An Online Defense against Object-Based LiDAR Attacks in Autonomous Driving

Yan Zhang[*]
Iowa State University
Ames, Iowa, USA
yanzh@iastate.edu

Zihao Liu[*]
Iowa State University
Ames, Iowa, USA
zihaoliu@iastate.edu

Chongliu Jia
Iowa State University
Ames, Iowa, USA
jcl0618@iastate.edu

Yi Zhu
Wayne State University
Detroit, Michigan, USA
yzhu39@wayne.edu

Chenglin Miao
Iowa State University
Ames, Iowa, USA
cmiao@iastate.edu

Abstract

LiDAR (Light Detection and Ranging) has been widely used in autonomous driving to perceive the surrounding environment of self-driving cars. Advanced LiDAR perception systems typically leverage deep neural networks (DNNs) to achieve high performance. However, the vulnerability of DNNs to malicious attacks provides attackers with the means to compromise the LiDAR perception system, potentially causing traffic accidents. Recently, object-based attacks against LiDAR perception systems have drawn significant attention. In such attacks, the attacker can easily fool the LiDAR perception system by placing physical objects within the driving environment. Despite the practicality of these attacks and their potential catastrophic consequences in autonomous driving, there is currently no effective and practical defense against them. To address this issue, we propose a novel online defense mechanism against object-based LiDAR attacks. This mechanism operates in an online manner, aiming to identify and remove the adversarial LiDAR points generated by the objects used by attackers before the data is fed into the perception module of autonomous driving systems. It is not only effective and efficient for real-world autonomous driving but also attack-agnostic and capable of identifying adversarial objects used by attackers. Extensive experiments in both simulated environments and real-world scenarios using a LiDAR perception testbed demonstrate the effectiveness and practicability of the proposed defense.

CCS Concepts

• **Security and privacy → Domain-specific security and privacy architectures**; • **Computer systems organization → Embedded and cyber-physical systems**.

Keywords

Online defense, LiDAR attacks, autonomous driving

ACM Reference Format:
Yan Zhang, Zihao Liu, Chongliu Jia, Yi Zhu, and Chenglin Miao. 2024. An Online Defense against Object-Based LiDAR Attacks in Autonomous Driving. In *ACM Conference on Embedded Networked Sensor Systems (SenSys '24), November 4–7, 2024, Hangzhou, China.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3666025.3699345

1 Introduction

In recent years, autonomous driving has garnered significant attention [2, 8, 20–22, 29, 38, 47, 48], and numerous autonomous vehicles have been operating on public roads. A key factor in the successful deployment of these vehicles is the integration of advanced sensors, such as cameras, LiDAR, and radar, which provide them with robust capabilities for perceiving their surrounding environments. Among these sensors, LiDAR stands out for its attractive characteristics and has been widely adopted by today's autonomous vehicles. It can not only create high-resolution 3D point clouds of the environment, providing precise information about objects on the road, but also perform well in various light and weather conditions.

State-of-the-art LiDAR perception systems in autonomous driving primarily rely on deep learning models to achieve outstanding performance [24, 39, 51]. However, deep neural networks (DNNs) have been shown to be vulnerable to malicious attacks, where an attacker can easily alter the model output by either adding small perturbations to the test data [14, 15, 44, 44, 49] or poisoning the training set [32, 55, 56]. This vulnerability provides attackers with an opportunity to compromise the deep learning models used by LiDAR perception systems in autonomous driving, potentially causing traffic accidents. Recently, many attack methods against LiDAR perception have been developed, which can be broadly classified into two categories: *laser-based attacks* and *object-based attacks.*

Laser-based attacks [3, 5, 16, 17, 35, 41, 43] aim to spoof LiDAR perception systems by using specialized devices to shoot lasers at LiDAR sensors, injecting spoofed LiDAR data points. Although laser-based attacks can achieve good attack performance, they typically require the laser to be transmitted to the sensor with high precision, which poses practicality issues given the unpredictable driving behavior of the victim autonomous vehicle. In contrast, object-based attacks [1, 4, 6, 30, 45, 56–58, 60] do not require specialized laser devices. Instead, attackers achieve their goals by placing physical objects at specific locations within the driving environment. These objects can be common items such as boxes and advertisement

[*] The first two authors contributed equally to this work.

SenSys '24, November 4–7, 2024, Hangzhou, China

ACM ISBN 979-8-4007-0697-4/24/11
https://doi.org/10.1145/3666025.3699345

boards, or objects with specific shapes easily created using 3D printers. For instance, an attacker can place a box on the rooftop of a car to hide it from the LiDAR perception system of an autonomous vehicle [45, 56]. Due to their high flexibility and low cost, object-based attacks are more practical in real-world driving environments and pose greater security threats to autonomous vehicles.

Despite the great practicability of object-based LiDAR attacks and their potential catastrophic consequences in autonomous driving, there is currently no effective and practical defense against them. Although some potential defense mechanisms have been proposed alongside the above attacks, they are insufficient to mitigate the threats effectively and typically have the following limitations. First, most of these defenses consider specific object-based attacks during the offline training of LiDAR perception models to make the trained models more robust to such attacks [45, 56, 60]. However, these approaches assume that the defender has knowledge of the attack methods being used, which is often impractical since it is usually difficult for the defender to know which object-based attack method the attacker will use. More importantly, these approaches have been shown in [45, 56, 60] to be ineffective in decreasing the attack success rates. Second, some potential defenses are based on multi-sensor fusion, involving additional sensors such as cameras and radar [56–58]. However, these defenses do not fundamentally address the vulnerability of LiDAR perception and increase the cost of autonomous vehicles. Moreover, existing studies have demonstrated that camera perception systems and radar are also susceptible to malicious attacks [23, 33, 50, 59]. Third, although some work proposes using 3D shadows to detect objects that attackers want to hide, this method has significant computational overhead and cannot recognize the type of object on the road [18]. Lastly, none of the existing defenses can identify the locations of the adversarial objects used by attackers, which are crucial for attack investigations and improving LiDAR perception in autonomous driving.

The above limitations raise an important quesiton: *Is it possible to design a defense mechanism against object-based LiDAR attacks that is not only effective and efficient for real-world autonomous driving but also attack-agnostic and capable of identifying adversarial objects used by attackers*? To answer this question, we need to consider the following challenges in designing an effective and practical defense.

First, although all object-based attacks achieve their goal by placing physical objects within the driving environment, their implementation methods can vary significantly. These different methods result in variations in the number of objects used by the attacker, as well as their locations, sizes, and shapes. Additionally, the defender typically does not know which attack method will be employed. Designing a unified defense that is effective against various object-based attacks without prior knowledge of the specific method being used is a challenging task. Second, the 3D LiDAR point cloud in autonomous driving is highly complex. It contains 3D points generated by all items surrounding the autonomous vehicle. Adversarial objects used by the attacker can be placed anywhere within this large 3D space, making it difficult to locate these objects and eliminate their impact on perception results. Third, many LiDAR sensors used in autonomous driving have a limited effective detection range, which implies that the defense mechanism must complete its task within a short range to prevent potential dangers. Given that an autonomous vehicle is often moving before recognizing any danger, there is very limited time for the defense to respond to an attack. Therefore, the defense mechanism must be efficient enough to prevent potential traffic accidents as intended by the attacker.

To address the above challenges, we propose a novel defense mechanism against object-based attacks in autonomous driving. This mechanism operates in an online manner and aims to process the collected LiDAR data to mitigate potential threats before the data is fed into the perception module of autonomous driving systems. The proposed defense mechanism consists of two stages: *suspicious point cluster extraction* and *attack detection*. The underlying philosophy is to emulate the reactions of a human driver when encountering dangers on the road. The goal of the first stage is to extract the suspicious point cluster in front of the autonomous vehicle, while the second stage aims to identify and remove the adversarial LiDAR points generated by the objects used by the attacker in this cluster. To enable an effective and efficient search for the adversarial LiDAR points in the second stage, we propose a reinforcement learning-based search method. This method can efficiently identify those points without requiring any prior knowledge about the number of objects used by the attacker, their locations, shapes, or sizes. Thus, the proposed online defense can be easily integrated into existing autonomous driving systems and is practical enough for real-world autonomous driving.

The performance of the proposed defense is evaluated in both simulated environments and real-world scenarios using a LiDAR perception testbed. The experimental results demonstrate that our defense can efficiently detect object-based LiDAR attacks with a high detection rate. Additionally, the results indicate that it can effectively prevent the potential dangers posed by these attacks. *To the best of our knowledge, this is the first defense against object-based LiDAR attacks that is not only effective and efficient for real-world autonomous driving but also attack-agnostic and capable of identifying adversarial objects used by attackers.*

2 Background and Related work

2.1 LiDAR Perception in Autonomous Driving

LiDAR has played a crucial role in the advancement of autonomous driving [46]. Many autonomous vehicles use LiDAR to perceive their surroundings and detect objects such as cars and pedestrians on the road. The output from a LiDAR sensor is a point cloud, a collection of data points representing the 3D environment, with each point characterized by its 3D coordinates and reflection intensity. From this point cloud, LiDAR detection systems can generate bounding boxes that precisely define the position, size, and orientation of detected objects. Existing LiDAR detection systems usually follow a structured pipeline that includes three main modules to identify objects [54]: sensor data representation, feature extraction, and core object detection. In the sensor data representation module, the raw point cloud data is transformed into structured formats, making it more compact and manageable. This transformation facilitates the subsequent feature extraction module, where high-dimensional, rich features are derived from the data. Finally, in the core object detection module, these features are processed to produce the bounding boxes that define the location, size, and orientation of the detected objects in 3D space.

To deal with highly complex point cloud data, advanced LiDAR object detection systems typically leverage deep learning to achieve good performance. Based on how point cloud data is transformed for processing, existing deep learning-based LiDAR object detection models can be divided into three categories: projection-based, voxel-based, and point-based models [56]. Projection-based models [28, 31, 51] convert point clouds into 2D images and use 2D convolutional neural networks (CNNs) for object detection. Voxel-based models [10, 19, 24, 26, 27, 37] partition point clouds into voxel grids, which are processed using 3D CNNs. Point-based models [7, 9, 39, 40, 53] operate directly on raw point clouds, employing point-wise neural network operations for object detection.

2.2 Object-Based Attacks against LiDAR Perception

Although DNNs enable LiDAR perception systems to achieve high accuracy, their vulnerability to malicious attacks allows attackers to mislead the system by slightly altering the sensing scenario. One of the most threatening attack types against LiDAR perception in autonomous driving is the object-based attack, where an attacker aims to mislead the perception system into making incorrect predictions by strategically placing physical objects within the driving environment. The basic idea behind such attacks is to use physical objects to introduce additional adversarial LiDAR points to fool the deep learning model used by the perception system. Figure 1 shows two examples of object-based LiDAR attacks. In these examples, the attacker aims to hide a car from the LiDAR perception system of an autonomous vehicle. In Figure 1a, the attacker places an object on the rooftop of the car to hide it. In Figure 1b, the attacker places two objects around the car. Object-based LiDAR attacks can utilize either objects with specific shapes or common objects.

Attacks using objects with specific shapes [1, 4, 6, 45, 57]. These attacks aim to optimize the shape of the object so that the LiDAR perception system can be misled when the object is placed within the driving environment. The shape is often uncommon in reality. It is usually derived digitally and then the corresponding object is generated using a 3D printer in the physical world. Cao et al. [6] introduce the first attack against LiDAR perception in autonomous driving using an adversarial object, deriving its specific shape with a LiDAR render. However, this attack is not universal and may not be reused in different 3D Scenes. To address this issue, Tu et al. [45] propose a universal attack against LiDAR perception by considering different scenes and vehicles when generating the adversarial object digitally. Figure 1a shows an example of the adversarial object generated in [45]. Although such an object can be used in various scenes, it has limitations in physical robustness, and its special shape may cause some errors in the obtained point cloud of the object during the LiDAR capturing process. To enhance the robustness, Zhu et al. [57] propose adjusting the geometric properties of the adversarial object to fit the discrete LiDAR signals by reconstructing its surfaces. Additionally, Abdelfattah et al. [1] and Cao et al. [4] extend such attacks to multi-sensor fusion systems, generating adversarial objects with specific shapes that can mislead the perception system based on both camera and LiDAR.

Attacks using common objects [30, 56, 58, 60]. These attacks do not require specific shapes for the objects used. Zhu et al. [60] discover that there are some critical adversarial locations in the physical space. By placing some common objects with reflective surfaces (e.g., drones and cardboard) around these locations, an attacker can effectively deceive the LiDAR object detection model. An example of such an attack is shown in Figure 1b, where the attacker uses two commercial drones as the objects and intends to hide the black car from the LiDAR perception system of an autonomous vehicle. The attacker first identifies two critical adversarial locations in the environment, then achieves the attack goal by controlling the drones to hover around these locations. The authors also demonstrate the effectiveness of such attacks against LiDAR semantic segmentation in autonomous driving, using some simple objects (e.g., cardboard and road signs) as adversarial objects [58]. Additionally, Zhang et al. [56] propose a backdoor attack strategy against LiDAR object detection models using common objects such as cargo carrier bags or cardboard boxes. In this attack, the attacker first poisons a small number of the detection model's training samples and then places one of these objects within the driving environment as the backdoor trigger during the testing stage. This method is also used to attack trajectory prediction in autonomous driving [30].

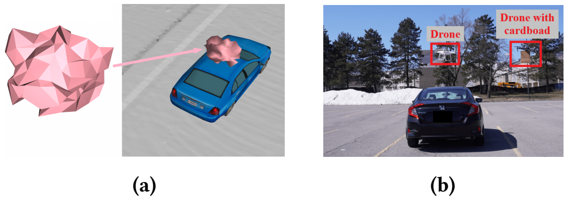


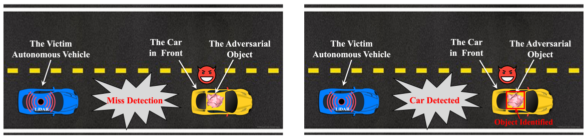
Figure 1: Two examples of object-based LiDAR attacks. (a) The attack using an object with a specific shape [45]. (b) The attack using objects with arbitrary shapes [60].

Since the objects used in most of the above attacks can be easily obtained and deployed in real-world scenarios, these attacks are highly practical and easy to conduct. Additionally, many of these attacks can be very stealthy, as they use a small number of common objects such as cargo carrier bags and road signs, making them difficult to notice in practice. Therefore, object-based LiDAR attacks pose a significant security threat to autonomous driving.

2.3 Defenses against Object-Based LiDAR Attacks

Despite the significant security threat and potential catastrophic consequences posed by various object-based LiDAR attacks in autonomous driving, there has been no effective and practical defense against these attacks so far. Although some potential defense methods have been proposed alongside the aforementioned attacks, they are limited in both effectiveness and practicality. These potential defenses can be roughly divided into three categories: offline model training, online data analyzing, and multi-sensor fusion.

Offline model training. Some works [4, 30, 45, 56, 58, 60] propose to improve the robustness of LiDAR perception models by taking into account the attack during the offline model training process. Specifically, defenders can use data augmentation or adversarial training methods to incorporate some point clouds with adversarial LiDAR points, generated by specific attacks, into the



(a) Vehicle hiding attack

(b) Defense goal

Figure 2: The attack setting and defense goal considered in this paper.

training set of LiDAR perception models. These approaches aim to make the trained perception models more resilient to the adversarial LiDAR points generated by physical objects used by attackers during the testing stage. However, to generate the necessary adversarial LiDAR points, these approaches assume that the defender knows the attack methods being used, which is impractical in real-world scenarios. It is usually difficult for the defender to know which object-based attack method will be used by the attacker. Additionally, the aforementioned works show that these approaches cannot significantly decrease the attack success rates.

Online data analyzing. These defenses aim to mitigate the threat of attacks by analyzing and processing the collected LiDAR point clouds during the autonomous vehicle's driving process. Hau et al. [17] propose leveraging 3D shadows of objects to detect obstacles hidden from the LiDAR perception system. In LiDAR perception, 3D shadows are regions void of measurements in 3D point clouds, caused by the occlusion of objects in a scene. The authors achieve the defense goal by searching for void regions and locating the obstacles that cause these shadows in the collected point cloud. However, the defense method proposed in [17] has a very high computational cost, with an average runtime of 36.5 seconds per scene for obstacle detection. This is impractical for real-world driving scenarios, as the autonomous vehicle may collide with the obstacle before the detection result is generated. Additionally, this defense cannot identify the type of the detected object (e.g., cars or bicycles), which further reduces its practicality. In [57], the authors propose an object-based LiDAR attack using objects with specific shapes. To defend against this attack, they suggest using a smoothing algorithm on LiDAR scan results, which can disrupt the adversarial perturbations generated by the specific shape of the object, leading the attack to fail. However, this defense is designed specifically for the attack proposed in [57]. In practice, it is usually difficult to predict which attack method an attacker will use. If the attack differs and uses some common objects, this defense cannot work, and this makes the practicability of this defense questionable. Moreover, this defense has a negative impact on the precision of detecting benign objects.

Multi-sensor fusion. To mitigate the threat of object-based LiDAR attacks, some works propose using multi-sensor fusion as a potential defense [36, 56–58]. This approach assumes that the autonomous vehicle is equipped with additional sensors, such as cameras and radar. The vehicle can then use the information collected from these sensors to make decisions when the LiDAR sensor is compromised. However, this method does not fundamentally address the vulnerability of LiDAR perception to object-based attacks. Moreover, existing research has demonstrated that camera and radar perception systems are also susceptible to malicious attacks [23, 33, 50, 59]. If an attacker compromises both the camera and radar systems while attacking LiDAR, they can still achieve their goals. Furthermore, incorporating additional sensors increases the overall cost of autonomous vehicles.

3 Threat Model

Attack setting. In this paper, we consider a scenario where an autonomous vehicle uses a LiDAR perception system to detect objects on the road, and there is an attacker who aims to mislead the perception system by conducting the object-based LiDAR attack. Specifically, we focus on a common and widely studied attack goal in existing object-based attack research: *vehicle hiding* [17, 45, 56–58, 60]. As illustrated in Figure 2a, the attacker aims to hide a car in front of the victim autonomous vehicle's LiDAR detection system by placing physical objects within the driving environment. This can potentially cause traffic accidents such as rear-end collisions. Regarding the attacker's capabilities, we assume that the attacker has full knowledge of the victim's LiDAR object detection model, including its structure and parameters. The attacker can select a specific road segment to launch the attack and employ any existing object-based LiDAR attack method to achieve their goal. The car in front could be any random car on the road or one owned by the attacker. For instance, the attacker could intentionally park a car on a chosen road segment and execute the attack to cause the victim autonomous vehicle to collide with it. Additionally, following the settings used by existing object-based attacks, we assume that the attacker uses the minimum number of objects necessary to achieve their goal, minimizing their effort and enhancing the stealthiness of the attack.

Defense goal. The defense goal is illustrated in Figure 2b. As the defender, our objective in this paper is to develop an effective and practical defense mechanism that can be integrated into existing autonomous driving systems and enable LiDAR object detection models to produce accurate predictions (i.e., detect the car in front) even in the presence of the above attack. To ensure practicality, we consider a realistic scenario where we have no prior knowledge about whether the collected LiDAR point clouds are clean or contain adversarial points. We aim to employ an online defense mechanism to process the point cloud data, rather than relying on an offline approach to modify the detection model, which typically cannot adapt well to different attack methods or handle unseen attack scenarios.

4 Defense Challenges

To achieve the above defense goal, we need to address several challenges.

First, many object-based LiDAR attack methods have been developed, allowing the attacker to choose from various approaches to achieve their goal. Different attack methods can involve varying numbers of adversarial objects, with differing locations, sizes, and shapes. For instance, the attacker can use either objects with specific shapes or common objects to launch the attack. They can also employ strategies such as a backdoor attack, where the training set of the perception model is poisoned and the backdoor is activated during the testing stage, or focus solely on misleading the clean perception model during the testing stage. However, in all cases, the defender lacks any prior knowledge of the specific attack method being used by the attacker. Designing an attack-agnostic defense mechanism that is effective against different attack methods and helps the perception system detect hidden cars, regardless of the deployment of various adversarial objects, is a challenging task.

Second, unlike 2D images, the 3D LiDAR point cloud in autonomous driving is highly complex. Each point cloud collected by the LiDAR sensor contains points generated by all items surrounding the autonomous vehicle in the driving environment. The adversarial object used by the attacker can be placed anywhere within this large 3D space. Additionally, if the attacker uses common objects (e.g., cargo carrier bags or billboards), the attack can be very stealthy, making these objects difficult to notice. Thus, precisely locating the adversarial object within the large 3D space is also a challenging task. This difficulty is further compounded if the attacker uses multiple adversarial objects and places them at different locations.

Third, in real-world driving scenarios, it is impossible to predict when and where an attack will occur. The defense mechanism must continuously monitor road conditions and provide accurate feedback on any suspicious activity. If suspicious activity is detected, the mechanism should not guide the autonomous vehicle to take aggressive actions (e.g., emergency braking) until confirming the presence of a real attack. This precaution is necessary because if the vehicle reacts aggressively to each suspicious activity, its normal driving behavior would be significantly disrupted due to potential false alarms in practice. Additionally, many LiDAR sensors used by autonomous vehicles have a limited effective detection range. For example, our experiments show that the widely adopted Velodyne VLP-32 LiDAR can reliably detect vehicles on the road up to approximately 70-80 meters in typical conditions. This limitation implies that the defense's effectiveness can only be verified within this short range. Therefore, the defense mechanism must operate swiftly, as the autonomous vehicle will continue driving before the attack is confirmed (i.e., the hidden car is detected). Thus, the defense mechanism must be efficient enough to prevent potential traffic accidents as intended by the attacker.

Lastly, different autonomous driving systems may use various LiDAR object detection models. To ensure the defense mechanism is effective across different systems, it should be compatible with multiple LiDAR object detection models and easy to integrate into these systems. Therefore, the designed defense mechanism should function as an independent component within the autonomous driving system.

5 Methodology

In this section, we first provide an overview of the proposed defense mechanism and then describe the details of each stage in the mechanism.

5.1 Overview

To address the aforementioned challenges, we propose a novel online defense mechanism designed to process the collected LiDAR point cloud and remove points generated by adversarial objects, thus enabling the perception model to detect the hidden car in front. As shown in Figure 3, this mechanism can be easily integrated into the pipeline of existing autonomous driving systems, which typically include modules such as sensing, perception, trajectory planning, and vehicle control. The proposed mechanism is positioned between the LiDAR sensing and perception modules.

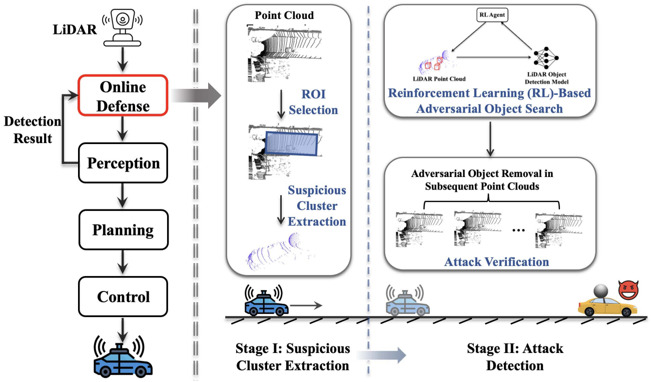

LiDAR
Online Defense
Detection Result
Perception
Planning
Control
Point Cloud
ROI Selection
Suspicious Cluster Extraction
RL Agent
LiDAR Point Cloud
LiDAR Object Detection Model
Reinforcement Learning (RL)-Based Adversarial Object Search
Adversarial Object Removal in Subsequent Point Clouds
...
Attack Verification
Stage I: Suspicious Cluster Extraction
Stage II: Attack Detection

Figure 3: An overview of the proposed defense mechanism. The blue car represents the victim autonomous vehicle, while the yellow car in front is the one selected by the attacker to launch the attack. The ball on the rooftop of the yellow car serves as an example of adversarial objects.

This mechanism comprises two stages: *suspicious point cluster extraction* and *attack detection*. The underlying philosophy of this defense is to emulate the behavior of a human driver when encountering a potential obstacle. In a scenario where a human driver is navigating a road and notices a potential obstacle far ahead, the driver typically responds by doubting whether it is a real obstacle, reducing vehicle speed to confirm, and taking appropriate actions upon confirmation. The behavior of our defense mechanism in the proposed stages is inspired by these human driver reactions.

Stage I: Suspicious point cluster extraction. In this stage, the defense mechanism continuously monitors the collected LiDAR point clouds as the autonomous vehicle drives on the road. It first selects a region of interest (ROI) (e.g., the road segment of a specific length ahead of the vehicle) in the point cloud and then checks for the presence of a suspicious point cluster in this region. Such a point cluster can be extracted using a clustering algorithm. If a suspicious point cluster is detected ahead but the LiDAR detection model does not report any detected object, the proposed defense mechanism immediately activates the second stage (i.e., Stage II in Figure 3). Meanwhile, the mechanism saves the point cloud containing the suspicious cluster, along with a specific number of subsequent point clouds, for use in the second stage.

Stage II: Attack detection. The second stage aims to detect and confirm the attack, preventing the autonomous vehicle from colliding with the car in front. Once Stage II is activated, the defense mechanism immediately guides the vehicle to reduce its speed and begins locating the adversarial LiDAR points in the suspicious cluster. The goal is to remove these adversarial LiDAR points and enable the detection model to identify the hidden car in the specific point cloud containing the suspicious cluster saved from Stage I. To achieve this, we propose a reinforcement learning-based search method to locate adversarial LiDAR points, which is more effective and efficient than intuitive search methods such as random search. The search operation stops when a convergence criterion is satisfied when the attack is detected. If no object is detected after a specific time period, the search operation also stops, and the defense mechanism returns to Stage I.

After completing the search process, if the hidden car is detected in the above point cloud, the mechanism further verifies the attack by removing the identified adversarial LiDAR points in subsequent point clouds and observing whether the car can be detected in these point clouds. If the car is detected in a predefined number of point clouds, the autonomous vehicle will take actions such as stopping or changing lanes.

5.2 Stage I: Suspicious Point Cluster Extraction

As described in Section 3, the attack goal considered in this paper is to hide a front car (the yellow car in Figure 3) from the LiDAR object detection model of the victim vehicle (the blue car in Figure 3). Although the front car and the adversarial objects cannot be "seen" by the detection model when the attack is conducted, their generated LiDAR points do not disappear, resulting in a point cluster in the point cloud. Therefore, Stage I is designed to continuously monitor the collected LiDAR point clouds and timely identify the suspicious point cluster in front of the victim vehicle on the road.

To efficiently identify the suspicious point cluster in a specific point cloud, the defense mechanism first segments the ground from other objects in the point cloud using the RANSAC (Random Sample Consensus) algorithm [11] and removes the points generated by the road surface. Then, it selects a region of interest (ROI) that covers the area where the suspicious point cluster is located. The ROI in our design is a rectangle directly ahead of the autonomous vehicle. Its length and width are usually set based on the LiDAR sensor's reliable detection range and the lane width. This operation addresses the second challenge described in Section 4 and enables the defense to focus on a small set of points that are above the ground and within the ROI, rather than on the entire complex point cloud. Although adversarial objects can be placed anywhere within the driving environment, existing object-based LiDAR attacks typically place these objects around the front car that the attacker intends to hide. This is because if the objects are far from the front car, they will not affect the car's geometric features, and the attack goal cannot be achieved. Therefore, it is easy to select an ROI that covers the area where both the front car and adversarial objects are located.

Finally, the defense mechanism uses the DBSCAN algorithm to cluster all the points within the ROI and extract the suspicious point cluster. DBSCAN is chosen for its robustness in identifying clusters of varying shapes and sizes, as well as its ability to handle noise effectively. In some cases, the clustering algorithm may output several clusters for the front car and adversarial objects if the attacker uses multiple objects that are not very close to the car. To address this issue, we use an additional step to merge clusters that are within a specific range of the larger cluster generated by the potential car.

The proposed defense mechanism periodically operates the above processes on the collected point clouds. Once a suspicious point cluster is extracted from a specific point cloud, denoted as P_i (where i is the index), but the LiDAR detection model does not report any detected object from this point cloud, the mechanism immediately activates Stage II. Meanwhile, the mechanism saves point cloud P_i and the subsequent n point clouds P_{i+1},P_{i+1},$\cdots$, P_{i+n} for use in the second stage.

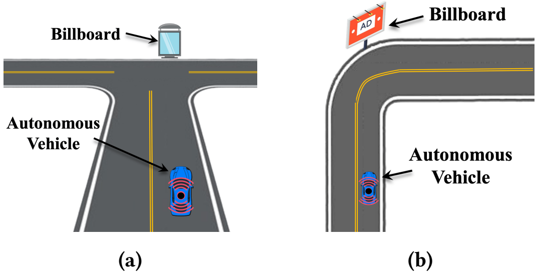


Figure 4: Two scenarios where the suspicious point cluster could be generated by a billboard.

5.3 Stage II: Attack Detection

Although a suspicious point cluster is identified in Stage I, we cannot determine whether this point cluster is generated by an attack or by a normal object in front of the autonomous vehicle. For instance, this point cluster may be generated by an object at the roadside. Figure 4 shows two possible scenarios where the suspicious point cluster could be generated by a billboard. Therefore, it is not advisable for the autonomous vehicle to take actions such as stopping or changing lanes solely based on the appearance of a suspicious point cluster in the collected point cloud. It is necessary to further detect and confirm whether the point cluster is caused by an attack.

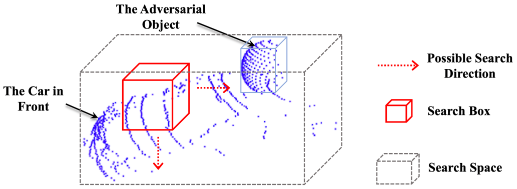


Figure 5: An example of adversarial object search.

Stage II is designed for the above purpose and to address the third challenge described in Section 4. In this stage, we aim to identify the LiDAR points generated by adversarial objects (if there is an attack) in the collected point cloud and mitigate their impact on the perception results. By iteratively searching for potential adversarial points, removing them from the point cloud, and observing the output of the LiDAR detection model on the modified point cloud, we can determine whether there is an object-based attack. Figure 5 shows an example that illustrates our goal in Stage II. In this example, we assume that the attacker uses a single adversarial object (a ball) to launch the attack and places this object on the rooftop of the car they intend to hide. Such an attack has been demonstrated effective in the physical world [56]. The blue points in Figure 5 represent the suspicious point cluster extracted in Stage I. Our objective is to use a search box (the red cuboid in Figure 5) to locate the points generated by the adversarial object and then remove them from the original point cloud containing the suspicious cluster (i.e., P_i). The search space is defined by an external cuboid surrounding the suspicious point cluster (the grey cuboid). We then use the modified point cloud to query the LiDAR detection model and determine whether an attack is present by observing if the detection model can output a bounding box for the car.

An intuitive approach to achieve the above goal is to use random search. Using Figure 5 as an example, with random search, we first determine an appropriate size of the search box and randomly select a position for it within the search space. We then remove all the points within the search box from the original point cloud and feed the modified point cloud into the LiDAR detection model. If the detection model outputs a bounding box for a car, an attack is detected. Otherwise, we repeat the process. Although this method is intuitive and simple, it may not be practical in reality. First, it is usually difficult to determine an appropriate size for the search box without knowledge of the specific attack being used. If the search box is too small, it cannot cover sufficient LiDAR points generated by the adversarial object, and if it is too large, it may include points generated by the car. In both cases, the detection of the attack may fail, making random search very challenging. Second, as discussed in Section 4, the defense mechanism must be efficient, as the autonomous vehicle will continue driving before the attack is detected. If the detection process takes too long, we may not be able to prevent potential traffic accidents as intended by the attacker. However, with the random search method, we cannot guarantee that the attack will be detected in a short period of time.

To address the above issues, we propose a reinforcement learning (RL)-based method to search for potential adversarial objects in the point cloud and determine whether there is an attack by interacting with the LiDAR object detection model. The proposed method is attack-agnostic and does not require any prior knowledge about the adversarial objects used by the attacker. It is intelligent and can automatically adjust the positions and sizes of the search boxes to accommodate the adversarial objects. This method addresses the first challenge as well as the efficiency challenge described in Section 4.

Given that the attacker may use multiple adversarial objects in the attack [60], the proposed method utilizes multiple search boxes to locate these objects simultaneously. We aim to determine the optimal positions and sizes of these search boxes to cover sufficient LiDAR points generated by adversarial objects in the point cloud. We formulate the search for the positions and sizes of these boxes as a reinforcement learning problem and define the attack detection in Stage II as a decision-making process. Reinforcement learning is a highly effective technique for decision-making and has been widely used in various decision-making tasks [34, 42, 52], which motivates the development of our method.

In decision-making, there is an agent that interacts with its environment by taking actions and observing the reward. In our context, the environment is the LiDAR detection model and the point cloud containing the suspicious point cluster (i.e., point cloud P_i saved in Stage I). The action is placing a set of search boxes $B = \{\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_N\}$, where $\mathbf{b}_j$ is the j-th search box and N is the number of these boxes, in the point cloud. The reward is defined based on the detection results produced by the LiDAR detection model. This approach formulates the determination of the search boxes (regarding their positions and sizes) as an optimization problem, which is much more effective and efficient compared to random search strategies.

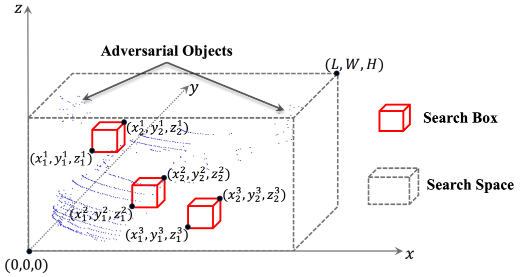


Figure 6: An example of adversarial object search with multiple search boxes.

Without loss of generality, we use three cuboids as the search boxes (i.e., $N = 3$) in the description of our proposed method. As shown in Figure 6, each search box $\mathbf{b}_j$ ($j \in \{1, 2, 3\}$) is parameterized by the coordinates of its opposite corner points (x_1^j, y_1^j, z_1^j) and (x_2^j, y_2^j, z_2^j). The range of these parameters are limited within an external cuboid (with dimensions $L \times W \times H$) surrounding the suspicious point cluster. The set of all possible values for the three search boxes' parameters is represented as:

$$\begin{aligned}\mathcal{S} = &\{(x_1^1, y_1^1, z_1^1, x_2^1, y_2^1, z_2^1, x_1^2, y_1^2, z_1^2, ..., x_2^3, y_2^3, z_2^3) \\ &|x_q^j \in [0, L], y_q^j \in [0, W], z_q^j \in [0, H]\},\end{aligned} \tag{1}$$

where $q \in \{1, 2\}$. Each element in set $\mathcal{S}$ has 18 dimensions, and we configure the agent to take 18 actions in sequence to select the values for these dimensions, generating an action sequence $\mathbf{a} = (a_1, a_2, ..., a_{18}) \in \mathcal{S}$, where $a_t (t \in \{1, 2, ..., 18\})$ is the action (or the selected value) for the t-th dimension. We also discretize the ranges $[0, L]$, $[0, W]$, and $[0, H]$ to obtain a discrete action space, framing the value selection of the above parameters as a classification problem.

In our design, the policy network p_π used by the agent is a combination of an embedding layer, a Long Short-Term Memory (LSTM) layer, and a fully connected layer. We use θ to denote its parameters. The embedding layer maps discrete actions into a continuous space, which is crucial for capturing complex relationships between different actions in $\mathbf{a}$. The LSTM layer maintains dependencies and captures long-term relationships between actions, helping to generate coherent and contextually relevant search boxes. The fully connected layer following the LSTM maps its outputs to the action space. At the t-th step, the policy network produces a probability distribution for the potential actions at this step. It then samples an action a_t and records the associated probability p_t. This process finally generates the action sequence $\mathbf{a}$ and a probability sequence $\mathbf{p} = \{p_1, p_2, ..., p_{18}\}$ that records the probabilities of sampling these actions. Based on $\mathbf{a}$, we can derive the positions and sizes of the three search boxes in the point cloud. By removing all LiDAR points within these boxes and feeding the modified point cloud into the

LiDAR detection model, we obtain the detection result on the modified point cloud. Using this detection result, we can further calculate a reward R. The loss function used to update the policy network is defined as $\mathcal{L} = -R \sum_{t=1}^{6N} \ln p_t$.

Once we derive a set of search boxes using the above approach, we update the parameters of the policy network using $\mathcal{L}$. The updated network is then employed to derive another set of search boxes using the same approach. This process is repeated iteratively until the convergence criterion is met or a specific number of iterations is reached. If no car is detected after the specified number of iterations, the attack is not detected in the point cloud P_i, and our defense returns to Stage I. However, if the process converges and the detection model finally outputs a bounding box of a car, the attack is detected in the point cloud P_i. To further confirm the attack, we apply the derived search boxes to the subsequent point clouds $P_{i+1}, P_{i+2}, \ldots, P_{i+n}$ and observe the outputs of the detection model on these modified point clouds. If the number of point clouds in which the attack is also detected exceeds a certain threshold, the attack is confirmed, and the autonomous vehicle immediately takes actions such as stopping or changing lanes.

The reward R, which represent the optimization goal, plays an important role in our proposed method. It guides the search for the optimal positions and sizes of the search boxes. In our design, the reward R comprises the following components.

- **Confidence score output by the LiDAR detection model**. A typical LiDAR object detection model learns geometric features from the input point cloud and outputs a set of bounding box proposals. Each proposal is accompanied by a confidence score indicating the likelihood that the bounding box contains a vehicle. Proposals with confidence scores below a certain threshold are ignored, while those with high confidence scores are accepted as detection results. Since our goal in Stage II is to detect the hidden front car, a higher confidence score for the output bounding box indicates better performance of the designed defense method. Therefore, the first component of reward R is the confidence score output by the detection model, which is denoted as C.
- **Overlap between search boxes and important regions within the search space**. To identify adversarial objects within the search space, the optimization process should guide the search boxes to continuously approach the regions where the adversarial objects are located. Therefore, compared to other regions in the space, the regions containing the adversarial objects are more critical in the optimization process. However, we do not have prior knowledge about these important regions before the search process begins. If we can identify them during the optimization process and use this information to guide the movement of the search boxes, the search would be more effective and efficient. To achieve this goal, we first divide the search space into many grids and assign an importance score to each grid containing LiDAR points. Grids without any LiDAR points are skipped. The initial value of the importance score is set to 0. During each iteration of the optimization, if the confidence score output by the detection model exceeds a threshold, the importance scores of the grids within the search boxes are incremented by 1. Next, we extract the grids whose importance scores are above a certain threshold and calculate the 3D Intersection over Union (3D IoU) between the search boxes and these grids. The 3D IoU (denoted as I) is used as the second component of the reward. Through the above operations, we aim to enable the proposed method to leverage the experience from previous search iterations to guide the search process in subsequent interactions.
- **The space occupied by search boxes**. To identify adversarial objects, the size of the search boxes should not be too large, as large boxes may include LiDAR points that are not generated by adversarial objects. Therefore, we use this component to limit the total space occupied by all search boxes in the optimization process. Specifically, we calculate the union volume V of all search boxes and use it as the third component of the reward.

The final reward R is calculated as $R = C + \alpha I - \beta V$, where α and β are the parameters used to adjust the balance between the three components.

6 Experiments in the Digital World

6.1 Experimental Setting

Dataset. We first evaluate our defense mechanism using the KITTI dataset [13], which is one of the most widely used public datasets in the field of autonomous driving. Since we focus on object-based attacks aimed at hiding a target front vehicle, we select LiDAR frames that contain a vehicle in front of the victim vehicle (i.e., the autonomous vehicle) in the same lane.

Object detection models. We consider two widely used state-of-the-art LiDAR object detection models: PIXOR [51] and PointPillars [25]. PIXOR achieves real-time detection using an efficient bird's-eye view (BEV) representation of a 3D driving scene. PointPillars is a voxel-based model that divides the point cloud into vertical columns (pillars) and utilizes PointNets to learn their features. The threshold for the confidence score is set to 0.5 in these models.

Attack methods. We consider various types of state-of-the-art object-based LiDAR attacks. For attacks using objects with specific shapes, we include the methods proposed in [45] and [57]. The method developed by Tu et al. [45] aims to hide a target vehicle from the LiDAR object detection model by placing a 3D-printed object with a specific shape on the rooftop of the target vehicle. To improve the physical robustness of this adversarial object, Zhu et al. [57] propose adjusting the geometric properties of the object to enhance the attack's effectiveness in the physical world. The attack methods proposed in [45] and [57] are denoted as **AdvObj** and **AE-Morpher**, respectively. For attacks using common objects, we consider those proposed in [60] and [56]. Zhu et al. [60] identify certain adversarial locations in the physical space where placing common objects can achieve the attacker's goal. This method is denoted as **AdvLoc**. Differing from the above attacks, which focus on modifying the driving environment during the testing stage of LiDAR object detection models, Zhang et al. [56] propose a backdoor attack that involves both poisoning the training set of the detection model and modifying the driving environment at the testing stage. The backdoor attack proposed in [56] is denoted as **BALiDAR**.

Table 1: Performance on different attack methods.

Attack	40 − 45m		50 − 55m		60 − 65m	
	DR(%)	RT(s)	DR(%)	RT(s)	DR(%)	RT(s)
AdvLoc	85.5	3.4	91.1	3.1	95.6	2.6
BALiDAR	87.5	5.1	91.4	4.7	90.4	3.8
AdvObj	83.3	4.4	85.6	4.6	81.1	4.0
AE-Morpher	90.0	3.6	94.4	3.8	96.7	3.8

Baselines. To the best of our knowledge, there is no existing defense strategy against object-based LiDAR attacks that is both attack-agnostic and detection model-agnostic while also being capable of identifying adversarial objects and practical for real-world driving scenarios. In our experiments, we consider two baseline methods for comparison. The first baseline method is *adversarial training*, where we retrain the detection model by inserting adversarial point clusters generated by potential adversarial objects into the training data. The second baseline method is *random search*, in which we randomly select the positions of search boxes within the search space and modify the input point cloud by removing all LiDAR points within these randomly positioned boxes.

Evaluation metrics. We mainly use the following metrics to evaluate the performance of our proposed mechanism.

- *Detection rate (DR)*: This metric is defined as the percentage of attacked LiDAR frames in which the hidden vehicle is successfully detected by the defense, relative to the total number of attacked LiDAR frames. A higher DR indicates a more effective defense method.
- *Runtime (RT)*: This metric is used to evaluate the efficiency of the defense. It is defined as the time required to successfully detect the hidden vehicle in a single attacked LiDAR frame. A shorter runtime indicates a more efficient defense.

Other settings. In this paper, the default number of search boxes used for identifying adversarial objects is set to 3. For the attack verification process in Stage II, we remove adversarial LiDAR points in three subsequent point clouds. If the attack can be detected in at least two of them, the attack is confirmed. All experiments in this paper are conducted on a computer equipped with an Intel i9-10920X processor and an Nvidia RTX 6000 GPU. In implementing our proposed reinforcement learning framework, the batch size is set to 5. Specifically, we create 5 copies of the suspicious point cluster identified in Stage I and place them in the same batch.

6.2 Detection Performance

Performance on different attack methods. The reinforcement learning-based attack detection designed in Stage II is the most crucial component of our defense mechanism. We first evaluate the effectiveness and efficiency of this component against various attack methods. In this experiment, we use the PIXOR object detection model. We implement the four attack methods mentioned above according to the settings in their original papers. For AdvLoc, we use two billboards hovered by drones as the adversarial objects and place them at two adversarial locations around the target vehicle (i.e., the vehicle the attacker intends to hide). In the BALiDAR attack, we use a sphere with a 0.4m radius as the backdoor trigger. For AdvObj and AE-Morpher, we follow their original settings, generating an adversarial object with a specific shape for each of them and placing the object on the rooftop of the target vehicle.

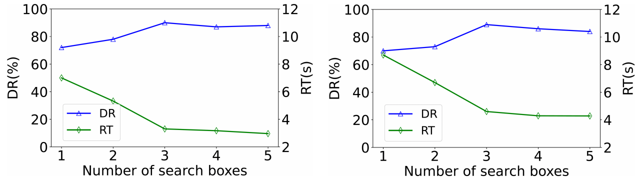
Table 2: Comparison with baseline methods.

Defense	50 − 55m		60 − 65m	
	DR(%)	RT(s)	DR(%)	RT(s)
Adversarial training	9.5	/	10.2	/
Random research	28.9	8.8	35.6	8.0
Ours	**91.3**	**3.9**	**93.0**	**3.2**

Table 1 shows the detection rate (DR) and the average runtime (RT) of the proposed reinforcement learning-based detection method for the four attacks mentioned above. In this experiment, we consider three distance ranges (40 − 45 meters, 50 − 55 meters, and 60 − 65 meters) between the target vehicle and the autonomous vehicle. For each range, we randomly select approximately 30 successfully attacked LiDAR frames for each type of attack, most of which are collected in different environments. The results in Table 1 demonstrate that our proposed detection method performs well across all cases, with the detection rate exceeding 85% in most instances. For example, the detection rates of our method against the AdvLoc attack for the three distance ranges are 85.5%, 91.1%, and 95.6%, respectively.

The runtime results in Table 1 indicate that the reinforcement learning-based detection method is efficient. For instance, when the distance range is 60 − 65 meters, the average runtime across the four attacks is only 3.6 seconds. Given that the autonomous vehicle reduces its speed before detection starts, this runtime allows sufficient time for the vehicle to take appropriate actions (e.g., stopping or changing lanes) once the attack is detected. For example, if the autonomous vehicle is traveling at 15 miles per hour during detection, the distance between the autonomous vehicle and the target vehicle would still be around 35 meters, providing ample space for the vehicle to stop or change lanes.

Comparison with baseline methods. We compare our proposed defense with two baseline methods: adversarial training and random search, which are described in Section 6.1. In this experiment, we still use the PIXOR detection model. We consider the AdvLoc and BALiDAR attacks. For adversarial training, we use each of the two attack methods to generate 100 point clouds with adversarial LiDAR points and add them to the training set of the detection model. For random search, we also use three search boxes and randomly select the positions and sizes of these boxes each time and observe the output of the detection model on the point cloud after removing points within these randomly selected boxes. Since our method stops the search once the reinforcement learning framework converges, to ensure a fair comparison, we define a successful detection for random search on a single LiDAR frame as detecting the hidden vehicle successfully three times. The runtime is defined as the time required for a successful detection. We also set a time threshold of 10 seconds to stop the search process manually, as the search process may otherwise never conclude in some cases. In the testing stage, we consider two distance ranges (50-55 meters



(a) AdvLoc **(b) BALiDAR**

Figure 7: Impact of the number of search boxes on the detection performance.

and 60-65 meters) between the autonomous vehicle and the target vehicle. Similar to the previous experiment, we randomly select approximately 30 LiDAR frames for each distance range. Table 2 reports the average results for two attack methods (i.e., AdvLoc and BALiDAR). Since adversarial training is conducted offline, we do not report its runtime. Our proposed defense significantly outperforms these baseline methods in terms of both detection rate and runtime. The runtime of our method is less than half that of random search, which is crucial in autonomous driving, as the vehicle usually has limited time and distance to take actions.

Impact of the number of search boxes. In our experiments, the default number of search boxes used for identifying adversarial objects is set to 3. To evaluate the impact of the number of search boxes on defense performance, we vary the number from 1 to 5 and calculate the average detection rate (DR) and runtime (RT) on a single LiDAR point cloud. We use PIXOR as the detection model and randomly select approximately 30 attacked LiDAR frames within the distance range of 40 to 65 meters for evaluation. The AdvLoc and BALiDAR attacks are considered in this experiment. The experimental results for the two attacks are shown in Figure 7a and Figure 7b, respectively. We observe that the average runtime decreases as the number of search boxes increases. This is reasonable because the more search boxes used, the higher the likelihood of including adversarial points within them. However, for the average detection rate, its value first increases and then decreases as the number of search boxes increases. A possible reason is that, while increasing the number of search boxes can improve the chances of capturing more adversarial points, it also increases the likelihood of including benign points (i.e., points generated by the target vehicle).

Performance on different object detection models. Our proposed defense mechanism is detection model-agnostic. It treats the detection model as a black box and only requires querying the model to obtain its output. Therefore, our mechanism can be applied to any LiDAR object detection model. To demonstrate the effectiveness of our defense across different detection models, we also use another widely adopted LiDAR object detection model, PointPillars, to evaluate the detection performance. Table 3 presents the average detection rate and runtime for the AdvLoc and BALiDAR attacks when PointPillars is used as the detection model. We randomly select approximately 50 attacked LiDAR frames within the distance range of 40 to 65 meters between the autonomous vehicle and the target vehicle for evaluation. We can observe that the proposed defense still achieves high detection rates within just a few seconds, and the results are similar to those obtained when the detection model is PIXOR.

Table 3: Defense performance on PointPillars.

Attack method	DR(%)	RT (*s*)
AdvLoc	93.3%	3.0
BALiDAR	90.0%	4.8

False positive detection. As described in the first paragraph of Section 5.3, benign objects at the roadside, such as those shown in Figure 4, may also generate suspicious clusters and activate Stage II of our defense. To evaluate whether our detection method might produce false positives in these scenarios, we conduct a case study using the CARLA simulator [12], a widely used open-source autonomous driving simulator. Specifically, we simulate the driving environment shown in Figure 4a and place a billboard at the roadside. The initial distance between the billboard and the autonomous vehicle is set to 60 meters, and the vehicle's speed to 20 miles per hour. The points generated by the billboard are successfully identified as a suspicious point cluster in Stage I of our defense, and Stage II is activated. We use PIXOR as the detection model and randomly select 30 LiDAR frames within the 40-60 meter distance range between the autonomous vehicle and the billboard, finding that no vehicles are detected in any of these frames. To further assess the potential for false positive detection, we reduce the detection threshold of PIXOR from 0.5 to 0.3, and still find no vehicles detected in these frames. The results above demonstrate that there are no false positive detections in the collected LiDAR frames, indicating that our defense maintains stable performance even in benign environments.

6.3 Overall Performance

In Section 6.2, we primarily evaluate the performance of the reinforcement learning-based detection component in Stage II of our defense. Next, we assess the performance of the overall pipeline of our proposed mechanism. Specifically, we integrate our mechanism into the autonomous driving system in the CARLA simulator to observe its impact on the behavior of the autonomous vehicle.

By being integrated into the autonomous driving system, our defense mechanism detects potential attacks in the driving environment and communicates its decisions to other system modules, which can then guide changes in the autonomous vehicle's behavior. To evaluate the impact of our defense on the vehicle's final behavior, we use the CARLA simulator to run the entire pipeline of the autonomous driving system with our mechanism integrated. The CARLA simulator supports customizing driving scenarios and provides APIs for integrating our mechanism with other modules in the driving system. We use AdvLoc as the attack method and PIXOR as the detection model. We implement the AdvLoc attack with two cardboard pieces hovered by drones (approximated using rectangular shape boxes). The speed of the autonomous vehicle is set to 20 miles per hour, and the initial distance between the autonomous vehicle and the front car is set to 60 meters.

As shown in Figure 8a, when there is no defense, the effect of the attack persist over a long distance, and PIXOR fails to detect the front car until the autonomous vehicle is very close (around 11.2 meters). The front car is detected at this closer range because it

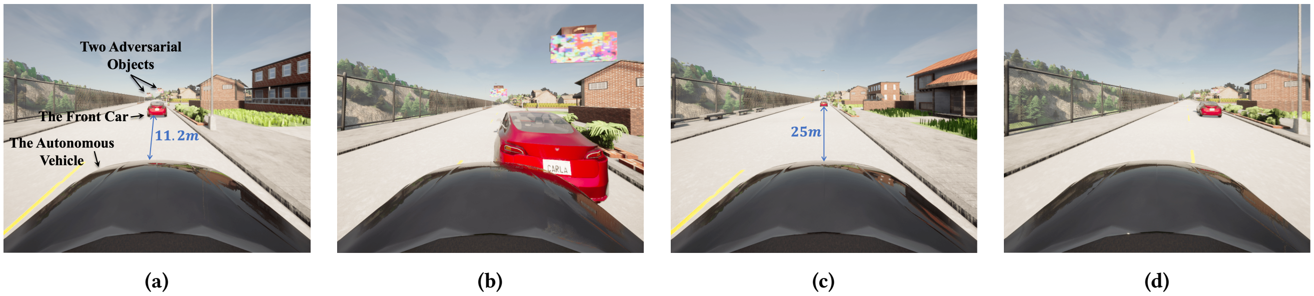


Figure 8: Impact of our defense on autonomous vehicle behavior. (a) The first LiDAR frame in which the car in front is detected when there is no defense. (b) The autonomous vehicle collides with the front car when there is no defense. (c) The first LiDAR frame after the attack is detected and verified using our defense. (d) The autonomous vehicle successfully changes lanes after the attack detection using our defense.

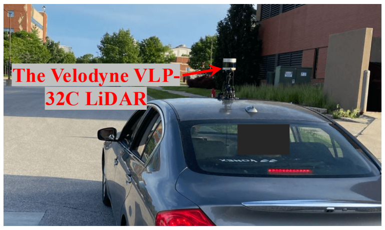


Figure 9: The LiDAR perception testbed with a Velodyne VLP-32C LiDAR.

generates more LiDAR points, which improve the detection model's feature learning. However, in this scenario, even if the autonomous vehicle takes immediate actions (e.g., changing lanes), it still collides with the front car (as shown in Figure 8b), following CARLA's default planning algorithm with the updated waypoints. In comparison, our defense mechanism enables early detection, providing sufficient distance and time for the autonomous vehicle to take appropriate actions. Figure 8c shows the first frame after the attack is detected and verified using our defense, with the distance between the two vehicles being approximately 25 meters. This distance is sufficient for the autonomous vehicle to take proper actions. As shown in Figure 8d, the autonomous vehicle successfully changes lanes, maintaining a safe distance from the front car.

The above experiments demonstrate that our proposed defense is both effective and practical when integrated into existing autonomous driving systems and applied to real driving scenarios.

7 Experiments in the Physical World

7.1 Experimental Setting

In this section, we evaluate the performance of our defense in real-world scenarios using a LiDAR perception testbed equipped with a Velodyne VLP-32C LiDAR, as shown in Figure 9. The Velodyne VLP-32C LiDAR has been widely used in autonomous driving. It features 32 channels and a 40° vertical field of view. The LiDAR sensor is mounted on the rooftop of the vehicle, approximately 1.8 meters above the ground. We use PIXOR as the object detection model and consider the AdvLoc and BALiDAR attacks, which have been implemented in the physical world [56, 60]. To launch these attacks, we use a Ford sedan as the target vehicle and park it in front of the testbed on the road. For the AdvLoc attack, we utilize two types of adversarial objects: foil-wrapped paper balls and pieces of cardboard. For the BALiDAR attack, we follow the original paper's settings, employing an exercise ball with a radius of 0.4 meters and a cargo carrier bag as adversarial objects.

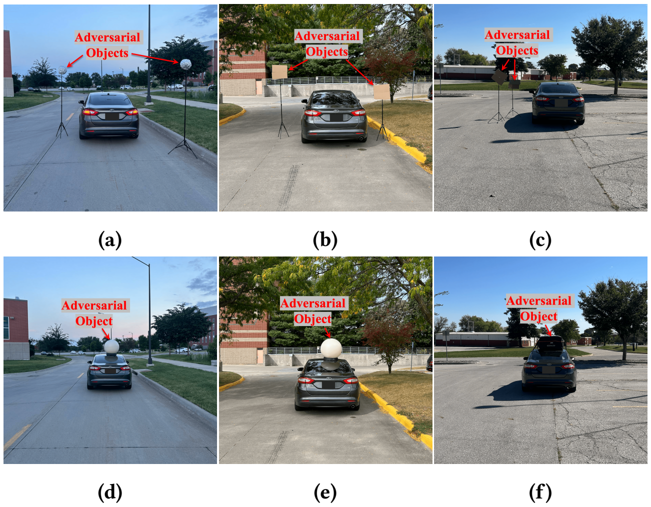


Figure 10: Real-world scenarios for AdvLoc and BALiDAR attacks. The first row illustrates the AdvLoc attack, while the second row depicts the BALiDAR attack. (a) and (d): Scenario 1; (b) and (e): Scenario 2; (c) and (f): Scenario 3.

7.2 Performance Evaluation

Performance in different driving scenarios. We first evaluate the performance of our proposed reinforcement learning-based detection method across different driving scenarios. Specifically, we consider three scenarios, as shown in Figure 10. The first row of the figure illustrates the AdvLoc attack, while the second row depicts the BALiDAR attack. In these scenarios, the Ford sedan (the target vehicle) is parked on the road, and the victim vehicle,

Table 4: Defense performance in different driving scenarios.

Attack	Scenario 1		Scenario 2		Scenario 3	
	DR(%)	RT(s)	DR(%)	RT(s)	DR(%)	RT(s)
AdvLoc	81.3	3.2	86.7	3.4	84.6	3.3
BALiDAR	83.3	2.5	88.9	2.9	82.2	3.7

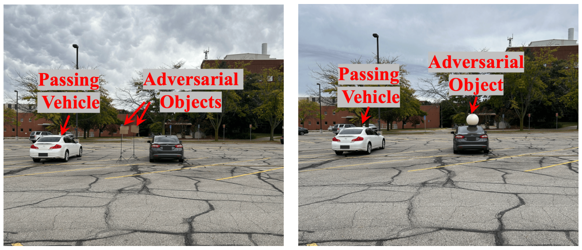
Table 5: Distance (in meters) between the victim and target vehicles after attack detection.

Attack method	5 mph	10 mph	15 mph
AdvLoc	42.8	33.1	24.6
BALiDAR	43.2	36.5	30.5

equipped with a LiDAR sensor, drives towards it. The locations of the adversarial objects are determined based on the respective attack methods. For each scenario, we randomly select 30 successfully attacked LiDAR frames within a distance range of 40 to 55 meters between the two vehicles for evaluation. The average detection rates and runtimes for AdvLoc and BALiDAR attacks in different scenarios are presented in Table 4. The results demonstrate that our proposed method achieves good detection rates across various physical-world driving scenarios involving different adversarial objects. The efficiency in the physical world is comparable to that in the digital world, with attacks being detected in just a few seconds. We also observe that detection rates in these scenarios are slightly lower than those in the digital world. A possible reason is that the number of adversarial LiDAR points collected in the physical world is typically lower than in the digital world due to factors such as hardware limitations and physical conditions, making it more challenging to identify the adversarial objects.

Next, we calculate the distances between the victim vehicle and the target vehicle after the attack is detected and verified at different speeds. Our aim is to evaluate the effectiveness of our defense in preventing potential collisions. Specifically, we consider three driving speeds for the victim vehicle in Scenario 1 (as shown in Figure 10a and Figure 10d): 5 miles per hour, 10 miles per hour, and 15 miles per hour. We randomly select attacked LiDAR frames within the distance range of 50 to 55 meters between the two vehicles. First, we calculate the average runtime required for detecting and verifying the attack. Then, we estimate the distance between the two vehicles when the victim vehicle begins to take actions such as stopping or changing lanes. Table 5 shows the results for the AdvLoc and BALiDAR attacks. We can observe that the distances in all cases are sufficient for the victim vehicle to take appropriate actions and prevent potential dangers.

Impact of the passing vehicle. In real-world driving scenarios, other vehicles may be present around the target vehicle when an attack is launched. To assess the impact of surrounding vehicles on our attack detection performance, we simulate a real-world scenario with a passing vehicle near the target vehicle that the attacker aims to hide. This scenario is depicted in Figure 11. In this experiment, we evaluate both AdvLoc and BALiDAR attacks, randomly selecting 30 attacked LiDAR frames within the distance range of 40 to 55 meters between the victim vehicle and the target vehicle. The average detection rates for the AdvLoc and BALiDAR attacks are 86.7% and 87.8%, respectively, with average runtimes of 3.5s and 3.6s. These results demonstrate that our defense maintains good effectiveness and efficiency, even in the presence of another vehicle in the driving environment.



(a) AdvLoc (b) BALiDAR

Figure 11: The scenario with a passing vehicle.

8 Limitations and Future Work

Efficiency. Although the proposed defense mechanism is highly efficient, it is not real-time and requires a few seconds to complete the attack detection process. If the victim autonomous vehicle is traveling at high speed, it would need to decelerate to allow sufficient time for attack detection and to take appropriate actions to prevent potential traffic accidents. While this is acceptable in most driving scenarios, a real-time defense mechanism against object-based LiDAR attacks would be more desirable for autonomous driving.

Other attacks. In this paper, we primarily focus on defending against vehicle hiding attacks, which have been widely studied in existing object-based LiDAR attack research. However, attackers may launch other types of object-based LiDAR attacks in practice. For example, they may target the LiDAR perception systems of autonomous vehicles to hide other objects, such as pedestrians or bicycles, by placing adversarial objects in the driving environment. Additionally, attackers could compromise trajectory prediction in autonomous driving systems through object-based LiDAR attacks. In future work, we plan to extend the proposed defense mechanism to address these additional attack types.

9 Conclusion

In this paper, we study how to effectively defend against object-based LiDAR attacks in autonomous driving. We propose a novel online defense mechanism that processes collected LiDAR data to mitigate potential threats before the data is fed into the perception module of autonomous driving systems. This mechanism is not only effective and efficient for real-world autonomous driving but also attack-agnostic and capable of identifying adversarial objects used by attackers. The performance of the proposed defense is evaluated in both simulated environments and real-world scenarios using a LiDAR perception testbed. The experimental results show that our defense can detect attacks within a few seconds with a high detection rate.

References

[1] Mazen Abdelfattah, Kaiwen Yuan, Z Jane Wang, and Rabab Ward. 2021. Towards universal physical attacks on cascaded camera-lidar 3d object detection models. In *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 3592–3596.

[2] Chengtai Cao, Xinhong Chen, Jianping Wang, Qun Song, Rui Tan, and Yung-Hui Li. 2024. CCTR: Calibrating Trajectory Prediction for Uncertainty-Aware Motion Planning in Autonomous Driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 20949–20957.

[3] Yulong Cao, S Hrushikesh Bhupathiraju, Pirouz Naghavi, Takeshi Sugawara, Z Morley Mao, and Sara Rampazzi. 2023. You Can't See Me: Physical Removal Attacks on {LiDAR-based} Autonomous Vehicles Driving Frameworks. In *32nd USENIX Security Symposium (USENIX Security 23)*. 2993–3010.

[4] Yulong Cao, Ningfei Wang, Chaowei Xiao, Dawei Yang, Jin Fang, Ruigang Yang, Qi Alfred Chen, Mingyan Liu, and Bo Li. 2021. Invisible for both camera and lidar: Security of multi-sensor fusion based perception in autonomous driving under physical-world attacks. In *2021 IEEE symposium on security and privacy (SP)*. IEEE, 176–194.

[5] Yulong Cao, Chaowei Xiao, Benjamin Cyr, Yimeng Zhou, Won Park, Sara Rampazzi, Qi Alfred Chen, Kevin Fu, and Z Morley Mao. 2019. Adversarial sensor attack on lidar-based perception in autonomous driving. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*. 2267–2281.

[6] Yulong Cao, Chaowei Xiao, Dawei Yang, Jing Fang, Ruigang Yang, Mingyan Liu, and Bo Li. 2019. Adversarial objects against lidar-based autonomous driving systems. *arXiv preprint arXiv:1907.05418* (2019).

[7] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. 2019. Fast point r-cnn. In *Proceedings of the IEEE/CVF international conference on computer vision*. 9775–9784.

[8] Jiahe Cui, Shuyao Shi, Yuze He, Jianwei Niu, Guoliang Xing, and Zhenchao Ouyang. 2024. {VILAM}: Infrastructure-assisted 3D Visual Localization and Mapping for Autonomous Driving. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*. 1831–1845.

[9] Jianning Deng, Gabriel Chan, Hantao Zhong, and Chris Xiaoxuan Lu. 2023. See beyond seeing: Robust 3d object detection from point clouds via cross-modal hallucination. *arXiv preprint arXiv:2309.17336* (2023).

[10] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. 2021. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 1201–1209.

[11] Konstantinos G Derpanis. 2010. Overview of the RANSAC Algorithm. *Image Rochester NY* 4, 1 (2010), 2–3.

[12] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An Open Urban Driving Simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*. 1–16.

[13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[14] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[15] Dongfang Guo, Yuting Wu, Yimin Dai, Pengfei Zhou, Xin Lou, and Rui Tan. 2024. Invisible Optical Adversarial Stripes on Traffic Sign against Autonomous Vehicles. In *Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications and Services*. 534–546.

[16] R Spencer Hallyburton, Yupei Liu, Yulong Cao, Z Morley Mao, and Miroslav Pajic. 2022. Security analysis of {Camera-LiDAR} fusion against {Black-Box} attacks on autonomous vehicles. In *31st USENIX Security Symposium (USENIX Security 22)*. 1903–1920.

[17] Zhongyuan Hau, Kenneth T Co, Soteris Demetriou, and Emil C Lupu. 2021. Object removal attacks on lidar-based 3d object detectors. *arXiv preprint arXiv:2102.03722* (2021).

[18] Zhongyuan Hau, Soteris Demetriou, and Emil C Lupu. 2022. Using 3D Shadows to Detect Object Hiding Attacks on Autonomous Vehicle Perception. In *2022 IEEE Security and Privacy Workshops (SPW)*. IEEE, 229–235.

[19] Chenhang He, Ruihuang Li, Shuai Li, and Lei Zhang. 2022. Voxel Set Transformer: A Set-to-Set Approach to 3D Object Detection from Point Clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8417–8427.

[20] Yuze He, Chen Bian, Jingfei Xia, Shuyao Shi, Zhenyu Yan, Qun Song, and Guoliang Xing. 2023. Vi-map: Infrastructure-assisted real-time hd mapping for autonomous driving. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*. 1–15.

[21] Yuze He, Li Ma, Jiahe Cui, Zhenyu Yan, Guoliang Xing, Sen Wang, Qintao Hu, and Chen Pan. 2022. AutoMatch: Leveraging Traffic Camera to Improve Perception and Localization of Autonomous Vehicles. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*. 16–30.

[22] Yuze He, Li Ma, Zhehao Jiang, Yi Tang, and Guoliang Xing. 2021. VI-eye: semantic-based 3D point cloud registration for infrastructure-assisted autonomous driving. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*. 573–586.

[23] Yunhan Jia Jia, Yantao Lu, Junjie Shen, Qi Alfred Chen, Hao Chen, Zhenyu Zhong, and Tao Wei Wei. 2020. Fooling detection alone is not enough: Adversarial attack against multiple object tracking. In *International Conference on Learning Representations (ICLR'20)*.

[24] Hongwu Kuang, Bei Wang, Jianping An, Ming Zhang, and Zehan Zhang. 2020. Voxel-FPN: Multi-scale voxel feature aggregation for 3D object detection from LIDAR point clouds. *Sensors* 20, 3 (2020), 704.

[25] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. 2019. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 12697–12705.

[26] Jiale Li, Hang Dai, Ling Shao, and Yong Ding. 2021. From voxel to point: Iou-guided 3d object detection for point cloud with voxel-to-point decoder. In *Proceedings of the 29th ACM International Conference on Multimedia*. 4622–4631.

[27] Zhichao Li, Feng Wang, and Naiyan Wang. 2021. Lidar r-cnn: An efficient and universal 3d object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7546–7555.

[28] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. 2018. Deep continuous fusion for multi-sensor 3d object detection. In *Proceedings of the European conference on computer vision (ECCV)*. 641–656.

[29] Yang Lou, Qun Song, Qian Xu, Rui Tan, and Jianping Wang. 2023. Uncertainty-Encoded Multi-Modal Fusion for Robust Object Detection in Autonomous Driving. In *ECAI 2023*. IOS Press, 1593–1600.

[30] Yang Lou, Yi Zhu, Qun Song, Rui Tan, Chunming Qiao, Wei-Bin Lee, and Jianping Wang. 2024. A First Physical-World Trajectory Prediction Attack via LiDAR-induced Deceptions in Autonomous Driving. In *33rd USENIX Security Symposium (USENIX Security 24)*.

[31] Gregory P Meyer, Ankit Laddha, Eric Kee, Carlos Vallespi-Gonzalez, and Carl K Wellington. 2019. Lasernet: An efficient probabilistic 3d object detector for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 12677–12686.

[32] Huy Phan, Yi Xie, Jian Liu, Yingying Chen, and Bo Yuan. 2022. Invisible and efficient backdoor attacks for compressed deep neural networks. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 96–100.

[33] Kui Ren, Qian Wang, Cong Wang, Zhan Qin, and Xiaodong Lin. 2019. The security of autonomous driving: Threats, defenses, and future directions. *Proc. IEEE* 108, 2 (2019), 357–372.

[34] Zhou Ren, Xiaoyu Wang, Ning Zhang, Xutao Lv, and Li-Jia Li. 2017. Deep reinforcement learning-based image captioning with embedding reward. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 290–298.

[35] Takami Sato, Yuki Hayakawa, Ryo Suzuki, Yohsuke Shiiki, Kentaro Yoshioka, and Qi Alfred Chen. 2022. Poster: Towards large-scale measurement study on LiDAR spoofing attacks against object detection. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. 3459–3461.

[36] Shuyao Shi, Jiahe Cui, Zhehao Jiang, Zhenyu Yan, Guoliang Xing, Jianwei Niu, and Zhenchao Ouyang. 2022. VIPS: Real-time perception fusion for infrastructure-assisted autonomous driving. In *Proceedings of the 28th annual international conference on mobile computing and networking*. 133–146.

[37] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. 2020. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10529–10538.

[38] Shuyao Shi, Neiwen Ling, Zhehao Jiang, Xuan Huang, Yuze He, Xiaoguang Zhao, Bufang Yang, Chen Bian, Jingfei Xia, Zhenyu Yan, et al. 2024. Soar: Design and Deployment of A Smart Roadside Infrastructure System for Autonomous Driving. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*. 139–154.

[39] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. 2019. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–779.

[40] Shaoshuai Shi, Zhe Wang, Xiaogang Wang, and Hongsheng Li. 2019. Part-a^ 2 net: 3d part-aware and aggregation neural network for object detection from point cloud. *arXiv preprint arXiv:1907.03670* 2, 3 (2019).

[41] Hocheol Shin, Dohyun Kim, Yujin Kwon, and Yongdae Kim. 2017. Illusion and dazzle: Adversarial optical channel exploits against lidars for automotive applications. In *Cryptographic Hardware and Embedded Systems–CHES 2017: 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*. Springer, 445–467.

[42] Michelle Shu, Chenxi Liu, Weichao Qiu, and Alan Yuille. 2020. Identifying model weakness with adversarial examiner. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 11998–12006.

[43] Jiachen Sun, Yulong Cao, Qi Alfred Chen, and Z Morley Mao. 2020. Towards robust {LiDAR-based} perception in autonomous driving: General black-box adversarial sensor attack and countermeasures. In *29th USENIX Security Symposium (USENIX Security 20)*. 877–894.

[44] Fnu Suya, Jianfeng Chi, David Evans, and Yuan Tian. 2020. Hybrid batch attacks: Finding black-box adversarial examples with limited queries. In *29th USENIX*

Security Symposium (USENIX Security 20). 1327–1344.

[45] James Tu, Mengye Ren, Sivabalan Manivasagam, Ming Liang, Bin Yang, Richard Du, Frank Cheng, and Raquel Urtasun. 2020. Physically realizable adversarial examples for lidar object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 13716–13725.

[46] Yingjie Wang, Qiuyu Mao, Hanqi Zhu, Jiajun Deng, Yu Zhang, Jianmin Ji, Houqiang Li, and Yanyong Zhang. 2023. Multi-modal 3d object detection in autonomous driving: a survey. *International Journal of Computer Vision* 131, 8 (2023), 2122–2152.

[47] Yuting Wu, Xin Lou, Pengfei Zhou, Rui Tan, Zbigniew T Kalbarczyk, and Ravishankar K Iyer. 2023. Susceptibility of Autonomous Driving Agents to Learning-Based Action-Space Attacks. In *2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. IEEE, 76–83.

[48] Wenjing Xie, Tao Hu, Neiwen Ling, Guoliang Xing, Shaoshan Liu, and Nan Guan. 2023. Timely Fusion of Surround Radar/Lidar for Object Detection in Autonomous Driving Systems. *arXiv preprint arXiv:2309.04806* (2023).

[49] Yi Xie, Zhuohang Li, Cong Shi, Jian Liu, Yingying Chen, and Bo Yuan. 2021. Enabling fast and universal audio adversarial attack using generative model. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 14129–14137.

[50] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. 2020. Adversarial t-shirt! evading person detectors in a physical world. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*. Springer, 665–681.

[51] Bin Yang, Wenjie Luo, and Raquel Urtasun. 2018. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 7652–7660.

[52] Chenglin Yang, Adam Kortylewski, Cihang Xie, Yinzhi Cao, and Alan Yuille. 2020. Patchattack: A black-box texture-based attack with reinforcement learning. In *European Conference on Computer Vision*. Springer, 681–698.

[53] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. 2019. Std: Sparse-to-dense 3d object detector for point cloud. In *Proceedings of the IEEE/CVF international conference on computer vision*. 1951–1960.

[54] Georgios Zamanakos, Lazaros Tsochatzidis, Angelos Amanatiadis, and Ioannis Pratikakis. 2021. A comprehensive survey of LIDAR-based 3D object detection methods with deep learning for autonomous driving. *Computers & Graphics* 99 (2021), 153–181.

[55] Xianglong Zhang, Huanle Zhang, Guoming Zhang, Hong Li, Dongxiao Yu, Xiuzhen Cheng, and Pengfei Hu. 2023. Model Poisoning Attack on Neural Network Without Reference Data. *IEEE Trans. Comput.* 72, 10 (2023), 2978–2989.

[56] Yan Zhang, Yi Zhu, Zihao Liu, Chenglin Miao, Foad Hajiaghajani, Lu Su, and Chunming Qiao. 2022. Towards backdoor attacks against LiDAR object detection in autonomous driving. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*. 533–547.

[57] Shenchen Zhu, Yue Zhao, Kai Chen, Bo Wang, Hualong Ma, and Cheng'an Wei. 2024. AE-Morpher: Improve Physical Robustness of Adversarial Objects against LiDAR-based Detectors via Object Reconstruction. In *33rd USENIX Security Symposium (USENIX Security 24)*.

[58] Yi Zhu, Chenglin Miao, Foad Hajiaghajani, Mengdi Huai, Lu Su, and Chunming Qiao. 2021. Adversarial attacks against lidar semantic segmentation in autonomous driving. In *Proceedings of the 19th ACM conference on embedded networked sensor systems*. 329–342.

[59] Yi Zhu, Chenglin Miao, Hongfei Xue, Zhengxiong Li, Yunnan Yu, Wenyao Xu, Lu Su, and Chunming Qiao. 2023. TileMask: A Passive-Reflection-based Attack against mmWave Radar Object Detection in Autonomous Driving. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. 1317–1331.

[60] Yi Zhu, Chenglin Miao, Tianhang Zheng, Foad Hajiaghajani, Lu Su, and Chunming Qiao. 2021. Can we use arbitrary objects to attack lidar perception in autonomous driving?. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 1945–1960.