height $\geq$3 meters. If we can focus only on the area within 20° of horizontal and vertical angles, the laser diode can rest 97% of the total LiDAR scan time. In Appendix 9.B, we evaluate the impact of limiting vertical and horizontal attack angles and find that limiting both vertical and horizontal attack angles is necessary to achieve more than 25 MHz frequency.

9.5 Attack Capability Evaluation

We first evaluate the attack capability of our MVS system and the A-HFR attack through static indoor and outdoor experiments to validate our proposed designs.

9.5.1 MVS System Evaluation

Of the three subsystems of the MVS system, we begin with an evaluation of the IR camera-based detection and tracking system because the others require actual high-speed driving scenarios to adequately evaluate their capabilities, which will be covered in §9.6 and §9.7.

Detection Capability Evaluation – Vision v.s. IR Camera

We evaluate the detection capability of our IR camera-based target LiDAR detection (§9.3.2) through the comparison with the prior state-of-the-art vision-based detection [127, 125]. We calculate the detection success rate of each method at 4 different distances between the target LiDARs and the vision or IR camera. We note this is an indoor static experiment, i.e., both the LiDAR and camera remain stationary. For the vision-based detection, we train YOLOv5 [215] with 200 images of the target VLP-32c [46] LiDAR. For our IR camera-based detection, we train YOLOv5 with 532 images of the attack traces like those shown in Fig. 9.3.2 (right). The resolution of all vision and IR images is 640x640. Since the IR camera cannot always sense the emitted laser from the target LiDAR, we define the attack success rate as whether a detector can correctly output a point within the area of the LiDAR with the previous 10 camera frames. The camera operates at 60 Hz, meaning 10 frames span 0.17 seconds. For both vision- and IR camera-based detection, we finally output the averaged LiDAR position of the frames excluding the frames that failed to detect the target LiDAR.

Table 9.5.1. Detection success rates of vision- and IR camera-based methods at different distances between the target LiDAR and the cameras.

	LiDAR	5 m	10 m	15 m	20 m
Vision-based Detection	VLP-32c	**100%**	0%	0%	0%
IR Camera-based Detection (Ours)	VLP-32c	**100%**	**100%**	**100%**	**100%**
	Horizon	**100%**	**100%**	**100%**	**100%**
	AT128	**90%**	**90%**	**100%**	**90%**

Results: Table 9.5.1 lists the detection success rates of vision- and IR camera-based methods at different distances between the target LiDAR and the cameras. As listed, the vision-based detection fails to detect the target LiDARs even at 10 m away from the camera. On the other hand, our IR camera-based detection achieves $\geq$90% detection rates for 3 different LiDARs. Among the 3 LiDARs, AT128 [6] proves slightly more difficult to stably receive its laser by the IR camera. We suspect that AT128 has some irregular scan patterns for the vertical angles. Nevertheless, $\geq$90% detection rates are already sufficiently high, and we find that the remaining 10% of frames should be complementable by the tracking system as evaluated in the next section.

Tracking Capability Evaluation

We then evaluate the tracking performance for each combination of the detection and tracking methods. Since a dynamic experimental setup is required to evaluate the performance of tracking, we record a video as shown in Fig. 9.5.1 where we put a target LiDAR (AT128) on a push cart and manually push it toward an IR camera from 40 meters away at around 8 km/h speed. We apply each method for this video and calculate the tracking success rate defined by the ratio of whether the tracking point is within the LiDAR area or not.

Results: Table 9.5.2 lists the tracking success rates of different combinations of detection and tracking methods for each 2-meter interval bin for AT128 LiDAR. As listed, YOLOv5 with $N_p = 3$ achieves the highest tracking rates regardless of the existence of the Kalman filter. This result indicates that the DNN-based tracking is highly effective in our threat

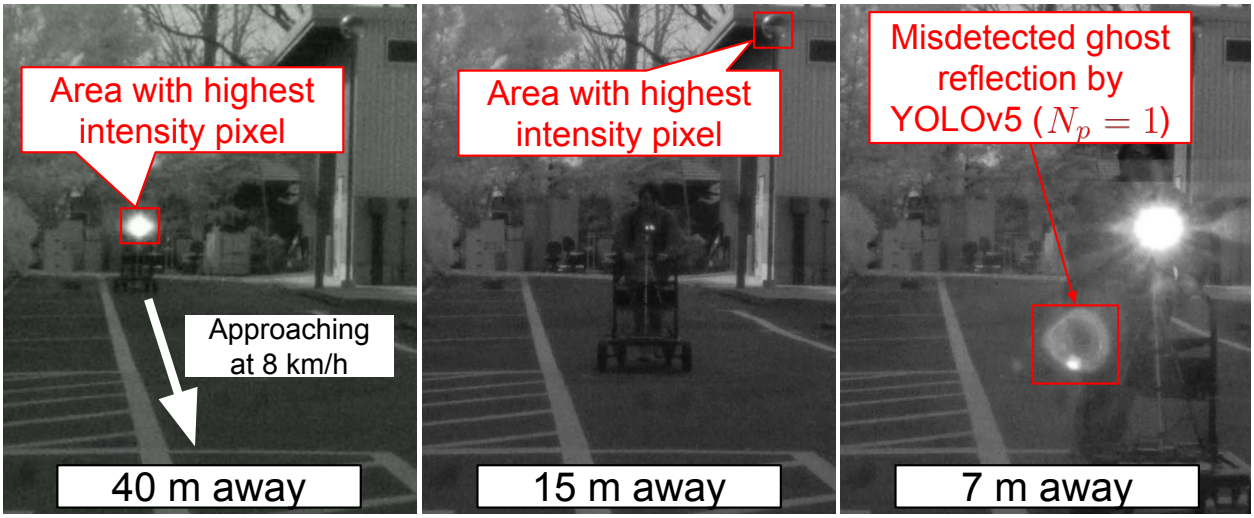


Figure 9.5.1: IR camera frames at 40, 15, and 7 meters away from the target LiDAR. The highest intensity pixel in the 15-meter frame is the reflection of the sunlight on a street light. The 7-meter frame has ghosting mistedected by YOLOv5 with N_p=1.

model. As shown in Fig. 9.5.1, the highest-intensity methods and YOLOv5 with $N_p = 1$ cannot adequately handle several cases. For example, at a distance of 15 meters away, the pixel with the highest intensity is not the LiDAR but rather a street light reflecting sunlight into the IR camera. Similarly, at 7 meters away, YOLOv5 with $N_p = 1$ misdirects the ghost reflection [97] caused by strong direct light from the LiDAR entering the lens. Since such strong direct light does not occur in consecutive frames, YOLOv5 with $N_p = 3$ appears unaffected by ghost reflections. For these methods, the Kalman filter sometimes helps to correct the LiDAR localization but not always since the Kalman filter also sometimes causes the long-last effect of misdetections at the current frame for the following frames. Although other state-of-the-art tracking methods (e.g, extended Kalman filter [183], the unscented Kalman filter [53], and particle filter [156]) may overcome the issue, we adopt the tracking method with YOLOv5 ($N_p = 3$) and the Kalman filter in the later experiments since it already has almost 100% accuracy until 6 meters away, which is much shorter than the braking distances (20 meters) at 60 km/h.

Table 9.5.2. Tracking success rates of different combinations of detection and tracking methods for AT128 LiDAR.

	Distance (m)								
	2-4	4-6	6-8	8-10	10-12	12-14	14-16	16-40	Avg.
Highest Intensity	**100%**	**100%**	91%	91%	75%	63%	73%	58%	68%
+ Kalman Filter	0%	0%	91%	**100%**	**100%**	**100%**	**100%**	**100%**	89%
YOLOv5 ($N_p = 1$)	77%	0%	14%	**100%**	**100%**	**100%**	**100%**	**100%**	89%
+ Kalman Filter	0%	0%	95%	**100%**	**100%**	**100%**	**100%**	**100%**	89%
YOLOv5 ($N_p = 3$)	55%	70%	**100%**	**100%**	**100%**	**100%**	**100%**	98%	**95%**
+ Kalman Filter	55%	70%	**100%**	**100%**	**100%**	**100%**	**100%**	98%	**95%**
# of frames	22	20	21	23	20	16	15	145	

Table 9.5.3. Three production LiDARs with pulse fingerprinting functionalities based on our measurements or official documentation. n/a means that we could not measure it.

	Livox Mid-360[23]	Hesai XT32[49]	Hesai AT128[6]
			
Scanning Type	Prism Rotating	Rotating	Mirror Rotating
T_{min}	1250 ns	250 ns	n/a
T_{max}	1550 ns	450 ns	n/a

9.5.2 A-HFR Attack Evaluation

We evaluate the attack effectiveness and robustness of the A-HFR attack against LiDARs with pulse fingerprinting by comparing the attack results of the HFR attack, which is the current state-of-the-art removal attack.

Evaluation LiDARs with Pulse Fingerprinting

We identify three mass-produced LiDARs with pulse fingerprinting as listed in Table 9.5.3. For XT32 [49] and Mid-360 [23], we find that the two LiDARs emit a pair of lasers for a single distance measurement while varying the pulse interval in the pair between the minimum (T_{min}) and maximum (T_{max}) pulse intervals specific to each LiDAR. For AT128 [6], we confirm the fingerprinting features through their official document, although we could not specify T_{min} and T_{max}. We mainly evaluate the A-HFR attack against XT32 and AT128 because the fingerprinting in Mid-360 [23] does not have sufficient complexity and can be bypassed by the HFR attack alone, as discussed in Appendix 9.C.

Attack Effectiveness

Fig. 9.5.2 shows the point removal rates for HFR and A-HFR attacks at different frequencies. We evaluate the ratio of removed points within the angle of 20° (v=50%) horizontally and 16° vertically based on our preliminary analysis showing that ≥95% of objects located more than 6 meters away in the KITTI dataset [174] fit within this area. We placed the spoofer 2 meters away from the LiDAR and 3 meters away from the target room wall. For XT32, we restrict the vertical attack angle to half (16°) to have more rest. As shown, our A-HFR attack can achieve significantly higher attack success rates compared to the current state-of-the-art HFR attack. For AT128 and XT32, due to the overheating issue, the HFR attack can achieve a maximum success rate of only 20% at around 10 MHz, and the attack success rate even starts dropping around 17 MHz. In contrast, the A-HFR attack can remove 100% of the points for AT128 at 15 MHz for AT128 ($T_\alpha = 67$ ns) and 96% of the points at 24 MHz for XT32 ($T_\alpha = 42$ ns).

Fig. 9.5.3 demonstrates the effectiveness of the HFR and A-HFR attacks against AT128. For the HFR attack, the majority of the points still remain, although we can see some points have been removed and the shape appears blurred. With the A-HFR attack, almost all points of the person have been completely removed. We note that this attack can be sustained for a long time, such as >100 seconds.

Robustness to Wide Attack Horizontal Angles

We evaluate the robustness of the A-HFR attack when the attack angle is increased. Although horizontal 20° can cover most road objects, wider attack angles can help compensate for spoofing errors and sudden movements of the target vehicle. Table 9.5.4 lists the point removal rates of the HFR attacks at different attack horizontal ranges for AT128 and XT32. As listed, the point removal rate decreases as the attack angle increases, especially for XT32,

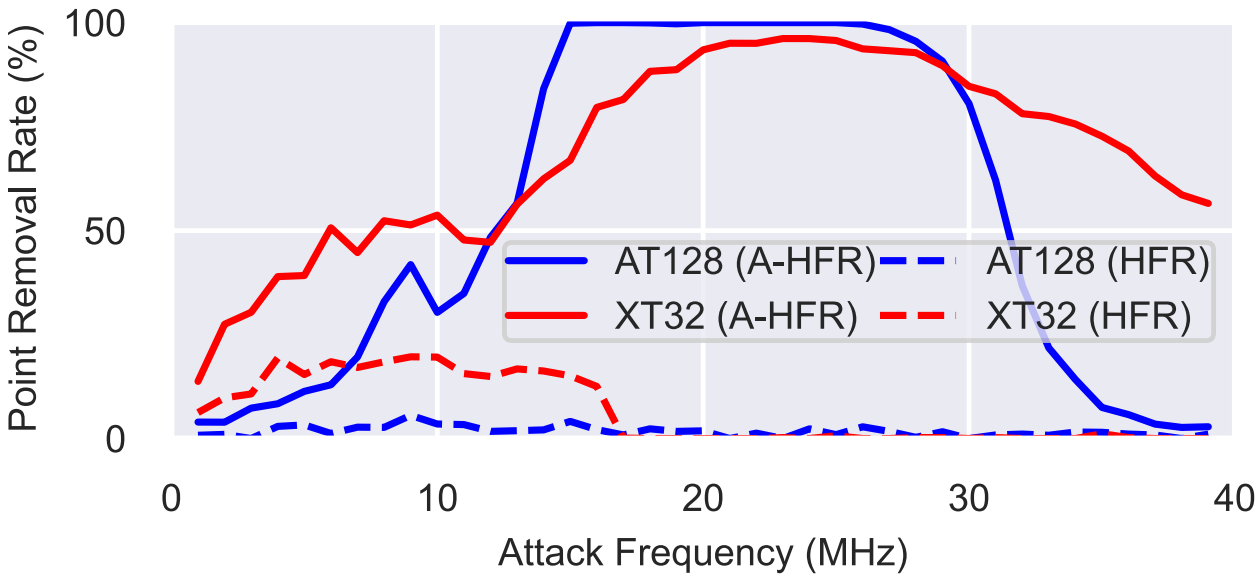


Figure 9.5.2: Point removal rates of the HFR and A-HFR attacks at different frequencies for LiDARs with pulse fingerprinting.

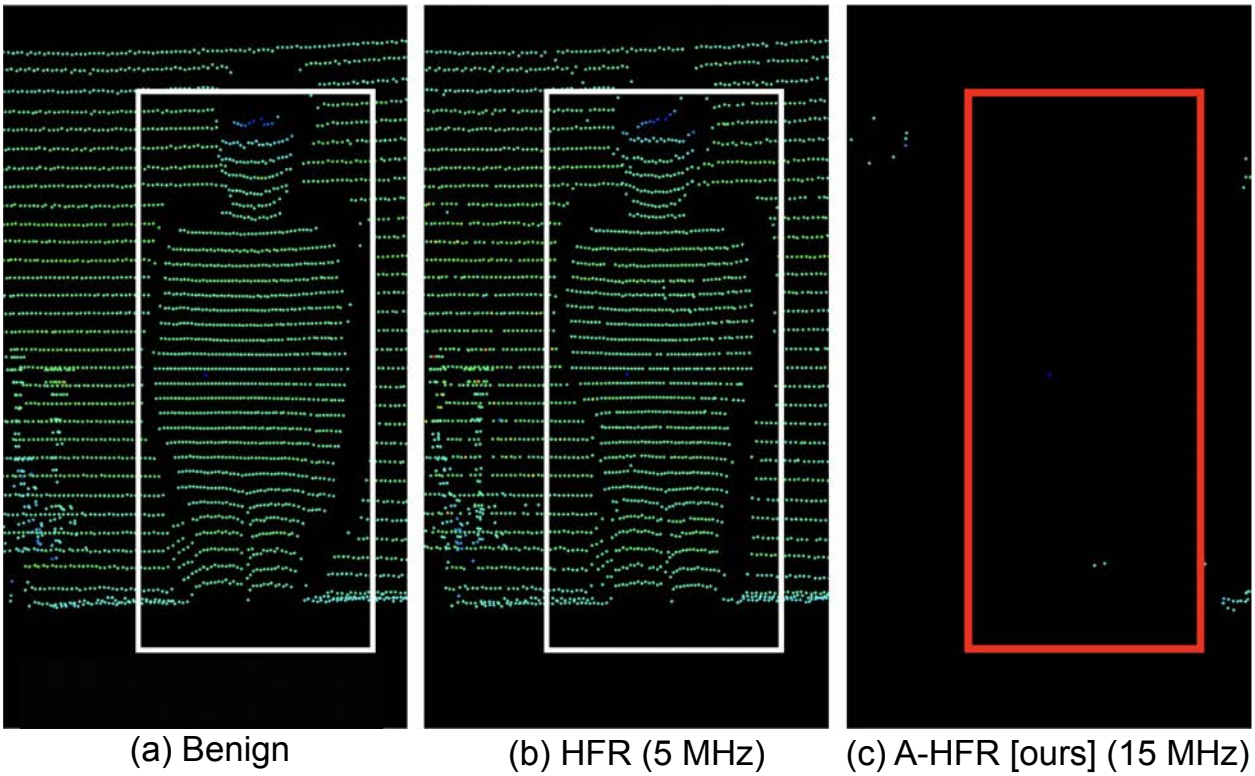


Figure 9.5.3: Comparison of the results of HFR and A-HFR attacks on AT128. A-HFR attack can completely remove almost all points (right).

which requires a higher frequency to attack as shown in §9.5.2. As a higher frequency leads to greater overheating of the diode, XT32 is less robust against wider attack angles than AT128 since the A-HFR attack is also affected by the overheating issue, at wider attack

Table 9.5.4. Point removal rates of the A-HFR attack with different attack horizontal ranges LiDARs with pulse fingerprinting.

	Attack Freq.	10°	20°	30°	40°	50°	60°
AT128	15 MHz	97%	100%	99%	98%	90%	90%
XT32	24 MHz	97%	96%	78%	58%	47%	38%

ranges. Nevertheless, for both LiDARs, the A-HFR attack can cover at least 20°, which is wide enough to hide at least one pedestrian.

Outdoor Evaluation

To further evaluate the effectiveness of the A-HFR attack, we conducted an outdoor experiment aiming to remove an entire car point cloud on XT32 LiDAR. We place the target car at a distance of 12 meters away. The car occupies around the area within 10° horizontally and 8° vertically in the LiDAR point cloud. As shown in Fig. 9.5.4, the target vehicle is entirely removed from the point cloud with a 98% success rate by the PointPillars of Apollo 6.0 [7] in 15 seconds. The several failures are due to the failure in weak synchronization to receive the laser from the LiDAR. Currently, we trigger the attack only when the laser correctly is received. We consider that this issue could be addressed by a speculative execution. Detailed discussions are in §9.8.

9.6 High-Speed Driving Evaluation with Real Car

We evaluate the attack feasibility and performance of LiDAR spoofing attacks with our MVS system against a real car driving at high speeds (e.g., 60 km/h). For removal attacks, we assess the effectiveness of HFR and A-HFR attacks. For injection attacks, we evaluate the synchronized injection attack.

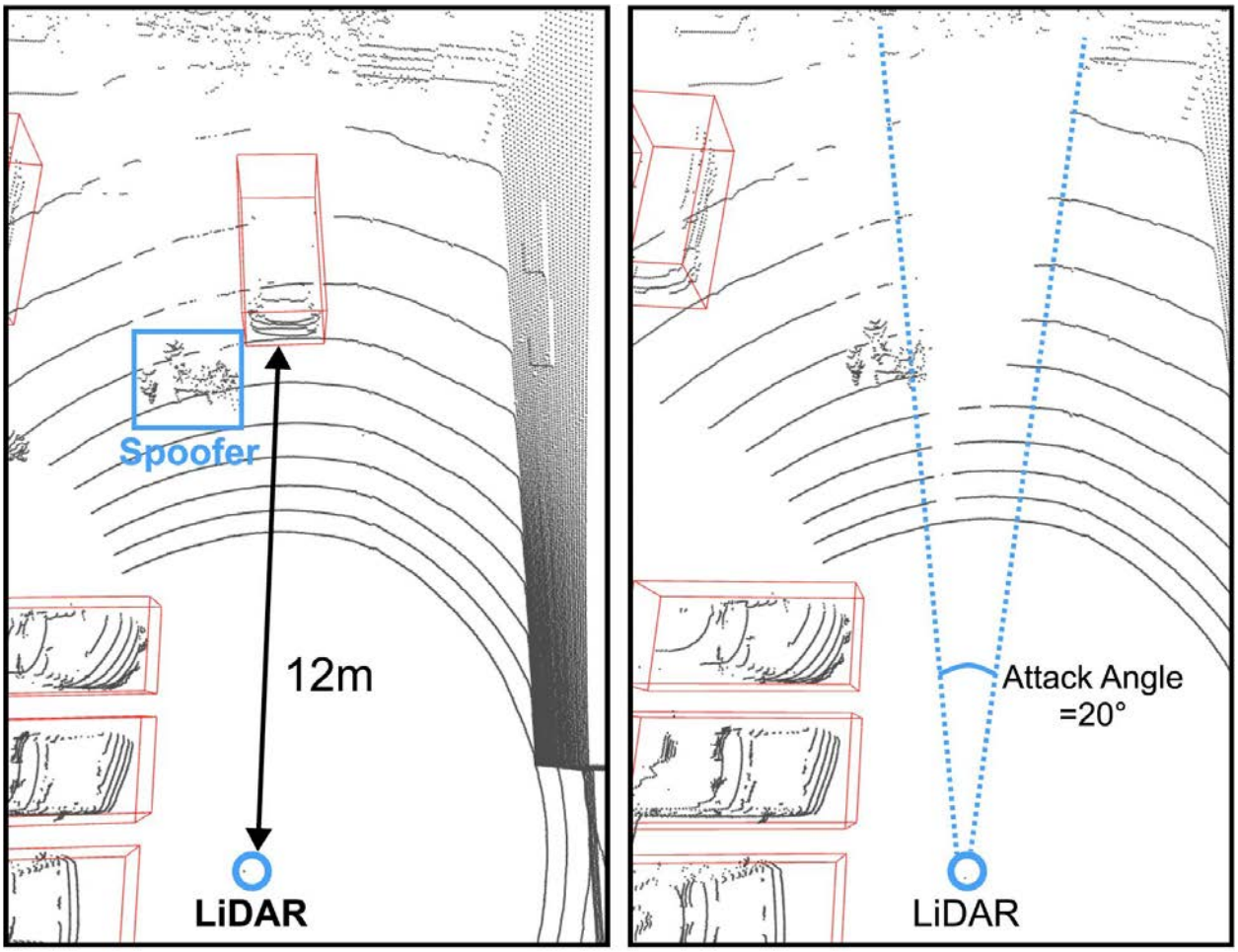


Figure 9.5.4: A-HFR attack on Hesai XT32 to hide a real vehicle. We limit the attack angle to 20° horizontally and 16° vertically. The red bounding box shows the detection results generated by Pointpillars in Apollo. When under attack, the vehicle becomes undetected with a 98% success rate for over 15 seconds.

9.6.1 Experimental Setup

Fig. 9.6.1 illustrates the overview of our experimental setup. We follow our threat model described in §9.2.4: We rent a private testing road with exclusive-use permission to ensure a controlled environment. We placed the MVS system device 2 m away from the driving lane and positioned the victim pedestrian 10 m away from the MVS system device. To prioritize safety, the victim pedestrian is also 2 m away from the driving lane, not directly in the lane. The target vehicle with a mounted LiDAR on its roof is approaching from 100 m away from the victim pedestrian, i.e., the attack distance between the MVS system and

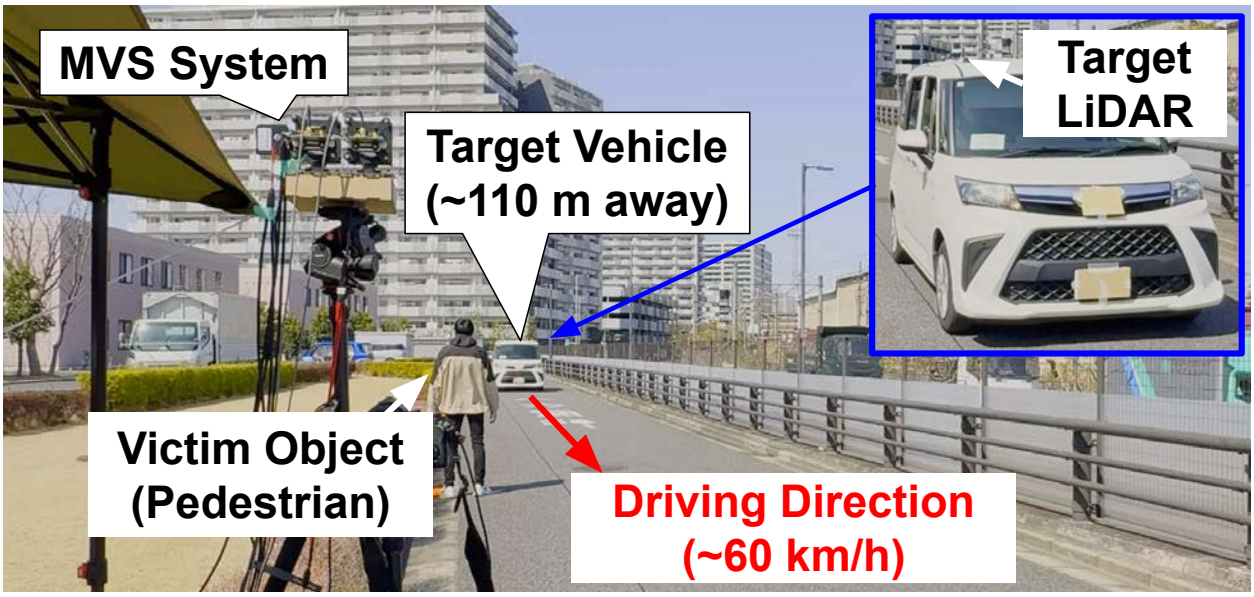


Figure 9.6.1: Experimental setup of high-speed driving evaluation with real car driving at ≤60 km/h. Our MVS system starts deploying LiDAR spoofing attacks around 110 meters away.

the target vehicle is up to 110 m. We measure the attack effectiveness starting from 70 m away from the victim, where the target vehicle reaches the desired speed. We evaluate the system at 6 different speeds from 10 km/h to 60 km/h. For each speed, we collected data from 4 different traces. All evaluation metrics are averaged over the 4 traces.

9.6.2 Removal Attack Evaluation

HFR Attack at High Speed

We evaluate the HFR attack against a high-speed driving vehicle with Livox Horizon [22], which is one of the New-Gen LiDARs whose manufacturer releases an official object detector, Livox Detection v2.0 [21]. We selected 0.25 as the confidence threshold for Livox Detection v2.0, ensuring that around 90% detection rate is achieved in benign scenarios.

Table 9.6.1 lists the attack success rates of the HFR attack for different speeds and the detection rates in the benign traces. As listed, the HFR attack with our MVS system can

Table 9.6.1. Attack success rates of HFR attack against Livox Horizon at different speeds. The "benign" row lists the detection success rates of the target pedestrian without attacks.

	Distance between victim and target (m)						
	0-10	10-20	20*-30	30-40	40-50	50-60	60-70
10 km/h	59%	83%	97%	**100%**	**100%**	**100%**	**100%**
20 km/h	65%	76%	99%	**100%**	**100%**	**100%**	**100%**
30 km/h	67%	78%	**100%**	**100%**	**100%**	**100%**	**100%**
40 km/h	71%	76%	**100%**	**100%**	**100%**	**100%**	**100%**
50 km/h	**93%**	**97%**	**100%**	**100%**	**100%**	**100%**	**100%**
60 km/h	85%	85%	96%	**100%**	**100%**	**100%**	**100%**
Benign	50%	100%	100%	100%	89%	94%	76%

*: the braking distance at 60 km/h

hide the victim pedestrian with almost 100% attack success rates until the vehicle is 20 meters away from the pedestrian. We do not observe evident performance degradation due to the higher vehicle speeds, and higher speeds actually yield even greater attack success rates. This result indicates that our MVS system has sufficiently high targeting precision to effectively aim at a fast-driving vehicle.

Table 9.6.2 lists the point removal rates of the HFR attack. We calculate the rate by dividing the number of points belonging to the victim pedestrian by the number of points in the corresponding benign frames at the same location. We manually annotate the area of the victim for all frames. As listed, the HFR attack succeeds in removing the majority of the victim's points, but it appears that $\geq$95% of the points need to be removed to completely fool the object detector. Fig. 9.6.2 shows examples of the LiDAR point clouds for both benign and attack scenarios at 10 and 40 meters away from the victim pedestrian. As shown, the majority of the victim's points have been removed by the HFR attack.

Notably, the attack success until 20 meters away has a significant impact on the safety implications considering that the braking distance at 60 km/h is 20 meters without the reaction distance [39]. To more rigorously estimate the impact on safety, we will conduct a closed-loop end-to-end evaluation with a popular AD stack in §9.7. We also discuss possible countermeasures in §9.8.

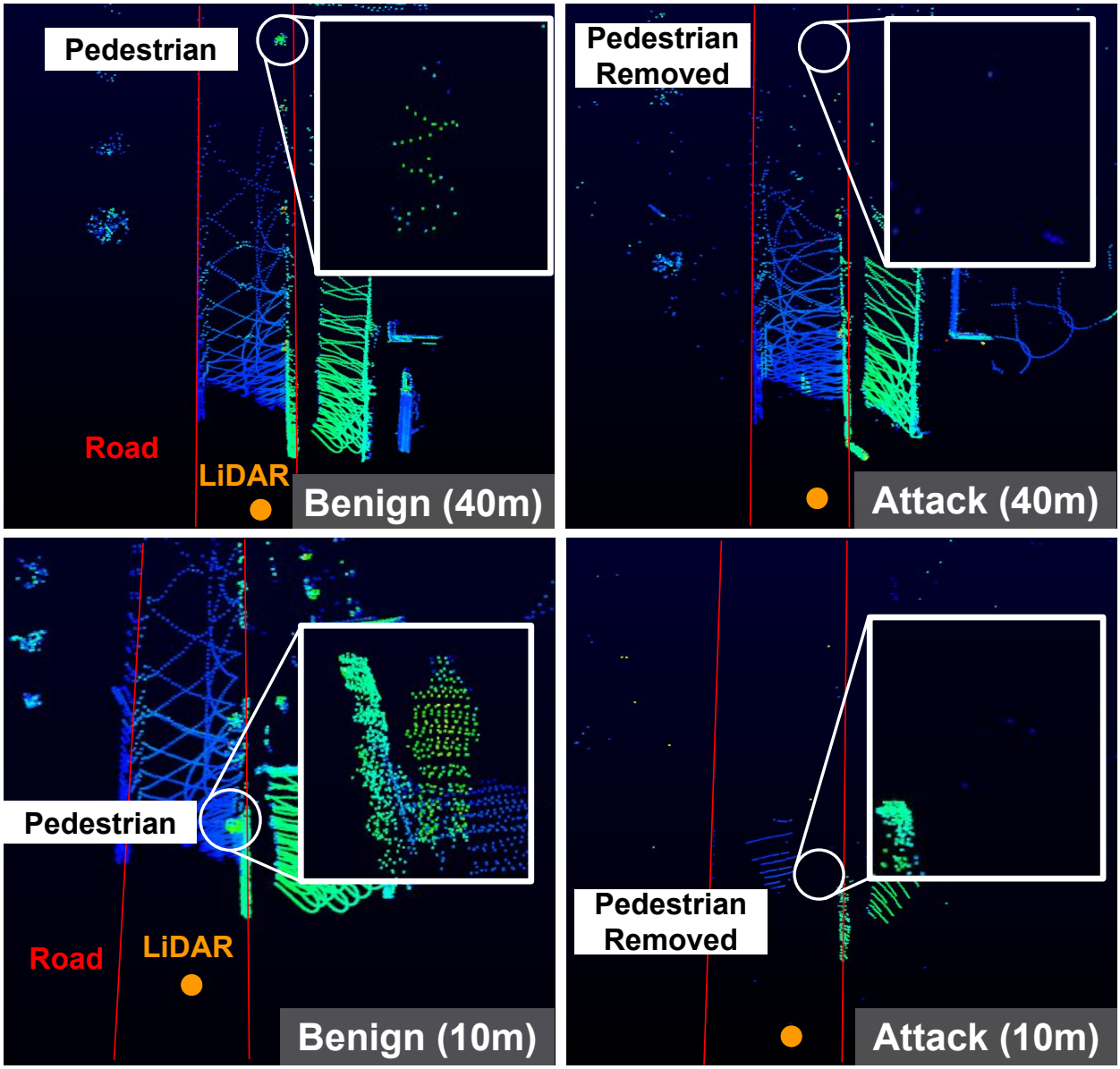


Figure 9.6.2: An example of LiDAR point clouds visible from a vehicle moving at 60 km/h. The HFR attack with our MVS system successfully eliminated a pedestrian 40 m away, and it is possible to continue to remove it until it approaches 10 m.

Table 9.6.2. Point removal rates of the HFR attack against Livox Horizon for different speeds by comparing the points in the benign scenarios at the same distance.

	Distance between victim and target (m)						
	0-10	10-20	20*-30	30-40	40-50	50-60	60-70
10 km/h	**90%**	89%	99%	**100%**	**100%**	**100%**	**100%**
20 km/h	79%	90%	98%	**100%**	**100%**	**100%**	**100%**
30 km/h	75%	92%	**100%**	**100%**	**100%**	99%	95%
40 km/h	75%	90%	97%	**100%**	**100%**	**100%**	**100%**
50 km/h	87%	**97%**	**100%**	**100%**	**100%**	**100%**	93%
60 km/h	87%	93%	97%	**100%**	**100%**	**100%**	97%

*: the braking distance at 60 km/h

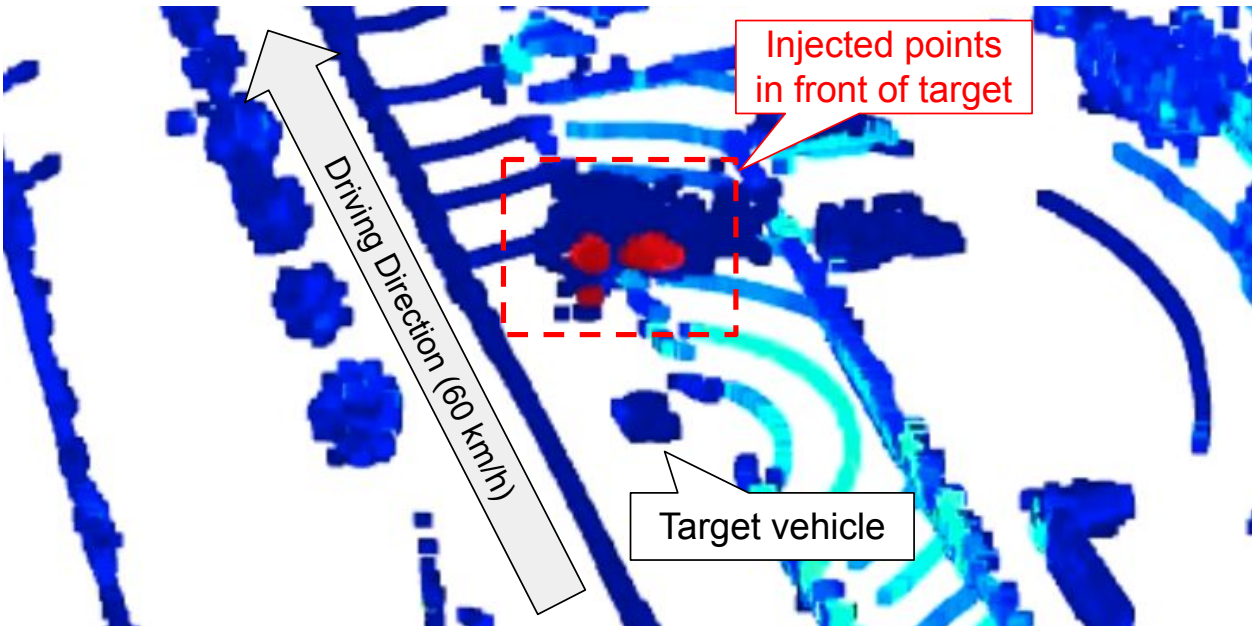


Figure 9.6.3: Point cloud under the synchronized injection attack in the high-speed driving scenario at 60 km/h

A-HFR Attack at High Speed

We evaluate the A-HFR attack against a high-speed driving vehicle with AT128 [6], which is one of the New-Gen LiDARs with pulse fingerprinting. The attack frequency is set at 15 MHz. For the object detector, we use the PointPillars used in Apollo 6.0 [7]. Table 9.6.3 lists the attack success rates of the A-HFR attack for 6 different speeds. As listed, the A-HFR attack is quite successful at distances of up to 40 meters away from the victim, with attack success rates of 100% at 60 km/h. Similar to the observations in the HFR experiment, there was no strong correlation between the vehicle's speed and the attack success rate, indicating that accurate tracking was achieved. However, the attack success rate drops to 50% when the vehicle is 20 meters away. This degradation is mainly due to the occasional failure to receive the laser from the LiDAR in the optical receiver during certain LiDAR scans. Currently, for each LiDAR scan, the A-HFR attack is triggered only if the laser is received by the PD; consequently, the attack inevitably fails for scans in which the LiDAR laser is undetected by the optical receiver. Even with a parabolic antenna, the laser light from a LiDAR mounted on a distant vehicle follows a sparse pattern, making it prone to occasional misses by the

Table 9.6.3. Attack success rates of the A-HFR attack against AT128 for different speeds. The "benign" row lists the detection success rates of the target pedestrian without attacks.

	Distance between victim and target (m)						
	0-10	10-20	20*-30	30-40	40-50	50-60	60-70
10 km/h	42%	61%	46%	84%	95%	**100%**	**100%**
20 km/h	37%	70%	42%	80%	94%	**100%**	**100%**
30 km/h	**44%**	67%	45%	86%	97%	**100%**	**100%**
40 km/h	**44%**	66%	40%	80%	91%	**100%**	**100%**
50 km/h	34%	61%	**50%**	73%	81%	**100%**	**100%**
60 km/h	27%	**72%**	**50%**	**87%**	**100%**	**100%**	**100%**
Benign	32%	50%	49%	53%	70%	**100%**	**100%**

*: the braking distance at 60 km/h

Table 9.6.4. Point removal rates of the HFR attack against the target vehicle driving at 60 km/h for the different numbers of beams and sizes of lenses in the MVS system.

		Distance between Pedestrian & LiDAR (m)						
# of beams	Lens size	0-10	10-20	20-30	30-40	40-50	50-60	60-70
1	1 inch	49%	63%	61%	76%	88%	94%	99%
1	2 inches	49%	43%	63%	69%	89%	**100%**	**100%**
2	2 inches	**87%**	**93%**	**97%**	**100%**	**100%**	**100%**	97%

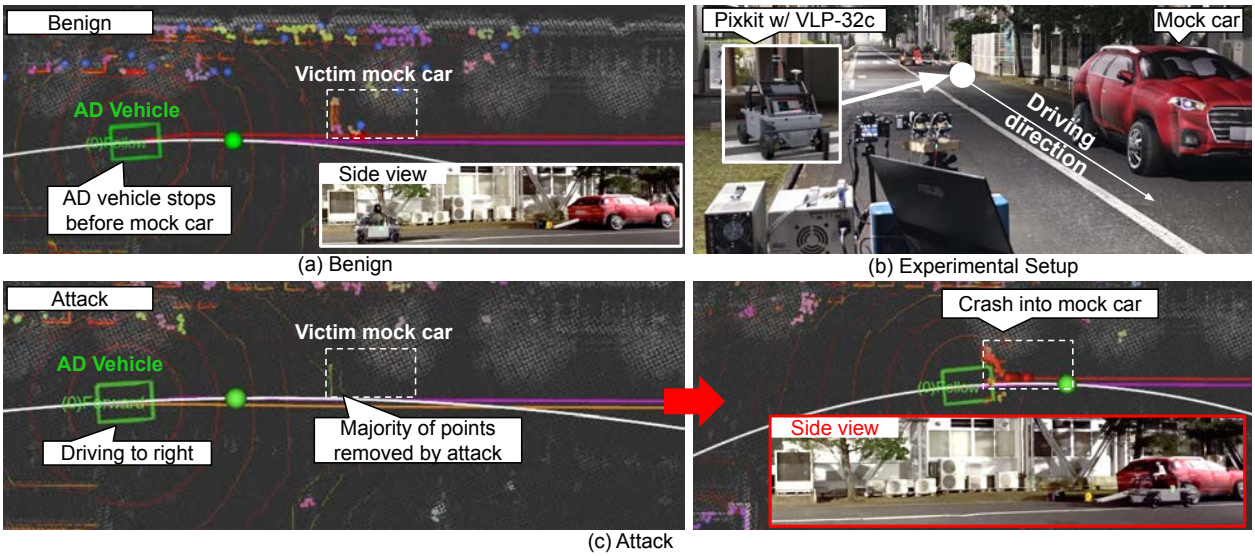
receiver. This limitation can be handled by two approaches: improving the optical receiver sensitivity or implementing speculative execution. Further details are provided in §9.8.

9.6.3 Ablation Study on Hardware Configuration of MVS Systems

We evaluate the validity of our MVS system design introduced in §9.3 in practical high-speed and long-range attack scenarios. Table 9.6.4 lists the point removal rates of the HFR attack for different numbers of beams and sizes of lenses in the scenario with a 60 km/h driving speed. As listed, our setup with 2 beams and 2-inch lenses achieves $\geq$97% attack success rates until 20 meters, which corresponds to the braking distance at 60 km/h [39]. We consider that the beam size of roughly 4 inches x 4 inches proves a sufficiently large total beam size to cover inaccuracies in the current MVS system, as the improvement observed when doubling the lens size is not as significant as the improvement achieved by doubling both the lens size and the number of beams.

Table 9.6.5. Averaged number of points and their angles injected by the synchronized attack against VLP-32c for each 10-meter bin in the 60 km/h scenario.

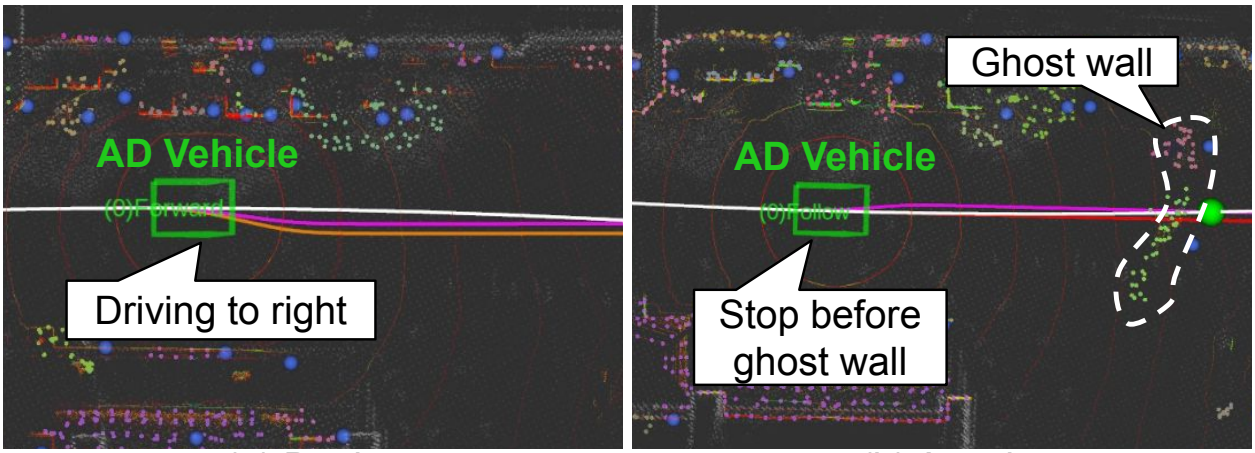
	Distance between injected wall and target (m)					
	0-10	10-20	20-30	30-40	40-50	50-60
# of injected points	410	676	1,016	897	623	248
Point Range	4.7	6.6	10.4	10.2	8.1	3.1



(a) Benign (b) Experimental Setup

(c) Attack

Figure 9.6.4: Experimental Setup and removal attack results on AD vehicle. We deploy the HFR attack with our MVS system against PIXKIT with VLP-32c controlled by Autoware.ai. PIXKIT drives from 40 meters away from the MVS system.



(a) Benign (b) Attack

Figure 9.6.5: Injection attack results on AD vehicle. We deploy the synchronized injection attack with our MVS system against the PIXKIT with VLP-32c controlled by Autoware.ai.

9.6.4 Injection Attack Evaluation

We evaluate the synchronized injection attack in the high-speed driving scenarios against VLP-32c [46] with a deterministic scan pattern, which is a requirement for the white-box

attack. We inject a ghost wall in front of the target vehicle at a fixed position in the world coordinate by following the methodology in [213] as shown in Fig. 9.6.3. Table 9.6.5 lists the average number of points and their angles injected by the synchronized attack against VLP-32c for each 10-meter bin from the injected wall to the target vehicle in the 60 km/h scenario. As listed, the attack can inject $\geq$1k points at around 20 meters away, which is already the braking distance at 60 km/h [39], meaning that hard braking is inevitable to avoid the injected wall. Unlike the removal attack, the injection attack may cause serious safety consequences (e.g., hard braking) even if the attack is only successful for a few frames. We will evaluate the system-level impact of the injection attack in the next section (§9.7).

9.7 End-to-End Evaluation on AD Vehicle

We finally evaluate the attack effectiveness of LiDAR spoofing attacks with our MVS system against an actual AD vehicle. We deploy the attacks against Autoware.ai [219], a popular open-sourced AD stack, installed on PIXKIT [88], an AD development platform.

Experimental Setup

Fig. 9.6.4 (b) shows the experimental setup for the end-to-end attack evaluation on an AD vehicle, PIXKIT [88] with a VLP-32c [46] LiDAR controlled by Autoware.ai version 1.14.1 [219]. For the LiDAR object detection, we use the LiDAR Euclidean cluster detection, which is officially supported by Autoware.ai. We placed our MVS system on the roadside of our private road with exclusive use permission. The AD vehicle started accelerating 40 meters away from our MVS system. For the removal attack, we placed an SUV-sized inflatable mock car 5 meters away from the MVS system and tried to make the mock car undetected by the HFR attack. For the injection attack, we injected a ghost wall as shown

in Fig. 9.6.5 to see if the AD vehicle would stop before the ghost wall. We accelerate the AD vehicle up to 5 km/h and 15 km/h for injection and removal attacks, respectively. The 15 km/h for injection attacks is the maximum speed we could test on the testing road. The 5 km/h for removal attacks is an acceptable speed to avoid damage to the facilities. To ensure safety, the experiments were conducted only during nighttime.

Removal Attack Results

As shown in Fig. 9.6.4 (a) the AD vehicle adequately stopped 7 meters before the front mock vehicle in the benign scenario. The "Follow" word on the AD vehicle indicates the current driving mode, meaning that the AD vehicle is following a front object and must stop before it. On the other hand, the AD vehicle failed to detect the mock car and crashed into it in the attack scenario as shown in Fig. 9.6.4 (c), as evidenced by the driving mode "Foward", meaning the AD vehicle is decided to drive forward. These results strongly support the end-to-end safety impacts of object removal effects of LiDAR spoofing attacks. The HFR attack with our MVS system has a high potential to cause a hard crash into objects on roads, such as cars and pedestrians, in the real world.

Injection Attack Results

Fig. 9.6.5 shows the attack results of the synchronized injection attack. In the benign scenario (the left figure), the AD vehicle keeps driving forward at 15 km/h. In the attack scenario (the right figure), the AD vehicle permanently stops before the injected ghost wall during the attack. The videos of the camera views and perception results are available on our website: **https://sites.google.com/view/cav-sec/new-gen-lidar-sec**. These indicate that the injection attack can make an AD vehicle stop on a road and thus has a high potential to cause serious safety and transportation concerns such as hard braking and traffic jams.

9.8 Discussions and Limitations

Safety Implications

The safety implications (e.g., hard crash into a parking car and hard braking) demonstrated with an AD vehicle in §9.7 are very likely to be feasible for AD vehicles driving at fast speeds, with our MVS system. Although we could not evaluate the attack on a high-speed vehicle actually controlled by an AD stack due to the limitations in our testing facility, the attack effectiveness in the high-speed driving scenarios has been addressed in §9.6, which shows that the HFR attack with our MVS system can achieve $\geq$96% attack success rates at 60 km/h until the braking distance, 20 meters away from the victim. We believe that this work poses the serious safety implications of LiDAR spoofing attacks significantly more clearly than prior work. We have finished the responsible vulnerability disclosure process for AD companies, considering the high security and safety impacts of this work on the AD industry and some of them are investigating the impacts.

Possible Countermeasures

We consider that the use of New-Gen LiDARs is mandatory to be more robust against LiDAR spoofing attacks, as also mentioned in [290]. For example, New-Gen LiDARs with timing randomization can effectively defend against synchronized injection attacks. For removal attacks, pulse fingerprinting shows high attack mitigation capability. While the A-HFR attack can potentially bypass it, this attack increases the hardware requirements of the MVS system. However, as discussed in §9.2.2, timing randomization and pulse fingerprinting are installed originally for anti-interference, not for security purposes. We encourage LiDAR manufacturers to enhance the magnitude of randomization and fingerprinting as high as possible to make them work as security features. We also strongly recommend that AD

companies develop and install an availability check method to determine if their LiDARs are functioning properly. For example, Tesla shows an alert when all three front cameras do not correctly work [341]. Particularly, the HFR and A-HFR attacks work like "jamming". It should not be hard to detect their distinctive randomized point pattern as in Fig. 9.5.3 and also shown in [290], e.g., by an entropy-based method [351].

Multiple Sensor Fusion

Multiple sensor fusion could be an effective mitigation strategy since the current MVS systems are designed to track only one LiDAR sensor on a vehicle. However, it is not technically difficult to attack multiple LiDARs by driving multiple MVS systems simultaneously for the HFR or A-HFR attacks. Furthermore, LiDARs struggle with scanning duplicated areas due to interference, meaning each area is likely scanned by only a single LiDAR. The MVS system thus may not need to attack multiple LiDARs even if the target AD has multiple LiDARs.

Fusion with different types of sensors such as cameras could be a mitigation strategy. However, the current AD vehicles, especially Level-4 AD, heavily rely on LiDARs as our attack works on the actual AD stack as demonstrated in §9.7. Sensor fusion generally improves robustness, but it is currently far from sufficient defense. Moreover, more sensors result in increased costs and may even open new attack channels.

Attack Deployment on Uneven Roads

Our current threat model only assumes a flat and straight road as an attack site, i.e., we assume that the height of the target LiDAR is constant and known, and thus the auto-aiming is only designed for the horizontal direction. Thus, the current MVS system has a limitation in vertical tracking. Meanwhile, we note that the majority of high-speed scenarios should be only on flat and straight roads because the vehicle cannot accelerate on uneven and winding

roads. In low-speed scenarios, tracking both for horizontal and vertical directions should not be challenging as demonstrated in prior work [127, 125]. Furthermore, our arrayed laser design (§9.3.4) allows us to even eliminate the need for vertical tracking by vertically arraying more lasers. The large number of arrayed lasers harms the portability of the MVS system. The attacker needs to decide the number of lasers based on the tracking performance in the target attack scenarios.

Speculative Execution for A-HFR Attack

As discussed in §9.6.2, we find that the current optical receiver fails to receive lasers from the lasers at many frames and fails to launch the attack at the frames. There are generally two approaches to handling the issue. The first approach is to improve the optical receiver. We can enlarge the antenna to receive more signals or integrate the antenna into the auto-aiming system to more surely face the antenna to the LiDAR. Another approach is to implement speculative execution of the attack at each frame based on the attack timing in the previous frame. Currently, we trigger the A-HFR attack only when the laser is received for each frame, i.e., the attack always fails if we fail to receive the laser for the frame. To handle this issue, we can start the attack even if the laser is not received at a frame based on the intervals between the lasers from the LiDAR in the previous frames. Since the interval period between the LiDAR frame is periodic, such speculative execution should effectively improve the A-HFR attack. This should result in a shorter rest and more overheating, but it should be better than a mere fail without launching the attack. We consider that the speculative execution can be realizable in FPGAs to execute the logic in real time.

Ethical and Safety Considerations

The experiments were safely carried out in controlled conditions on a private road with exclusive-use permission. A human with a driving license drove the experimental vehicle, and the area was surveilled to keep people off the road. During the experiments, participants potentially exposed to the attack laser wore protective goggles for eye safety.

9.9 Conclusion

In this study, we investigate the safety and security impact of LiDAR spoofing attacks against AD vehicles driving at high speeds in practical long-range attack scenarios. We first identify 3 research limitations in prior work to prevent them from deploying their LiDAR spoofing attacks in practical AD scenarios. To address the limitations, we design a novel MVS system that can detect, track, and aim at the target LiDAR moving at high speeds, and also design a new practical removal attack, A-HFR attack, which can be effective against New-Gen LiDARs even with pulse fingerprinting. We demonstrate that our designs in the MVS system can achieve significantly long attack distances (e.g., 110 meters) in a LiDAR-agnostic way. We demonstrated that both injection and removal attacks can be deployed against high-speed vehicles driving at 60 km/h, and the attacks can have end-to-end attack impacts on a popular AD stack. This study bridges critical gaps between LiDAR security and AD security research. We hope that our study can initiate a new sight to adequately understand the safety and security implications of LiDAR in AD perception.

Appendix

9.A IR Camera Configurations

Fig. 9.A.1 shows the configuration of our IR camera (ELP-USBFHD08S-MFV [92]) with the bandpass filter (FBH905-10 [93]). We replaced the pre-installed low-pass filter in front of the CMOS sensor with the bandpass filter to selectively receive the IR wavelength (905 ± 5 nm), which is used in the majority of LiDARs [20]. To attack LiDARs with other wavelengths, the adversary needs to use another bandpass filter corresponding to the wavelength range covering the wavelength.

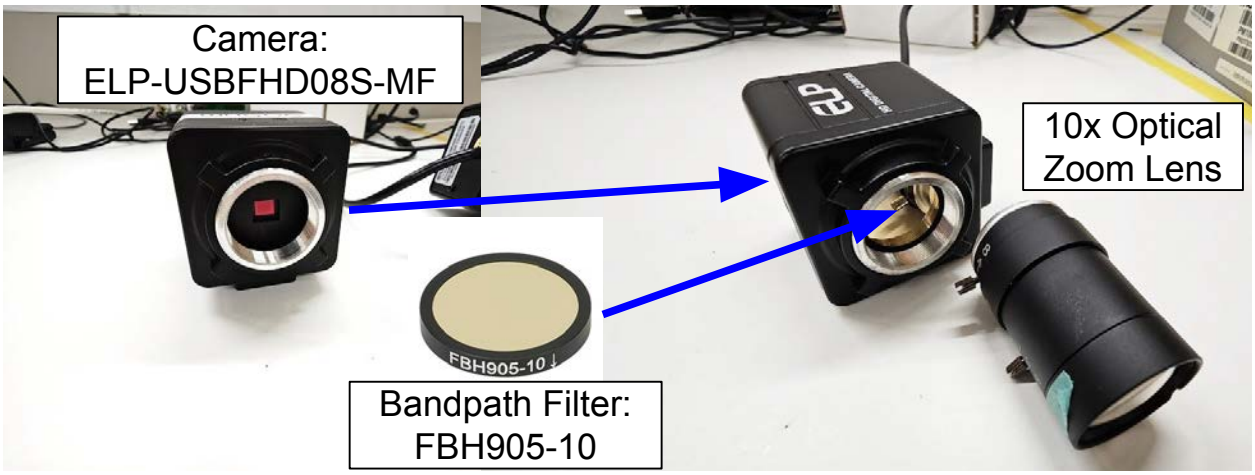


Figure 9.A.1: IR Camera Configurations. We install the bandpass filter in front of the image sensor and place the 10 times optical zoom lens on it.

9.B Impact of Vertical and Horizontal Attack Angles

We evaluate the laser pulse peak power at different attack frequencies to evaluate the improvements of the A-HFR attack over the HFR attack. For the A-HFR attack, we evaluate two different vertical attack angles: 32° (v=100%) and 16° (v=50%). Fig. 9.B.1 shows the laser pulse peak power at different attack frequencies. We calculate the average power of the

attack laser with a laser power meter and then divide this value by the laser frequency to obtain the power per pulse. As designed, the A-HFR attack can achieve higher peak powers at the same attack frequencies compared to the HFR attack. If we limit more vertical attack angles, the A-HFR can achieve higher attack capability by reducing the vertical attack angle as well. The A-HFR attack between 10 and 20 MHz can achieve substantially higher power, with a 10 times increase. This attack angle reduction approach of the A-HFR is necessary to achieve high frequencies such as 25 MHz.

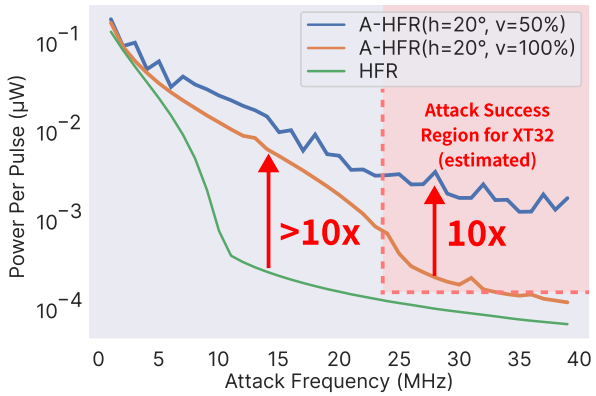


Figure 9.B.1: Laser pulse peak power of the HFR and A-HFR attacks at different attack frequencies. The A-HFR attack can achieve higher laser power at the same frequency.

Fig. 9.B.2 shows the efficacy of A-HFR's vertical attack angle reduction. Since the angle reduction boosts the peak power under high attack frequencies (Fig. 9.B.1), this expands the point cloud removal angle by about 20%.

9.C HFR Attack on Mid-360

Fig. 9.C.1 shows the efficacy of the conventional HFR attack on Mid-360, equipped with pulse fingerprinting. We found that even the HFR attack can fully remove points in 20° horizontally and vertically. The Mid-360's vulnerability can be attributed to its relatively less secure authentication system compared to other LiDARs. This suggests that a larger T_α results in potentially lower power in legitimate pulses, indicating that certain LiDAR models might have more lenient T_α settings and hence weaker security features.

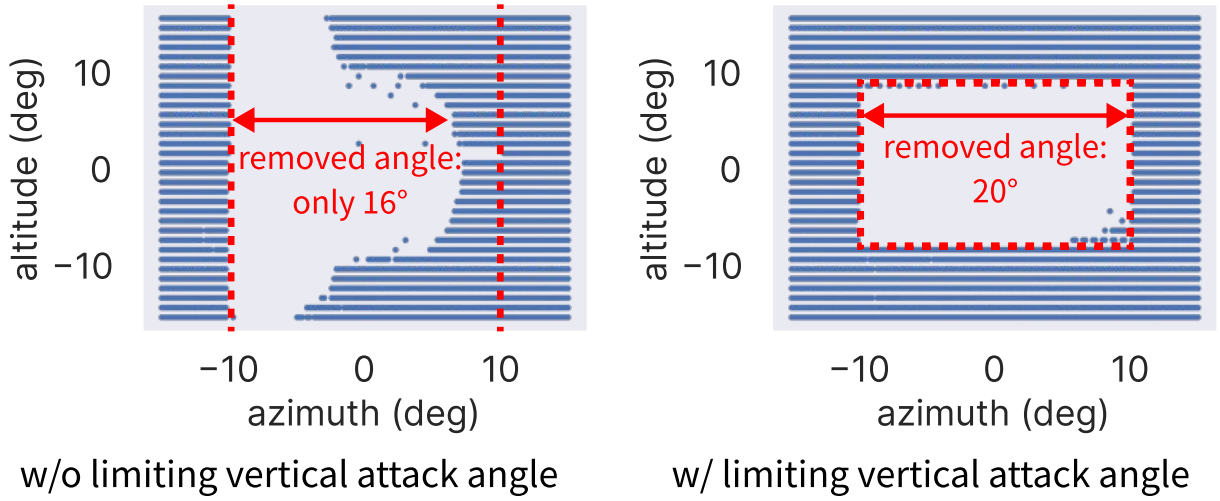


Figure 9.B.2: 2D-projected point cloud of the A-HFR attack on XT32 with horizontal 20°. Blue points mean the remaining points, and the red dotted line indicates the attack angle.

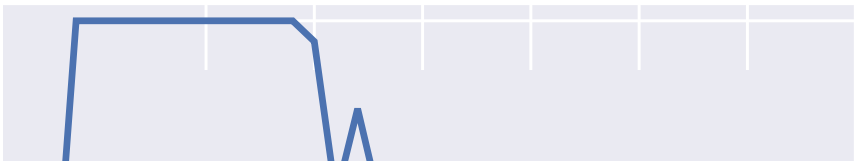


Figure 9.C.1: Point removal percentage by HFR attack against Livox Mid-360, under the same condition as Fig. 9.5.2.

Chapter 10

Conclusion and Future Work

10.1 Conclusion

In my dissertation, I have systematically conducted the security analysis of camera and LiDAR-based perception systems in Autonomous Driving (AD) vehicles. By addressing the four critical research challenges outlined in the introduction, I have gained meaningful insights into the end-to-end impact and safety implications of these security threats in real-world AD scenarios. My research contributions can be summarized as follows:

Security Analysis on Camera-based AD Perception:

- *Dirty Road Patch (DRP) Attack:* I am the first to systematically study the security of state-of-the-art deep learning-based ALC systems in their designed operational domains under physical-world adversarial attacks. I formulate the problem with a safety-critical attack goal, and a novel and domain-specific attack vector: dirty road patches. To systematically generate the attack, I adopt an optimization-based approach and overcome domain-specific design challenges such as camera frame inter-dependencies due to attack-

influenced vehicle control, and the lack of objective function design for lane detection models. I evaluate our attack on a production ALC using 80 scenarios from real-world driving traces. The results show that our attack is highly effective with over 97.5% success rates and less than 0.903 sec average success time, which is substantially lower than the average driver reaction time. This attack is also found (1) robust to various real-world factors such as lighting conditions and view angles, (2) general to different model designs, and (3) stealthy from the driver's view. To understand the safety impacts, I conduct experiments using software-in-the-loop simulation and attack trace injection in a real vehicle. The results show that our attack can cause a 100% collision rate in different scenarios, including when tested with common safety features such as automatic emergency braking. I also evaluate and discuss defenses. (Chapter 3)

- *LaneGuard:* I designed the first map-fusion-based off-road attack detection-based defense, LaneGuard, LaneGuard detects off-road attacks that lead victim vehicles out of their driving lane, based on the difference between the observed road shape and the driver-predefined route shape. I evaluate LaneGuar on large-scale real-world driving traces consisting of 80 attack scenarios and 11,558 benign scenarios. I find that LaneGuard can achieve an attack detection rate of 89% with a 12% false positive rate. In real-world highway driving experiments, LaneGuard exhibits no false positives while maintaining a near-zero false negative rate against simulated attacks. (Chapter 4)

- *Driving-Oriented Metrics for Lane Detection:* I have designed 2 new driving-oriented metrics for lane detection: End-to-End Lateral Deviation metric (E2E-LD) is directly formulated based on the requirements of autonomous driving, a core downstream task of lane detection; Per-frame Simulated Lateral Deviation metric (PSLD) is a lightweight surrogate metric of E2E-LD. To evaluate the validity of the metrics, I conduct a large-scale empirical study with 4 major types of lane detection approaches on the TuSimple dataset and our newly constructed dataset Comma2k19-LD. Our results show that the

conventional metrics have strongly negative correlations ($\leq$-0.55) with E2E-LD, meaning that some recent improvements purely targeting the conventional metrics may not have led to meaningful improvements in autonomous driving, but rather may actually have made it worse by overfitting to the conventional metrics. As autonomous driving is a security/safety-critical system, the underestimation of robustness hinders the sound development of practical lane detection models. I hope that our study will help the community achieve more downstream task-aware evaluations for lane detection. (Chapter 5)

- *Infrared Laser Reflection (ILR) Attack:* I have developed an effective physical-world attack that leverages the sensitivity of filterless image sensors and the properties of Infrared Laser Reflections (ILRs), which are invisible to humans. The attack is designed to affect CAV cameras and perception, undermining traffic sign recognition by inducing misclassification. In this work, I formulate the threat model and requirements for an ILR-based traffic sign perception attack to succeed. I evaluate the effectiveness of the ILR attack with real-world experiments against two major traffic sign recognition architectures on four IR-sensitive cameras. Our black-box optimization methodology allows the attack to achieve up to a 100% attack success rate in indoor, static scenarios and a $\geq$80.5% attack success rate in our outdoor, moving vehicle scenarios. I find the latest state-of-the-art certifiable defense is ineffective against ILR attacks as it mis-certifies $\geq$33.5% of cases. To address this, I propose a detection strategy based on the physical properties of IR laser reflections which can detect 96% of ILR attacks. (Chapter 6)

- *Natural Denoising Diffusion (NDD) attack:* I have discovered a new type of attack, called the Natural Denoising Diffusion (NDD) attack based on the finding that state-of-the-art deep neural network (DNN) models still hold their prediction even if I intentionally remove their robust features, which are essential to the human visual system (HVS), through text prompts. The NDD attack shows a significantly high capability to generate low-cost, model-agnostic, and transferable adversarial attacks by exploiting the natural

attack capability in diffusion models. To systematically evaluate the risk of the NDD attack, I perform a large-scale empirical study with our newly created dataset, the Natural Denoising Diffusion Attack (NDDA) dataset. I evaluate the natural attack capability by answering 6 research questions. Through a user study, I find that it can achieve an 88% detection rate while being stealthy to 93% of human subjects; I also find that the non-robust features embedded by diffusion models contribute to the natural attack capability. To confirm the model-agnostic and transferable attack capability, I perform the NDD attack against the Tesla Model 3 and find that 73% of the physically printed attacks can be detected as stop signs. Our hope is that the study and dataset can help our community be aware of the risks in diffusion models and facilitate further research toward robust DNN models. (Chapter 7)

Security Analysis on LiDAR-based AD Perception:

- *Large-Scale Measurement Study of LiDAR Spoofing Attacks on New-Gen LiDARs:* I have discovered the following three critical research gaps in prior LiDAR security research: (1) considering only one specific LiDAR (VLP-16); (2) assuming unvalidated attack capabilities; and (3) evaluating object detectors with limited spoofing capability modeling and setup diversity. To fill these critical research gaps, I conduct the first large-scale measurement study on LiDAR spoofing attack capabilities on object detectors with 9 popular LiDARs, covering both first- and new-generation LiDARs, and 3 major types of object detectors trained on 5 different datasets. To facilitate the measurements, I (1) identify spoofer improvements that significantly improve the latest spoofing capability, (2) identify a new object removal attack that overcomes the applicability limitation of the latest method to new-generation LiDARs, and (3) perform novel mathematical modeling for both object injection and removal attacks based on our measurement results. Through this study, I are able to uncover a total of 15 novel findings, including not only completely

new ones due to the measurement angle novelty, but also many that can directly challenge the latest understandings in this problem space. (Chapter 8)

- *LiDAR Spoofing Attacks against AD Vehicles at High Speed and Long Distance:* With my novel Moving Vehicle Spoofing (MVS) system, I am not only the first to demonstrate LiDAR spoofing attacks against practical AD scenarios where the victim vehicle is driving at high speeds (60 km/h) and the attack is launched from long distances (110 meters), but I are also the first to perform LiDAR spoofing attacks against a vehicle actually operated by a popular AD stack. Furthermore, I design a new object removal attack, an adaptive high-frequency removal (A-HFR) attack, that can be effective even against recent LiDARs with pulse fingerprinting features, by leveraging gray-box knowledge of the scan timing of target LiDARs. Our object removal attack achieves $\geq$96% attack success rates against the vehicle driving at 60 km/h to the braking distances (20 meters). Finally, I discuss possible countermeasures against attacks with our MVS system. This study not only bridges critical gaps between LiDAR security and AD security research but also sets a foundation for developing robust countermeasures against emerging threats. (Chapter 9)

The rapid integration of autonomous systems into daily life mandates rigorous research to ensure their security and reliability. This dissertation has revealed significant vulnerabilities in the perception systems of AD vehicles and highlighted the need for continued research and development in this field. By addressing these vulnerabilities, I believe that we can pave the way for the safe and widespread adoption of AD technology. This comprehensive security analysis of camera and LiDAR-based vulnerabilities in AD systems highlights the importance of proactive security measures and sets a foundation for future advancements in the field of AD security.

10.2 Future Work

While our research has made substantial progress in understanding and demonstrating the vulnerabilities in AD perception systems, several future research directions are motivated toward more secure AD:

10.2.1 Effective Countermeasures for AD Perception

Through my dissertation research, I demonstrated that the camera- and LiDAR-based perception systems have notable vulnerabilities against adversarial attacks (Chapter 3, 6, 6, and 7) and spoofing attacks (Chapter 8 and 9). Although I designed a possible detection-based defense that can be effective against off-road attacks including my DRP attack in Chapter 4, it does not sufficiently address the vulnerabilities in the AD perception systems. Generally, there are two major approaches to achieving more secure perception systems: model-level and sensor-level defenses

Model-level defenses aim to improve as similar quality of perception as human driver's level. However, even with numerous efforts to design defenses or migitations [241, 349, 352, 149], none of them have achieved acceptable performance and recent research efforts are in arm racing. Furthermore, these prior works concentrate on image classification or object detection models. Further research efforts are demanded in this area.

Sensor-level defenses aim to detect or remove attack traces in sensor inputs, especially for LiDARs as discussed in Chapter 8 and 9. For example, Tesla shows an alert when all three front cameras do not correctly work [341]. For the HFR and A-HFR attacks, these attacks work like "jamming" and thus their distinctive randomized point patterns should be detectable, e.g., by an entropy-based method [351].

I first explored detection-based defenses as attack identification should be the first step for all defense strategies. However, these defenses inherently require the following fail-safe designs to avoid safety concerns. Although even simple measures (e.g., slowing down) after attack detection may mitigate the fatality, adequate recovery is still highly scenario-dependent and an open problem. The fail-safe design has also huge room for research.

10.2.2 Security Analysis of AD Perception with Sensor Fusion

Multiple-sensor fusion has a high potential to enhance the attack robustness. As shown in Chapter 9, the current AD vehicles, especially Level-4 AD, heavily rely on LiDARs. Fusion with different types of sensors such as cameras, radars, and ultrasonic sensors could be a mitigation strategy. However, more sensors result in increased costs and may even open new attack channels. Prior research has also identified a wide variety of vulnerabilities in these sensors, cameras [289, 373], radars [257, 353], ultrasonic sensors [353]. Thus, security analysis of AD perception in sensor fusion scenarios is required to properly assess the benefits and risks of multiple sensor fusion for AD perception.

10.2.3 Security Analysis on Emerging Sensors

The driving quality of AD has been significantly improved along with the development of sensors such as LiDARs. For example, FMCW LiDARs [47] and mmWave radars [204] are under evaluation for use in AD since FMCW LiDARs can obtain relative speeds of the sensed objects and mmWave radars can achieve much more fine-grained sensing than normal radars. Future research is required to ensure the security and safety of these sensors in AD.

10.2.4 Security Analysis on LLM-based AD

Large Language Models (LLMs) [123, 99, 329, 330, 106] have been showing significant promise in decision-making tasks including AD when fine-tuned on specific applications, leveraging their inherent common sense and reasoning abilities learned from vast amounts of data. LLM-based decision-making and autonomous systems [345, 171, 235, 170, 297, 121, 217] take textual scenario descriptions as input to generate strategic decisions and corresponding behavior plans for autonomous agents. However, these systems are exposed to substantial safety and security risks during the fine-tuning phase. Before leveraging the power of LLM-based decision-making, adequate security analysis is mandatory.

Bibliography

[1] 81160A Pulse Function Arbitrary Noise Generator. https://www.keysight.com/us/en/product/81160A/81160a-pulse-function-arbitrary-noise-generator.html.

[2] AEye Automotive. https://www.aeye.ai/solutions/automotive/.

[3] AFG310 and AFG320 User Manual. https://www.tek.com/en/afg310-manual/afg310-and-afg320-user-manual.

[4] Alpha Prime. https://velodynelidar.com/products/alpha-prime/.

[5] Amazon Mechanical Turk. https://www.mturk.com/.

[6] AT128 Auto-Grade Ultra-High Resolution Long Range Lidar — HESAI Technology. https://www.hesaitech.com/product/at128/.

[7] Baidu Apollo. https://github.com/ApolloAuto/apollo.

[8] Bing Maps. https://www.bing.com/maps. Accessed: 2023-12.

[9] Cruise. https://www.getcruise.com/.

[10] datasheet-rev06-v2p3-os1.pdf. https://data.ouster.io/downloads/datasheets/datasheet-rev06-v2p3-os1.pdf.

[11] Dirty Road Patch Attack Project Website. https://sites.google.com/view/cav-sec/drp-attack.

[12] EOSRAM Radial T1 3/4, SPL PL90_3. https://www.osram.com/ecat/com/en/class_pim_web_catalog_103489/prd_pim_device_2220019/.

[13] EPC9126/EPC9126HC Lidar Demo Boards . https://epc-co.com/epc/Products/DemoBoards/EPC9126.aspx.

[14] Google Maps. https://www.google.com/maps/. Accessed: 2023-12.

[15] Intel RealSense LiDAR Camera L515 Datasheet. https://dev.intelrealsense.com/docs/lidar-camera-l515-datasheet.

[16] Introduction to Self-Driving Cars. https://www.coursera.org/learn/intro-self-driving-cars.

[17] Laser Safety Facts. https://www.lasersafetyfacts.com/laserclasses.html.

[18] Leddar Pixell. https://leddarsensor.com/solutions/leddar-pixell/.

[19] LGSVL Simulator. https://github.com/lgsvl/simulator/.

[20] LiDAR for Automotive and Industrial Applications 2021. https://www.i-micronews.com/products/lidar-for-automotive-and-industrial-applications-2021/.

[21] Livox Detection V2.0. https://github.com/Livox-SDK/livox_detection.

[22] Livox Horizon User Manual. https://www.livoxtech.com/3296f540ecf5458a8829e01cf429798e/assets/horizon/Livox%20Horizon%20user%20manual%20v1.0.pdf.

[23] Livox Mid-360. https://www.livoxtech.com/mid-360.

[24] Luminar Iris. https://www.luminartech.com/iris/.

[25] Modeling a Vehicle Dynamics System. https://www.mathworks.com/help/ident/ug/modeling-a-vehicle-dynamics-system.html.
[26] Motional. https://motional.com/.
[27] NTSB Investigation Into Deadly Uber Self-Driving Car Crash Reveals Lax Attitude Toward Safety. https://spectrum.ieee.org/ntsb-investigation-into-deadly-uber-selfdriving-car-crash-reveals-lax-attitude-toward-safety.
[28] Nuro. https://www.nuro.ai/.
[29] OpenPilot: Open Source Driving Agent. https://github.com/commaai/openpilot.
[30] OpenRouteService. https://openrouteservice.org/. Accessed: 2023-12.
[31] OpenStreetMap. https://www.openstreetmap.org. Accessed: 2023-12.
[32] Our Project Website. https://sites.google.com/view/cav-sec/new-gen-lidar-sec.
[33] Ouster Introduces Digital Flash (DF) Series Lidar for Automotive. https://ouster.com/blog/ouster-automotive-df-series/.
[34] pgRouting: A routing library for PostgreSQL. https://pgrouting.org/. Accessed: 2023-12.
[35] PhantomX Robot Turret Kit,. https://www.trossenrobotics.com/.
[36] Ride-Hailing App - Make the Most of Your Drive - Waymo One. https://waymo.com/waymo-one/.
[37] RS-Helios. https://www.robosense.ai/en/rslidar/RS-Helios.
[38] Si PIN photodiode S6775. https://www.hamamatsu.com/us/en/product/optical-sensors/photodiodes/si-photodiodes/S6775.html.
[39] Stopping Distances on Wet and Dry roads, Queensland Government. https://www.qld.gov.au/transport/safety/road-safety/driving-safely/stopping-distances/graph.
[40] Super Cruise - Hands Free Driving — Cadillac Ownership. https://www.cadillac.com/world-of-cadillac/innovation/super-cruise.
[41] Tesla Autopilot. https://www.tesla.com/autopilot.
[42] Tesla Autopilot Support. https://www.tesla.com/support/autopilot.
[43] Tinkla: Tinkering with Tesla. https://tinkla.us/.
[44] TL082 LINEAR INTEGRATED CIRCUIT. http://www.unisonic.com.tw/datasheet/TL082.pdf.
[45] Toyota 2020 RAV4 Owner's Manual. https://www.toyota.com/t3Portal/document/om-s/OM0R024U/xhtml/OM0R024U.html?locale=en.
[46] Ultra Puck Surround View Lidar Sensor — Velodyne Lidar. https://velodynelidar.com/products/ultra-puck/.
[47] Understanding the magnificent FMCW LiDAR. https://www.thinkautonomous.ai/blog/fmcw-lidar/. Accessed: 2024-08.
[48] VLP-16 User Manual. https://velodynelidar.com/wp-content/uploads/2019/12/63-9243-Rev-E-VLP-16-User-Manual.pdf.
[49] XT32 — Mid-Range Mechanical Lidar — HESAI Technology. https://www.hesaitech.com/product/xt32/.
[50] California Penal Code 594 PC. https://leginfo.legislature.ca.gov/faces/codes_displaySection.xhtml?lawCode=PEN&sectionNum=594, 1872.
[51] California Vehicle Code 21663. https://leginfo.legislature.ca.gov/faces/codes_displaySection.xhtml?lawCode=VEH&sectionNum=21663, 1959.

[52] California Vehicle Code 23113. https://leginfo.legislature.ca.gov/faces/codes_displaySection.xhtml?lawCode=VEH&sectionNum=23113, 2000.

[53] The Unscented Kalman Filter for Nonlinear Estimation, author=Wan, Eric A and Van Der Merwe, Rudolph. In *Proceedings of the IEEE 2000 adaptive systems for signal processing, communications, and control symposium (Cat. No. 00EX373)*, pages 153–158. Ieee, 2000.

[54] Honda's Collision Mitigation Braking System CMBS. https://youtu.be/NJcy5ySOrM4, 2013.

[55] IIHS Issues First Crash Avoidance Ratings Under New Test Program. https://www.iihs.org/news/detail/iihs-issues-first-crash-avoidance-ratings-under-new-test-program, 2013.

[56] Prolific. https://researcher-help.prolific.co/hc/en-gb, 2014.

[57] Analog Discovery 3: 125 MS/s USB Oscilloscope, Waveform Generator, Logic Analyzer, and Variable Power Supply. https://digilent.com/shop/analog-discovery-3/, 2016.

[58] Building Maps for a Self-Driving Car. https://link.medium.com/Bo5pCOov95, 2016.

[59] Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. *SAE International,(J3016)*, 2016.

[60] HD Maps: New Age Maps Powering Autonomous Vehicles. https://www.geospatialworld.net/article/hd-maps-autonomous-vehicles/, 2017.

[61] Toyota Safety Sense Pre-Collision System (PCS) Settings and Controls. https://youtu.be/IY4g_zG1Qj0, 2017.

[62] TuSimple Lane Detection Challenge. https://github.com/TuSimple/tusimple-benchmark/tree/master/doc/lane_detection, 2017.

[63] Baidu Apollo HD Map. http://ggim.un.org/unwgic/presentations/2.2_Ma_Changjie.pdf, 2018.

[64] Hands-on with Comma.ai's Add-on Level 2 Autonomous Tech. https://www.cnet.com/roadshow/news/hands-on-with-comma-ais-add-on-level-2-autonomous-tech/, 2018.

[65] LiDAR: New "Eye"' for Vehicle Autonomy. https://www.mobilityengineeringtech.com/component/content/article/43672-sae-ma-02983, 2018.

[66] PostGIS, Spatial and Geographic Objects for PostgreSQL. https://postgis.net, 2018.

[67] SPL PL90 3. https://look.ams-osram.com/m/249c8a382f5faab2/original/SPL-PL90_3.pdf, 2018.

[68] Velodyne Just Cut the Price of Its Most Popular Lidar Sensor in Half. https://www.thedrive.com/tech/17297/velodyne-just-cut-the-price-of-its-most-popular-lidar-sensor-in-half, 2018.

[69] Waymo Has Launched its Commercial Self-Driving Service in Phoenix. https://www.businessinsider.com/waymo-one-driverless-car-service-launches-in-phoenix-arizona-2018-12, 2018.

[70] Adhesive Patch can Seal Potholes and Cracks on the Road. https://www.startupselfie.net/2019/05/07/american-road-patch-seals-potholes-road-cracks/, 2019.

[71] American Road Patch - Deployment Demonstration Video for Adhesive Road Patch from 4:54 to 5:04. https://youtu.be/Vr_Dxg1LdxU?t=294, 2019.

[72] 'Anyone Relying on Lidar is Doomed,' Elon Musk Says. https://techcrunch.com/2019/04/22/anyone-relying-on-lidar-is-doomed-elon-musk-says/, 2019.

[73] Does Your Car Have Automated Emergency Braking? It's a Big Fail for Pedestrians. https://www.zdnet.com/article/does-your-car-have-automated-emergency-braking-its-a-big-fail-for-pedestrians/, 2019.
[74] Experimental Security Research of Tesla Autopilot. https://keenlab.tencent.com/en/whitepapers/Experimental_Security_Research_of_Tesla_Autopilot.pdf, 2019.
[75] GM Cadillac CT6 Owner's Manual. https://www.cadillac.com/content/dam/cadillac/na/us/english/index/ownership/technology/supercruise/pdfs/2020-cad-ct6-owners-manual.pdf, 2019.
[76] Toyota 2019 Camry Owner's Manual. https://www.toyota.com/t3Portal/document/om-s/OM06142U/pdf/OM06142U.pdf, 2019.
[77] Watch Tesla Drivers Apparently Asleep at the Wheel, Renewing Autopilot Safety Questions. https://www.cnbc.com/2019/09/09/watch-tesla-drivers-apparently-asleep-at-the-wheel-renewing-safety-questions.html, 2019.
[78] Waymo Open Dataset: An Autonomous Driving Dataset. https://www.waymo.com/open, 2019.
[79] Collision Avoidance Strikeable Targets for AEB. http://www.pedstrikeabletargets.com/, 2020.
[80] Driver Take-Over Decision Survey with Automated Lane Centering System in our Attack Stealthiness User Study. https://storage.googleapis.com/driving-decision-survey/driving_decision_survey.pdf, 2020.
[81] Is a $1000 Aftermarket Add-On as Capable as Tesla's Autopilot and Cadillac's Super Cruise? https://www.caranddriver.com/features/a30341053/self-driving-technology-comparison/, 2020.
[82] Lane Keeping Assist System Using Model Predictive Control. https://www.mathworks.com/help/mpc/ug/lane-keeping-assist-system-using-model-predictive-control.html, 2020.
[83] Manual on Uniform Traffic Control Devices Part 3 Markings. https://mutcd.fhwa.dot.gov/pdfs/millennium/06.14.01/3ndi.pdf, 2020.
[84] OpenPCDet: An Open-source Toolbox for 3D Object Detection from Point Clouds. https://github.com/open-mmlab/OpenPCDet, 2020.
[85] Tesla Admits its Approach to Self-Driving is Harder But Might be Only Way to Scale. https://electrek.co/2020/06/18/tesla-approach-self-driving-harder-only-way-to-scale/, 2020.
[86] Tesla Model 3 Owner's Manual. https://www.tesla.com/sites/default/files/model_3_owners_manual_north_america_en.pdf, 2020.
[87] We Hit the Road with Comma.ai's Assisted-Driving Tech at CES 2020. https://www.cnet.com/roadshow/news/comma-ai-assisted-driving-george-hotz-ces-2020/, 2020.
[88] PIXKIT: An Autonomous Driving Development and Education Kit. https://www.pixmoving.com/pixkit, 2021.
[89] RoboSense Releases The Latest Version Of Its Sensor Evaluation System For Autonomous Driving. https://www.robosense.ai/en/news-show-1473/, 2021.
[90] Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. https://www.sae.org/standards/content/j3016_202104/, 2021.
[91] Comma2k19 LD. https://www.kaggle.com/tkm2261/comma2k19-ld, 2022.

[92] ELP-USBFHD08S-MFV with 10x Optical Zoom. https://www.elpcctv.com/20-megapixel-high-speed-10x-zoom-550mm-webcam-usb-camera-for-live-driving-teaching-system-p-281.html, 2022.
[93] FBH905-10. https://www.thorlabs.co.jp/thorproduct.cfm?partnumber=FBH905-10, 2022.
[94] MidJourney. https://www.midjourney.com/, 2023.
[95] Our User Study Form: AI Images' Realism Survey. https://drive.google.com/file/d/1ifbOT8YE-iHfLvss_h4nzNsq6PZwOIwo/view?usp=drive_link, 2023.
[96] Antenna Design for Space. https://www.rantecantennas.com/blog/antenna-design-for-space/, 2024.
[97] Lens Characteristics: Flare, Ghosting and Aberration. https://av.jpn.support.panasonic.com/support/global/cs/dsc/knowhow/knowhow15.html, 2024.
[98] Our Proejct Website. https://sites.google.com/view/cav-sec/ndd-attack, 2024.
[99] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*, 2023.
[100] Adobe. Adobe Firefly - Generative AI for creatives. https://www.adobe.com/sensei/generative-ai/firefly.html, 2023.
[101] Adverb. BigSleep. https://github.com/lucidrains/big-sleep, 2021.
[102] G. Ahearn. Cameras that See Beyond Visible Light: Inspecting the Seen and Unseen. https://www.qualitymag.com/articles/96211, 2020.
[103] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A Next-Generation Hyperparameter Optimization Framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2019.
[104] F. Almutairy, T. Alshaabi, J. Nelson, and S. Wshah. ARTS: Automotive Repository of Traffic Signs for the United States. *IEEE Transactions on Intelligent Transportation Systems*, 2019.
[105] B. Amos, I. Jimenez, J. Sacks, B. Boots, and J. Z. Kolter. Differentiable MPC for end-to-end planning and control. In *NeurIPS*, 2018.
[106] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen, et al. Palm 2 Technical Report. *arXiv preprint arXiv:2305.10403*, 2023.
[107] Apollo Auto. Apollo Hardware Development Platform. https://developer.apollo.auto/platform/hardware.html, 2016.
[108] R. Araneta. Seeing the unseen: How infrared cameras capture beyond the visible. https://possibility.teledyneimaging.com/seeing-the-unseen-how-infrared-cameras-capture-beyond-the-visible/, 2019.
[109] A. Athalye, N. Carlini, and D. Wagner. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In *International Conference on Machine Learning (ICML)*, 2018.
[110] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok. Synthesizing Robust Adversarial Examples. In *International Conference on Machine Learning (ICML)*, 2018.
[111] M. Bai, G. Mattyus, N. Homayounfar, S. Wang, S. K. Lakshmikanth, and R. Urtasun. Deep Multi-Sensor Lane Detection. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3102–3109. IEEE, 2018.

[112] T. Bai, J. Zhao, J. Zhu, S. Han, J. Chen, B. Li, and A. Kot. AI-GAN: Attack-Inspired Generation of Adversarial Examples. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2543–2547. IEEE, 2021.

[113] M. Bansal, A. Krizhevsky, and A. S. Ogale. ChauffeurNet: Learning to Drive by Imitating the Best and Synthesizing the Worst. In *RSS*, 2019.

[114] B. E. Bayer. Color imaging array, July 20 1976. US Patent 3,971,065.

[115] C. Becker, L. J. Yount, S. Rozen-Levy, and J. D. Brewer. Functional Safety Assessment of an Automated Lane Centering System. In *National Highway Traffic Safety Administration*, 2018.

[116] K. Behrendt and R. Soussan. Unsupervised Labeled Lane Marker Dataset Generation Using Maps. In *IEEE International Conference on Computer Vision*, 2019.

[117] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for Hyper-Parameter Optimization. In *25th annual conference on neural information processing systems (NIPS 2011)*, volume 24. Neural Information Processing Systems Foundation, 2011.

[118] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al. End to End Learning for Self-Driving Cars. *arXiv preprint arXiv:1604.07316*, 2016.

[119] A. Boora, I. Ghosh, and S. Chandra. Identification of free flowing vehicles on two lane intercity highways under heterogeneous traffic condition. *Transportation Research Procedia*, 21:130–140, 2017.

[120] A. Brock, J. Donahue, and K. Simonyan. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *International Conference on Learning Representations (ICLR)*, 2019.

[121] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, et al. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. *arXiv preprint arXiv:2307.15818*, 2023.

[122] T. Brown, D. Mane, A. Roy, M. Abadi, and J. Gilmer. Adversarial Patch. *arXiv preprint arXiv:1712.09665*, 2017.

[123] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language Models Are Few-Shot Learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[124] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuScenes: A Multimodal Dataset for Autonomous Driving. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[125] Y. Cao, S. H. Bhupathiraju, P. Naghavi, T. Sugawara, Z. M. Mao, and S. Rampazzi. You Can't See Me: Physical Removal Attacks on LiDAR-based Autonomous Vehicles Driving Frameworks. In *USENIX Security Symposium*, 2023.

[126] Y. Cao, Y. Guo, T. Sato, Q. A. Chen, Z. M. Mao, and Y. Cheng. Demo: Remote Adversarial Attack on Automated Lane Centering. In *AutoSec*, 2022.

[127] Y. Cao, J. Ma, K. Fu, R. Sara, and M. Mao. Automated Tracking System for LiDAR Spoofing Attacks on Moving Targets. In *Proc. Workshop Automot. Auto. Vehicle Secur.(AutoSec)*, page 1, 2021.

[128] Y. Cao, N. Wang, C. Xiao, D. Yang, J. Fang, R. Yang, Q. A. Chen, M. Liu, and B. Li. Invisible for both Camera and LiDAR: Security of Multi-Sensor Fusion based Perception in Autonomous Driving Under Physical-World Attacks. In *IEEE Symposium on Security and Privacy (SP)*, 2021.

[129] Y. Cao, N. Wang, C. Xiao, D. Yang, J. Fang, R. Yang, Q. A. Chen, M. Liu, and B. Li. Invisible for Both Camera and LiDAR: Security of Multi-Sensor Fusion based Perception in Autonomous Driving under Physical-World Attacks. In *IEEE Symposium on Security and Privacy (SP)*, pages 176–194, 2021.

[130] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu, and Z. M. Mao. Adversarial Sensor Attack on Lidar-Based Perception in Autonomous Driving. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 2267–2281, 2019.

[131] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-End Object Detection with Transformers. In *European Conference on Computer Vision (ECCV)*, pages 213–229. Springer, 2020.

[132] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin. On Evaluating Adversarial Robustness. *arXiv preprint arXiv:1902.06705*, 2019.

[133] N. Carlini, J. Hayes, M. Nasr, M. Jagielski, V. Sehwag, F. Tramer, B. Balle, D. Ippolito, and E. Wallace. Extracting Training Data from Diffusion Models. *arXiv preprint arXiv:2301.13188*, 2023.

[134] N. Carlini and D. Wagner. Towards Evaluating the Robustness of Neural Networks. In *2017 IEEE symposium on security and privacy (SP)*, pages 39–57. IEEE, 2017.

[135] S. Casas, A. Sadat, and R. Urtasun. MP3: A Unified Model to Map, Perceive, Predict and Plan. In *CVPR*, 2021.

[136] G. Chang. Robust Kalman Filtering based on Mahalanobis Distance as Outlier Judging Criterion. *Journal of Geodesy*, 2014.

[137] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, et al. Argoverse: 3D Tracking and Forecasting with Rich Maps. In *CVPR*, 2019.

[138] J. Chen, H. Chen, K. Chen, Y. Zhang, Z. Zou, and Z. Shi. Diffusion Models for Imperceptible and Transferable Adversarial Attack. *arXiv preprint arXiv:2305.08192*, 2023.

[139] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[140] S. Chen, K. Ma, and Y. Zheng. Med3d: Transfer Learning for 3D Medical Image Analysis. *arXiv preprint arXiv:1904.00625*, 2019.

[141] S.-T. Chen, C. Cornelius, J. Martin, and D. H. P. Chau. Shapeshifter: Robust Physical Adversarial Attack on Faster R-CNN Object Detector. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 52–68. Springer, 2018.

[142] X. Chen, X. Gao, J. Zhao, K. Ye, and C.-Z. Xu. AdvDiffuser: Natural Adversarial Example Synthesis with Diffusion Models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4562–4572, 2023.

[143] Y. Chen, S. Liu, X. Shen, and J. Jia. Fast Point R-CNN. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[144] H.-D. Cheng, X. H. Jiang, Y. Sun, and J. Wang. Color image segmentation: advances and prospects. *Pattern recognition*, 34(12):2259–2281, 2001.

[145] M. Cheng, S. Singh, P. H. Chen, P.-Y. Chen, S. Liu, and C.-J. Hsieh. Sign-OPT: A Query-Efficient Hard-label Adversarial Attack. In *International Conference on Learning Representations (ICLR)*, 2020.

[146] 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)*, pages 132–137. IEEE, 2019.

[147] Y.-C. Chiu, H.-Y. Lin, and W.-L. Tai. A Two-Stage Learning Approach for Traffic Sign Detection and Recognition. In *7th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2021)*, pages 276–283, 2021.

[148] CivilLaser. 780nm 1W 2W Powerful IR Laser Module Dot With Cooling Fan. https://www.civillaser.com/index.php?main_page=product_info&products_id=477, 2018.

[149] J. Cohen, E. Rosenfeld, and Z. Kolter. Certified Adversarial Robustness via Randomized Smoothing. Proceedings of Machine Learning Research, pages 1310–1320. PMLR.

[150] M. Contributors. MMDetection3D: OpenMMLab Next-Generation Platform for Feneral 3D Object Detection. https://github.com/open-mmlab/mmdetection3d, 2020.

[151] O. contributors. Open Source Routing Machine. https://github.com/Project-OSRM/osrm-backend. Accessed: 2023-12.

[152] N. S. Council. *Reference Material for DDC Instructors, 5th Edition*. 2005.

[153] GM Safety Report 2018. https://www.gm.com/content/dam/company/docs/us/en/gmcom/gmsafetyreport.pdf.

[154] DALL-E-2. DALL-E-2. https://openai.com/dall-e-2, 2022.

[155] DALL-E-3. DALL-E-3. https://openai.com/dall-e-3, 2023.

[156] P. Del Moral. Nonlinear Filtering: Interacting Particle Resolution. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, 325(6):653–658, 1997.

[157] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A Large-Scale Hierarchical Image Database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE, 2009.

[158] Department of Transportation. Federal Highway Administration (FHWA). https://www.govinfo.gov/content/pkg/FR-2009-12-16/pdf/E9-28322.pdf, 2009.

[159] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, 2019.

[160] P. Dhariwal and A. Nichol. Diffusion Models Beat GANs on Image Synthesis, 2021.

[161] R. C. Dorf and R. H. Bishop. *Modern Control Systems*. Pearson, 2011.

[162] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An Open Urban Driving Simulator. In *CoRL*, 2017.

[163] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An Open Urban Driving Simulator. In *CoRL*, pages 1–16, 2017.

[164] R. Duan, X. Mao, A. K. Qin, Y. Chen, S. Ye, Y. He, and Y. Yang. Adversarial Laser Beam: Effective Physical-World Attack to DNNs in a Blink. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16062–16071, 2021.

[165] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy. A Study of the Effect of JPG Compression on Adversarial Images. *arXiv preprint arXiv:1608.00853*, 2016.

[166] C. Ertler, J. Mislej, T. Ollmann, L. Porzi, G. Neuhold, and Y. Kuang. The Mapillary Traffic Sign Dataset for Detection and Classification on a Global Scale. In *European Conference on Computer Vision (ECCV)*, pages 68–84, 2020.

[167] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, F. Tramer, A. Prakash, T. Kohno, and D. Song. Physical Adversarial Examples for Object Detectors. In *WOOT*, 2018.

[168] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[169] U. D. for Transport. *The Official Highway Code Book*. 2015.

[170] D. Fu, W. Lei, L. Wen, P. Cai, S. Mao, M. Dou, B. Shi, and Y. Qiao. LimSim++: A Closed-Loop Platform for Deploying Multimodal LLMs in Autonomous Driving. *arXiv preprint arXiv:2402.01246*, 2024.

[171] D. Fu, X. Li, L. Wen, M. Dou, P. Cai, B. Shi, and Y. Qiao. Drive Like a Human: Rethinking Autonomous Driving With Large Language Models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 910–919, 2024.

[172] C. Gackstatter, P. Heinemann, S. Thomas, and G. Klinker. Stable Road Lane Model Based on Clothoids. In *Advanced Microsystems for Automotive Applications 2010*, pages 133–143. Springer, 2010.

[173] Y. Ge, Y. Xiao, Z. Xu, X. Wang, and L. Itti. Contributions of Shape, Texture, and Color in Visual Recognition. In *European Conference on Computer Vision (ECCV)*, pages 369–386. Springer, 2022.

[174] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision Meets Robotics: The KITTI Dataset. *The International Journal of Robotics Research*, 2013.

[175] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[176] R. Girshick. Fast R-CNN. In *CVPR*, pages 1440–1448, 2015.

[177] R. C. Gonzales and P. Wintz. *Digital image processing*. Addison-Wesley Longman Publishing Co., Inc., 1987.

[178] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and Harnessing Adversarial Examples. *arXiv preprint arXiv:1412.6572*, 2014.

[179] J. Goodman. Statistical properties of laser speckle patterns. *Laser speckle and related phenomena*, pages 9–75, 1975.

[180] J. W. Goodman. Some fundamental properties of speckle. *JOSA*, 66(11):1145–1150, 1976.

[181] Google. Introducing Duet AI for Google Workspace. https://workspace.google.com/blog/product-announcements/duet-ai, 2023.

[182] S. Gowal, K. D. Dvijotham, R. Stanforth, R. Bunel, C. Qin, J. Uesato, R. Arandjelovic, T. Mann, and P. Kohli. Scalable Verified Training for Provably Robust Image

Classification. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4842–4851, 2019.

[183] M. S. Grewal and A. P. Andrews. Applications of Kalman Filtering in Aerospace 1960 to the Present [Historical Perspectives]. *IEEE Control Systems Magazine*, 30(3):69–78, 2010.

[184] K. Grill-Spector and R. Malach. The Human Visual Cortex. *Annu. Rev. Neurosci.*, 27:649–677, 2004.

[185] R. S. Hallyburton, Y. Liu, Y. Cao, Z. M. Mao, and M. Pajic. Security Analysis of Camera-LiDAR Fusion Against Black-Box Attacks on Autonomous Vehicles. In *USENIX Security Symposium*, 2022.

[186] E. Hamilton. JPEG File Interchange Format. 2004.

[187] S. Hari, Babu, L. Y.B., S. Ram, and K. Ashok. Airborne Infrared Search and Track Systems. *Defense Science Journal*, 57(5):739–753, 2007.

[188] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2003.

[189] A. Hata and D. Wolf. Road Marking Detection using LIDAR Reflective Intensity Data and Its Application to Vehicle Localization. In *ITSC*. IEEE, 2014.

[190] Z. Hau, K. Co, S. Demetriou, and E. Lupu. Object Removal Attacks on LiDAR-based 3D Object Detectors. In *NDSS Workshop on Automotive and Autonomous Vehicle Security (AutoSec)*, 2021.

[191] Z. Hau, S. Demetriou, L. Muñoz-González, and E. C. Lupu. Shadow-Catcher: Looking into Shadows to Detect Ghost Objects in Autonomous Vehicle 3D Sensing. In *European Symposium on Research in Computer Security*, pages 691–711. Springer, 2021.

[192] Y. Hayakawa, T. Sato, R. Suzuki, K. Ikeda, O. Sako, R. Nagata, Q. A. Chen, and K. Yoshioka. WIP: An Adaptive High Frequency Removal Attack to Bypass Pulse Fingerprinting in New-Gen LiDARs. 2024.

[193] J. Hayes. On Visible Adversarial Perturbations & Digital Watermarking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1597–1604, 2018.

[194] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *IEEE conference on computer vision and pattern recognition (CVPR)*, pages 770–778, 2016.

[195] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song. Natural Adversarial Examples. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15262–15271, 2021.

[196] A. B. Hillel, R. Lerner, D. Levi, and G. Raz. Recent Progress in Road and Lane Detection: A Survey. *Machine vision and applications*, 25(3):727–745, 2014.

[197] J. Ho, A. Jain, and P. Abbeel. Denoising Diffusion Probabilistic Models. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:6840–6851, 2020.

[198] Hocheol Shin and Dohyun Kim and Yujin Kwon and Yongdae Kim. Illusion and dazzle: Adversarial optical channel exploits against lidars for automotive applications. Cryptology ePrint Archive, Paper 2017/613, 2017. https://eprint.iacr.org/2017/613.

[199] Y. Hou, Z. Ma, C. Liu, T.-W. Hui, and C. C. Loy. Inter-Region Affinity Distillation for Road Marking Segmentation. In *CVPR*, 2020.

[200] Y. Hou, Z. Ma, C. Liu, and C. C. Loy. Learning Lightweight Lane Detection CNNs by Self Attention Distillation. In *CVPR*, 2019.

[201] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel. Detection of Traffic Signs in Real-World Images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks (IJCNN)*, number 1288, 2013.

[202] Y.-C. Hsu, Z. Xu, Z. Kira, and J. Huang. Learning to Cluster for Proposal-Free Instance Segmentation. In *IJCNN*, 2018.

[203] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely Connected Convolutional Networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4700–4708, 2017.

[204] D. Hunt, S. Luo, A. Khazraei, X. Zhang, S. Hallyburton, T. Chen, and M. Pajic. Radcloud: Real-time high-resolution point cloud generation using low-cost radars for aerial and ground vehicles. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2024.

[205] G. Ilharco, M. Wortsman, R. Wightman, C. Gordon, N. Carlini, R. Taori, A. Dave, V. Shankar, H. Namkoong, J. Miller, H. Hajishirzi, A. Farhadi, and L. Schmidt. OpenCLIP. https://doi.org/10.5281/zenodo.5143773, July 2021.

[206] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Adversarial Examples Are Not Bugs, They Are Features. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[207] A. Jain, L. Del Pero, H. Grimmett, and P. Ondruska. Autonomy 2.0: Why Is Self-Driving Always 5 Years Away? *arXiv preprint arXiv:2107.08142*, 2021.

[208] W. Jia, Z. Lu, H. Zhang, Z. Liu, J. Wang, and G. Qu. Fooling the Eyes of Autonomous Vehicles: Robust Physical Adversarial Examples Against Traffic Sign Recognition Systems. In *29th Annual Network and Distributed System Security Symposium (NDSS)*, 2022.

[209] Y. Jia, Y. Lu, J. Shen, Q. A. Chen, H. Chen, Z. Zhong, and T. Wei. Fooling Detection Alone is Not Enough: Adversarial Attack Against Multiple Object Tracking. In *International Conference on Learning Representations (ICLR)*, 2019.

[210] Y. Jia, Y. Lu, J. Shen, Q. A. Chen, H. Chen, Z. Zhong, and T. Wei. Fooling Detection Alone is Not Enough: Adversarial Attack Against Multiple Object Tracking. In *International Conference on Learning Representations (ICLR)*, 2020.

[211] Z. Jiayu, Z. Zheng, Y. Yun, and W. Xingang. DiffBEV: Conditional Diffusion Model for Bird's Eye View Perception. *arXiv preprint arXiv:2303.08333*, 2023.

[212] 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)*, pages 1822–1839. IEEE, 2023.

[213] Z. Jin, J. Xiaoyu, Y. Cheng, B. Yang, C. Yan, and W. Xu. PLA-LiDAR: Physical Laser Attacks against LiDAR-based 3D Object Detection in Autonomous Vehicle. In *IEEE Symposium on Security and Privacy (SP)*, pages 710–727, 2023.

[214] P. Jing, Q. Tang, Y. Du, L. Xue, X. Luo, T. Wang, S. Nie, and S. Wu. Too Good to Be Safe: Tricking Lane Detection in Autonomous Driving with Crafted Perturbations. In *USENIX Security*, 2021.

[215] G. Jocher. Ultralytics YOLOv5. https://github.com/ultralytics/yolov5, 2020.

[216] G. Jocher, A. Chaurasia, and J. Qiu. Ultralytics YOLO. https://github.com/ultralytics/ultralytics, 2023.
[217] F. Joublin, A. Ceravola, P. Smirnov, F. Ocker, J. Deigmoeller, A. Belardinelli, C. Wang, S. Hasler, D. Tanneberg, and M. Gienger. Copal: Corrective Planning of Robot Actions With Large Language Models. *arXiv preprint arXiv:2310.07263*, 2023.
[218] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 1960.
[219] S. Kato, S. Tokunaga, Y. Maruyama, S. Maeda, M. Hirabayashi, Y. Kitsukawa, A. Monrroy, T. Ando, Y. Fujii, and T. Azumi. Autoware On Board: Enabling Autonomous Vehicles with Embedded Systems. In *ICCPS'18*, pages 287–296. IEEE Press, 2018.
[220] A. Kazerouni, E. K. Aghdam, M. Heidari, R. Azad, M. Fayyaz, I. Hacihaliloglu, and D. Merhof. Diffusion Models in Medical Imaging: A Comprehensive Survey. *Medical Image Analysis*, page 102846, 2023.
[221] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet. Level 5 Perception Dataset. https://level-5.global/level5/data/, 2019.
[222] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representation (ICLR)*, 2015.
[223] Y. Ko, J. Jun, D. Ko, and M. Jeon. Key Points Estimation and Point Instance Segmentation Approach for Lane Detection, 2020.
[224] S. Köhler, G. Lovisotto, S. Birnbach, R. Baker, and I. Martinovic. They see me rollin': Inherent vulnerability of the rolling shutter in cmos image sensors. In *Annual Computer Security Applications Conference*, pages 399–413, 2021.
[225] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli. Kinematic and Dynamic Vehicle Models for Autonomous Driving Control Design. In *IV*, 2015.
[226] A. Krizhevsky. Learning Multiple Layers of Features from Tiny Images, 2009.
[227] A. Krizhevsky, G. Hinton, et al. Learning Multiple Layers of Features from Tiny Images. 2009.
[228] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial Examples in the Physical World. *arXiv preprint arXiv:1607.02533*, 2016.
[229] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom. PointPillars: Fast Encoders for Object Detection from Point Clouds. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12697–12705, 2019.
[230] Y. LeCun and C. Cortes. MNIST Handwritten Digit Database. 2010.
[231] J.-W. Lee and B. Litkouhi. A Unified Framework of the Automated Lane Centering/Changing Control for Motion Smoothness Adaptation. In *ITSC*, 2012.
[232] Leopard. LI-USB30-OV10635-GMSL-057H Camera Module. https://www.leopardimaging.com/product/autonomous-camera/maxim-gmsl-cameras/li-ov10635-gmsl/li-usb30-ov10635-gmsl-057h/, 2023.
[233] Leopard Imaging. LI-USB30-AR023ZWDR. https://www.mouser.com/datasheet/2/233/LI-USB30-AR023ZWDR_datasheet-1101519.pdf, 2016.
[234] A. Levine and S. Feizi. (de)randomized smoothing for certifiable defense against patch attacks. In *International Conference on Neural Information Processing Systems (NIPS)*, pages 6465–6475, 2020.

[235] B. Li, Y. Wang, J. Mao, B. Ivanovic, S. Veer, K. Leung, and M. Pavone. Driving Everywhere With Large Language Model Policy Adaptation. *arXiv preprint arXiv:2402.05932*, 2024.

[236] J. Li, X. Mei, D. Prokhorov, and D. Tao. Deep Neural Network for Structural Prediction and Lane Detection in Traffic Scene. *IEEE transactions on neural networks and learning systems*, 28(3):690–703, 2016.

[237] S. Li, A. Neupane, S. Paul, C. Song, S. V. Krishnamurthy, A. K. Roy-Chowdhury, and A. Swami. Stealthy Adversarial Perturbations Against Real-Time Video Classification Systems. In *26th Annual Network and Distributed System Security Symposium, NDSS*, 2019.

[238] X. Li, J. Li, X. Hu, and J. Yang. Line-CNN: End-to-End Traffic Line Detection with Line Proposal Unit. *ITSC*, 2019.

[239] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen. PointCNN: Convolution on X-Transformed Points. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018.

[240] Y. Li, F. Yang, Q. Liu, J. Li, and C. Cao. Light can be Dangerous: Stealthy and Effective Physical-world Adversarial Attack by Spot Light. *Computers & Security*, 132:103345, 2023.

[241] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu. Defense Against Adversarial Attacks Using High-Level Representation Guided Denoiser. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1778–1787, 2018.

[242] S. Liberatore. Tesla cars tricked into autonomously accelerating up to 85 MPH in a 35 MPH zone while in cruise control using just a two-inch strip of electrical tape. https://www.dailymail.co.uk/sciencetech/article-8021567/.html, 2020.

[243] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision (ECCV)*, pages 740–755, 2014.

[244] X. Ling, S. Ji, J. Zou, J. Wang, C. Wu, B. Li, and T. Wang. Deepsec: A Uniform Platform for Security Analysis of Deep Learning Model. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 673–690. IEEE, 2019.

[245] J. Liu and J.-M. Park. "Seeing is Not Always Believing": Detecting Perception Error Attacks Against Autonomous Vehicles. *IEEE Transactions on Dependable and Secure Computing*, 18(5):2209–2223, 2021.

[246] L. Liu, X. Chen, S. Zhu, and P. Tan. CondLaneNet: A Top-To-Down Lane Detection Framework Based on Conditional Convolution. In *ICCV*, 2021.

[247] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single Shot Multibox Detector. In *European Conference on Computer Vision (ECCV)*, pages 21–37. Springer, 2016.

[248] H. Loeb, A. Belwadi, J. Maheshwari, and S. Shaikh. Age and Gender Differences in Emergency Takeover from Automated to Manual Driving on Simulator. *Traffic injury prevention*, pages 1–3, 2019.

[249] G. Lovisotto, H. Turner, I. Sluganovic, M. Strohmeier, and I. Martinovic. SLAP: Improving Physical Adversarial Examples with Short-Lived Adversarial Perturbations. In *USENIX Security Symposium*, pages 1865–1882, 2021.

[250] C. Lyu, W. Zhang, H. Huang, Y. Zhou, Y. Wang, Y. Liu, S. Zhang, and K. Chen. RTMDet: An Empirical Study of Designing Real-Time Object Detectors. *arXiv preprint arXiv:2212.07784*, 2022.

[251] C. Ma, N. Wang, Q. A. Chen, and C. Shen. SlowTrack: Increasing the Latency of Camera-Based Perception in Autonomous Driving Using Adversarial Examples. *AAAI*, 2024.

[252] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representation (ICLR)*, 2018.

[253] Y. Man, M. Li, and R. Gerdes. GhostImage: Remote perception attacks against camera-based image classification systems. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*, pages 317–332, Oct. 2020.

[254] MarkLines Co., Ltd. BMW 320i Teardown: ADAS/onboard devices . https://www.marklines.com/en/report_all/rep2018_202004, 2020.

[255] D. Meng and H. Chen. Magnet: a Two-pronged Defense Against Adversarial Examples. In *ACM SIGSAC conference on computer and communications security*, pages 135–147, 2017.

[256] Microsoft. LifeCam HD-3000. https://www.microsoft.com/en/accessories/products/webcams/lifecam-hd-3000, 2023.

[257] N. Miura, T. Machida, K. Matsuda, M. Nagata, S. Nashimoto, and D. Suzuki. A Low-Cost Replica-based Distance-Spoofing Attack on Mmwave Fmcw Radar. In *Proceedings of the 3rd ACM Workshop on Attacks and Solutions in Hardware Security Workshop*, pages 95–100, 2019.

[258] MobilEye. About MobilEye. https://www.mobileye.com/about/, 2020.

[259] A. Mogelmose, M. M. Trivedi, and T. B. Moeslund. Vision-based Traffic Sign Detection and Analysis for Intelligent Driver Assistance Systems: Perspectives and Survey. *IEEE Transactions on Intelligent Transportation Systems*, 2012.

[260] F. Morales. A Packaged and Flexible Version of the CRAFT Text Detector and Keras CRNN Recognition Model. https://github.com/faustomorales/keras-ocr, 2022.

[261] B. Nassi, D. Nassi, R. Ben-Netanel, Y. Mirsky, O. Drokin, and Y. Elovici. Phantom of the ADAS: Phantom Attacks on Driver-Assistance Systems. *IACR Cryptol. ePrint Arch.*, 2020:85, 2020.

[262] B. Nassi, D. Nassi, R. Ben-Netanel, Y. Mirsky, O. Drokin, and Y. Elovici. Phantom of the ADAS: Phantom Attacks on Driver-Assistance Systems. *IACR Cryptol. ePrint Arch.*, 2020:85, 2020.

[263] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool. Towards End-to-End Lane Detection: An Instance Segmentation Approach. In *IV*, 2018.

[264] S. of California Department of Motor Vehicles. *California Commercial Driver Handbook: Section 2 – Driving Safely*. 2019. Available at https://www.dmv.ca.gov/portal/uploads/2020/06/comlhdbk.pdf.

[265] A. A. of State Highway and T. O. (AASHTO). *Policy on Geometric Design of Highways and Streets (7th Edition)*. American Association of State Highway and Transportation Officials (AASHTO), 2018.

[266] OmniVision. OV10635 Image Sensor. https://www.ovt.com/products/ov10635-n29y-pb/, 2018.

[267] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang. Spatial as Deep: Spatial CNN for Traffic Scene Understanding. In *AAAI*, 2018.

[268] T. Pearce. Conditional Diffusion MNIST. https://github.com/TeaPearce/Conditional_Diffusion_MNIST, 2022.

[269] K. Pei, Y. Cao, J. Yang, and S. Jana. Deepxplore: Automated Whitebox Testing of Deep Learning Systems. In *Symposium on Operating Systems Principles*, pages 1–18, 2017.

[270] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl. Remote Attacks on Automated Vehicles Sensors: Experiments on Camera and Lidar. *Black Hat Europe*, 11:2015, 2015.

[271] J. Philion. FastDraw: Addressing the Long Tail of Lane Detection by Adapting a Sequential Prediction Network. In *CVPR*, 2019.

[272] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

[273] R. Qian, X. Lai, and X. Li. 3D Object Detection for Autonomous Driving: A Survey. *Pattern Recognition*, 2022.

[274] Qin, Zequn and Wang, Huanyu and Li, Xi. Ultra Fast Structure-Aware Deep Lane Detection. In *ECCV*, 2020.

[275] Z. Qu, H. Jin, Y. Zhou, Z. Yang, and W. Zhang. Focus on Local: Detecting Lane Marker From Bottom Up via Key Point. In *CVPR*, 2021.

[276] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning Transferable Visual Models from Natural Language Supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021.

[277] R. Rajamani. *Vehicle Dynamics and Control*. Springer Science & Business Media, 2011.

[278] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical Text-Conditional Image Generation with Clip Latents. *arXiv preprint arXiv:2204.06125*, 2022.

[279] Raspberry Pi Ltd. Raspberry Pi High Quality Camera. https://datasheets.raspberrypi.com/hq-camera/hq-camera-product-brief.pdf, 2023.

[280] F. Reda, J. Kontkanen, E. Tabellion, D. Sun, C. Pantofaru, and B. Curless. FILM: Frame Interpolation for Large Motion. In *European Conference on Computer Vision (ECCV)*, 2022.

[281] J. Redmon and A. Farhadi. YOLO9000: Better, Faster, Stronger. In *IEEE conference on computer vision and pattern recognition (CVPR)*, 2017.

[282] J. Redmon and A. Farhadi. YOLOv3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[283] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NeurIPS*, 2015.

[284] Richalet, J. and Rault, A. and Testud, J. L. and Papon, J. Paper: Model Predictive Heuristic Control. *Automatica*, 14(5):413–428, Sept. 1978.

[285] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022.

[286] R. Roriz, J. Cabral, and T. Gomes. Automotive LiDAR Technology: A Survey. *IEEE TITS*, 2021.

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

[288] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, et al. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[289] T. Sato, S. H. V. Bhupathiraju, M. Clifford, T. Sugawara, Q. A. Chen, and S. Rampazzi. Invisible Reflections: Leveraging Infrared Laser Reflections to Target Traffic Sign Perception. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2024.

[290] T. Sato, Y. Hayakawa, R. Suzuki, Y. Shiiki, K. Yoshioka, and Q. A. Chen. LiDAR Spoofing Meets the New-Gen: Capability Improvements, Broken Assumptions, and New Attack Strategies. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2024.

[291] 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. *USENIX Security Symposium*, 2021.

[292] 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 *USENIX Security Symposium*, 2021.

[293] T. Sato, J. Shen, N. Wang, Y. J. Jia, X. Lin, and Q. A. Chen. Dirty Road Can Attack: Security of Deep Learning based Automated Lane Centering under Physical-World Attack. *arXiv:2009.06701*, 2021.

[294] H. Schafer, E. Santana, A. Haden, and R. Biasini. A Commute in Data: The comma2k19 Dataset. *arXiv preprint arXiv:1812.05752*, 2018.

[295] M. Serkan and H. Kirkici. Reshaping of a divergent elliptical gaussian laser beam into a circular, collimated, and uniform beam with aspherical lens design. *IEEE Sensors Journal*, 9(1):36–44, 2009.

[296] R. Shapley and P. Lennie. Spatial frequency analysis in the visual system. *Annual review of neuroscience*, 8(1):547–581, 1985.

[297] S. Sharan, F. Pittaluga, M. Chandraker, et al. LLM-Assist: Enhancing Closed-Loop Planning With Language-Based Reasoning. *arXiv preprint arXiv:2401.00125*, 2023.

[298] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter. Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition. In *ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 1528–1540, 2016.

[299] S. Sharma. GTSRB - CNN (98% Test Accuracy). https://www.kaggle.com/code/shivank856/gtsrb-cnn-98-test-accuracy, 2021.

[300] J. Shen, N. Wang, Z. Wan, Y. Luo, T. Sato, Z. Hu, X. Zhang, S. Guo, Z. Zhong, K. Li, Z. Zhao, C. Qiao, and Q. A. Chen. SoK: On the Semantic AI Security in Autonomous Driving. *arXiv preprint arXiv:2203.05314*, 2022.

[301] 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 *USENIX Security Symposium*, 2020.

[302] 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 *USENIX Security*, 2020.

[303] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10529–10538, 2020.

[304] S. Shi, X. Wang, and H. Li. PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–779, 2019.

[305] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li. From Points to Parts: 3D Object Detection from Point Cloud with Part-Aware and Part-Aggregation Network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(8):2647–2664, 2020.

[306] H. Shin, D. Kim, Y. Kwon, and Y. Kim. Illusion and Dazzle: Adversarial Optical Channel Exploits Against Lidars for Automotive Applications. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 445–467. Springer, 2017.

[307] P. Smuda, R. Schweiger, H. Neumann, and W. Ritter. Multiple Cue Data Fusion With Particle Filters for Road Course Detection in Vision Systems. In *2006 IEEE Intelligent Vehicles Symposium*, pages 400–405. IEEE, 2006.

[308] C. G. Someda. *Electromagnetic waves*. CRC Press, 2017.

[309] Sony Semiconductor Solutions Corporation. Sony IMX477-AACK Image Sensor. https://www.sony-semicon.com/files/62/pdf/p-13_IMX477-AACK_Flyer.pdf, 2018.

[310] G. Srinivasan and G. Shobha. Statistical texture analysis. In *Proceedings of world academy of science, engineering and technology*, volume 36, pages 1264–1269, 2008.

[311] StabilityAI. DeepFloyd IF. https://github.com/deep-floyd/IF, 2023.

[312] J. Sun, Y. Cao, Q. A. Chen, and Z. M. Mao. Towards Robust LiDAR-based Perception in Autonomous Driving: General Black-box Adversarial Sensor Attack and Countermeasures. In *USENIX Security Symposium*, 2020.

[313] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[314] R. Suzuki, T. Sato, Y. Hayakawa, K. Ikeda, O. Sako, R. Nagata, Q. A. Chen, and K. Yoshioka. WIP: Towards Practical LiDAR Spoofing Attack against Vehicles Driving at Cruising Speeds. 2024.

[315] O. Svelto, D. C. Hanna, et al. *Principles of lasers*, volume 1. Springer, 2010.

[316] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing Properties of Neural Networks. In *ICLR*, 2014.

[317] L. Tabelini, R. Berriel, T. M. P. ao, C. Badue, A. F. D. Souza, and T. Oliveira-Santos. Keep your Eyes on the Lane: Real-time Attention-guided Lane Detection. In *CVPR*, 2021.

[318] L. Tabelini, R. Berriel, T. M. Paixao, C. Badue, A. F. De Souza, and T. Oliveira-Santos. Polylanenet: Lane Estimation via Deep Polynomial Regression. In *ICPR*, 2021.

[319] M. Tan and Q. Le. Efficientnet: Rethinking Model Scaling for Convolutional Neural Networks. In *International Conference on Machine Learning (ICML)*, pages 6105–6114, 2019.

[320] R. T. Tan. Specularity, specular reflectance. In *Computer Vision: A Reference Guide*, pages 1185–1188. Springer, 2021.

[321] S. Tanaka, K. Yamada, T. Ito, and T. Ohkawa. Vehicle Detection Based on Perspective Transformation Using Rear-View Camera. *Hindawi Publishing Corporation International Journal of Vehicular Technology*, 9, 03 2011.

[322] J. Tang, S. Li, and P. Liu. A Review of Lane Detection Methods Based on Deep Learning. *Pattern Recognition*, 111:107623, 2021.

[323] K. Tang, J. Shen, and Q. A. Chen. Fooling Perception via Location: A Case of Region-of-Interest Attacks on Traffic Light Detection in Autonomous Driving. In *Workshop on Automotive and Autonomous Vehicle Security (AutoSec)*, 2021.

[324] Y. Tassa, N. Mansard, and E. Todorov. Control-limited differential dynamic programming. In *ICRA*, 2014.

[325] R. Thakur. Infrared sensors for autonomous vehicles. In *Recent Development in Optoelectronic Devices*, chapter 5. IntechOpen, Rijeka, 2017.

[326] The University of Chicago. Laser Safety Calculations Formulas for Calculating the MPE, NOHD, and NHZ. https://d3qi0qp55mx5f5.cloudfront.net/researchsafety/docs/Laser_Safety_Calculations.pdf?mtime=1610127144, 2023.

[327] H. Tian, B. Fowler, and A. Gamal. Analysis of Temporal Noise in CMOS Photodiode Active Pixel Sensor. *IEEE Journal of Solid-State Circuits*, 36(1):92–101, 2001.

[328] Y. Tian, K. Pei, S. Jana, and B. Ray. Deeptest: Automated Testing of Deep-Neural-Network-Driven Autonomous Cars. In *International Conference on Software Engineering*, pages 303–314, 2018.

[329] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288*, 2023.

[330] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288*, 2023.

[331] F. Tramer, N. Carlini, W. Brendel, and A. Madry. On Adaptive Attacks to Adversarial Example Defenses. *arXiv preprint arXiv:2002.08347*, 2020.

[332] T. Trippel, O. Weisse, W. Xu, P. Honeyman, and K. Fu. WALNUT: Waging Doubt on the Integrity of MEMS Accelerometers with Acoustic Injection Attacks. In *EuroS&P*, pages 3–18. IEEE, 2017.

[333] Y. Tu, Z. Lin, I. Lee, and X. Hei. Injected and Delivered: Fabricating Implicit Control over Actuation Systems by Spoofing Inertial Sensors. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1545–1562, 2018.

[334] UK ACPO Road Policing Enforcement Technology Committee. *ACPO Code of Practice for Operational Use of Enforcement Equipment*. 2002.

[335] C. Urmson, J. A. Bagnell, C. Baker, M. Hebert, A. Kelly, R. Rajkumar, P. E. Rybski, S. Scherer, R. Simmons, S. Singh, et al. Tartan Racing: A Multi-Modal Approach to the DARPA Urban challenge. 2007.

[336] F. Van Diggelen and P. Enge. The World's First GPS MOOC and Worldwide Laboratory using Smartphones. In *ION GNSS+*, 2015.

[337] D. Wang, C. Watkins, and H. Xie. MEMS Mirrors for LiDAR: A Review. *Micromachines*, 11(5):456, 2020.

[338] D. Wang, W. Yao, T. Jiang, C. Li, and X. Chen. RFLA: A Stealthy Reflected Light Adversarial Attack in the Physical World, 2023.

[339] J. Wang and Y. Su. Fast detection of gpr objects with cross correlation and hough transform. *Progress In Electromagnetics Research C*, 38:229–239, 2013.

[340] N. Wang, Y. Luo, T. Sato, K. Xu, and Q. A. Chen. Does Physical Adversarial Example Really Matter to Autonomous Driving? Towards System-Level Effect of Adversarial Object Evasion Attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4412–4423, October 2023.

[341] W. Wang, Y. Yao, X. Liu, X. Li, P. Hao, and T. Zhu. I Can See the Light: Attacks on Autonomous Vehicles Using Invisible Lights. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1930–1944, 2021.

[342] Z. Wang, W. Ren, and Q. Qiu. LaneNet: Real-Time Lane Detection Networks for Autonomous Driving. *arXiv preprint arXiv:1807.01726*, 2018.

[343] D. Watzenig and M. Horn. *Automated Driving: Safer and More Efficient Future Driving*. Springer, 2016.

[344] Waymo Safety Report 2021. https://storage.googleapis.com/waymo-uploads/files/documents/safety/2021-08-waymo-safety-report.pdf.

[345] L. Wen, D. Fu, X. Li, X. Cai, T. Ma, P. Cai, M. Dou, B. Shi, L. He, and Y. Qiao. Dilu: A Knowledge-Driven Approach to Autonomous Driving With Large Language Models. *arXiv preprint arXiv:2309.16292*, 2023.

[346] C. Xiang, A. N. Bhagoji, V. Sehwag, and P. Mittal. PatchGuard: A Provably Robust Defense against Adversarial Patches via Small Receptive Fields and Masking. In *USENIX Security Symposium*, pages 2237–2254, 2021.

[347] C. Xiang, S. Mahloujifar, and P. Mittal. PatchCleanser: Certifiably Robust Defense against Adversarial Patches for Any Image Classifier. In *USENIX Security Symposium*, 2022.

[348] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-Mnist: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[349] C. Xie, Y. Wu, L. v. d. Maaten, A. L. Yuille, and K. He. Feature Denoising for Improving Adversarial Robustness. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 501–509, 2019.

[350] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated Residual Transformations for Deep Neural Networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1492–1500, 2017.

[351] Y. Xingwei, L. Dawei, Z. Jun, and W. Jianwei. Radar Jamming Detection based on Approximate Entropy and Moving-cut Approximate Entropy. 2012.

[352] W. Xu, D. Evans, and Y. Qi. Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks. *arXiv preprint arXiv:1704.01155*, 2017.

[353] C. Yan, W. Xu, and J. Liu. Can You Trust Autonomous Cehicles: Contactless Attacks Against Sensors of Self-Driving Vehicle. *DEF CON*, 24(8):109, 2016.

[354] C. Yan, Z. Xu, Z. Yin, X. Ji, and W. Xu. Rolling Colors: Adversarial Laser Exploits against Traffic Light Recognition. In *31st USENIX Security Symposium*, pages 1957–1974, 2022.

[355] Y. Yan, Y. Mao, and B. Li. SECOND: Sparsely Embedded Convolutional Detection. *Sensors*, 18(10):3337, 2018.

[356] Z. Yang. Fast template matching based on normalized cross correlation with centroid bounding. In *2010 International Conference on Measuring Technology and Mechatronics Automation*, volume 2, pages 224–227. IEEE, 2010.

[357] Z. Yang, Y. Sun, S. Liu, and J. Jia. 3DSSD: Point-based 3D Single Stage Object Detector. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11040–11048, 2020.

[358] C.-H. Yeh, P.-Y. Sung, C.-H. Kuo, and R.-N. Yeh. Robust laser speckle recognition system for authenticity identification. *Optics express*, 20(22):24382–24393, 2012.

[359] P. yeh Chiang, R. Ni, A. Abdelkader, C. Zhu, C. Studor, and T. Goldstein. Certified Defenses for Adversarial Patches. In *International Conference on Learning Representations (ICLR)*, 2020.

[360] S. Yenikaya, G. Yenikaya, and E. Düven. Keeping the Vehicle on the Road - A Survey on On-Road Lane Detection Systems. *ACM Computing Surveys (CSUR)*, 46(1):1–43, 2013.

[361] S. Yoo, H. S. Lee, H. Myeong, S. Yun, H. Park, J. Cho, and D. H. Kim. End-to-End Lane Marker Detection via Row-Wise Classification. In *CVPR Workshops*, 2020.

[362] J. Yoshida. Teardown: Lessons Learned From Audi A8. https://www.eetasia.com/teardown-lessons-learned-from-audi-a8/, 2020.

[363] K. Yoshioka. A Tutorial and Review of Automobile Direct LiDAR SoCs: Evolution of Next-Generation LiDARs. *IEICE Transactions on Electronics*, E105.C(10):534–543, 2022.

[364] C. Yu, J. Chen, Y. Xue, Y. Liu, W. Wan, J. Bao, and H. Ma. Defending against Universal Adversarial Patches by Clipping Feature Norms. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16434–16442, 2021.

[365] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell. Bdd100k: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In *CVPR*, 2020.

[366] M. Yu, M. Shi, W. Hu, and L. Yi. FPGA-based Dual-pulse Anti-interference Lidar System Using Digital Chaotic Pulse Position Modulation. *IEEE Photonics Technology Letters*, 2021.

[367] L. Yufeng, Y. Fengyu, L. Qi, L. Jiangtao, and C. Chenhong. Light can be Dangerous: Stealthy and Effective Physical-world Adversarial Attack by Spot Light. *Computers & Security*, page 103345, 2023.

[368] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda. A Survey of Autonomous Driving: Common Practices and Emerging Technologies. *IEEE access*, 8:58443–58469, 2020.

[369] L. Zhang and M. Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models. *arXiv preprint arXiv:2302.05543*, 2023.

[370] X. Zhang, N. Wang, H. Shen, S. Ji, X. Luo, and T. Wang. Interpretable Deep Learning under Fire. In *USENIX Security Symposium*, 2020.

[371] Y. Zhang and P. Liang. Defending Against Whitebox Adversarial Attacks via Randomized Discretization. volume 89 of *Proceedings of Machine Learning Research*, pages 684–693. PMLR, 16–18 Apr 2019.

[372] D. Zhao, Y. Guo, and Y. J. Jia. Trafficnet: An open naturalistic driving scenario library. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8. IEEE, 2017.

[373] Y. Zhao, H. Zhu, R. Liang, Q. Shen, S. Zhang, and K. Chen. Seeing isn't Believing: Practical Adversarial Attack Against Object Detectors. In *ACM SIGSAC Conference on Computer and Communications Security (ACM CCS)*, page 1989–2004, 2019.

[374] T. Zheng, H. Fang, Y. Zhang, W. Tang, Z. Yang, H. Liu, and D. Cai. RESA: Recurrent Feature-Shift Aggregator for Lane Detection, 2020.

[375] T. Zheng, H. Fang, Y. Zhang, W. Tang, Z. Yang, H. Liu, and D. Cai. RESA: Recurrent Feature-Shift Aggregator for Lane Detection. *AAAI*, 2021.

[376] Y. Zhong, X. Liu, D. Zhai, J. Jiang, and X. Ji. Shadows can be Dangerous: Stealthy and Effective Physical-world Adversarial Attack by Natural Phenomenon. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15345–15354, 2022.

[377] Z. Zhong, W. Xu, Y. Jia, and T. Wei. Perception Deception: Physical Adversarial Attack Challenges and Tactics for DNN-Based Object Detection. In *Black Hat Europe*, 2018.

[378] H. Zhou, W. Li, Y. Zhu, Y. Zhang, B. Yu, L. Zhang, and C. Liu. Deepbillboard: Systematic Physical-World Testing of Autonomous Driving Systems. In *International Conference on Software Engineering*, 2020.

[379] Y. Zhou and O. Tuzel. VoxelNet: End-to-End Learning for Point Cloud based 3D Object Detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4490–4499, 2018.

[380] Z. Zhou, D. Tang, X. Wang, W. Han, X. Liu, and K. Zhang. Invisible Mask: Practical Attacks on Face Recognition with Infrared. *arXiv preprint arXiv:1803.04683*, 2018.

[381] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang. Robust Lane Detection From Continuous Driving Scenes Using Deep Neural Networks. *IEEE Transactions on Vehicular Technology*, PP:1–1, 10 2019.