Towards Real-Time 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

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

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

Abstract

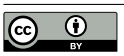
LiDAR (Light Detection and Ranging)-based object detection is a cornerstone of autonomous vehicle perception systems. Modern LiDAR perception relies heavily on deep neural networks (DNNs), which enable accurate object detection by learning geometric features from 3D point clouds. However, recent studies have shown that these systems are vulnerable to object-based adversarial attacks, where physical adversarial objects are strategically placed in the environment to manipulate LiDAR point clouds and mislead detection models. These attacks are practical, stealthy, and require no specialized hardware, posing a serious threat to the safety and reliability of AVs. Despite these risks, existing defense methods suffer from significant limitations, including high computational overhead, limited generalizability and effectiveness, and the inability to operate in real time. In this paper, we propose the first real-time defense mechanism against object-based LiDAR attacks in autonomous driving. Our solution is both detection model-agnostic and attack-agnostic, requiring no prior knowledge of the number, shape, size, or placement of adversarial objects. Positioned between the sensing and perception modules of the AV pipeline, the defense processes LiDAR point clouds in real time and employs a novel generative model that enables efficient and effective identification and removal of adversarial points from suspicious regions. Extensive experiments in both simulated and real-world environments demonstrate that our approach achieves high attack detection rates with minimal latency. This work offers a practical and robust defense solution to a growing security threat in autonomous driving.

CCS Concepts

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

Keywords

LiDAR perception, autonomous driving, adversarial attacks, defense



CCS '25, Taipei

ACM ISBN 979-8-4007-1525-9/2025/10
https://doi.org/10.1145/3719027.3765227

ACM Reference Format:
Yan Zhang, Zihao Liu, Yi Zhu, and Chenglin Miao. 2025. Towards Real-Time Defense against Object-Based LiDAR Attacks in Autonomous Driving. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security (CCS '25), October 13–17, 2025, Taipei.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3719027.3765227

1 Introduction

Autonomous driving has recently garnered substantial attention, driven by its potential to improve transportation safety, efficiency, and accessibility [6, 12, 20–22, 27, 31, 34, 40]. Central to the perception capabilities of autonomous vehicles (AVs) is LiDAR, a sensor technology that provides high-resolution, three-dimensional representations of the surrounding environment [2, 3, 5, 11, 16, 37]. LiDAR-based object detection enables AVs to recognize critical obstacles such as vehicles, pedestrians, and cyclists, making it a core component of modern autonomous driving systems.

Recent advances in LiDAR object detection have been largely driven by deep neural networks (DNNs), which have demonstrated strong performance across a range of perception tasks. However, despite their effectiveness, DNN-based detection models have been shown to be susceptible to malicious attacks. A growing body of research has uncovered various attack strategies capable of compromising LiDAR perception. Among these, *object-based LiDAR attacks* [1, 8, 10, 28, 38, 46, 48, 49, 51] have drawn particular attention due to their high practicality, stealth, and effectiveness. These attacks involve strategically placing adversarial objects, such as 3D-printed structures or common physical items, in the driving environment to introduce additional points into the LiDAR point cloud and mislead the detection model into producing incorrect results. For example, an attacker can hide a car parked on the road from the LiDAR object detection system of an AV by placing common objects (e.g., billboards) at some specific locations around the car. In contrast to laser-based spoofing attacks [7, 9, 17, 18, 23, 32, 35, 36], which mislead LiDAR perception by actively projecting laser beams into the sensor, object-based attack methods do not require specialized equipment and are more feasible for real-world deployment, making them a serious threat to the safety and reliability of autonomous driving.

To counter object-based LiDAR attacks, several categories of defense strategies have been proposed, including adversarial training, multi-sensor fusion, and runtime LiDAR data analysis. However,

existing approaches within these categories exhibit significant limitations that hinder their effectiveness and deployment in real-world autonomous driving scenarios.

Adversarial training [8, 28, 38, 46, 51] aims to improve the robustness of LiDAR detection models by incorporating attack-generated point clouds during training. While this can increase resilience to known threats, it relies on prior knowledge of specific attack types and struggles to generalize to diverse, unseen adversarial object configurations, resulting in limited effectiveness in real-world scenarios. Multi-sensor fusion [33, 46, 49] enhances perception robustness by integrating data from complementary sensors such as cameras and radar to cross-validate LiDAR inputs. Although this improves redundancy, it does not fundamentally address the vulnerabilities of the LiDAR module and remains susceptible to coordinated attacks targeting multiple modalities [8, 24, 50]. More recently, a runtime defense approach has been proposed that monitors incoming LiDAR data to identify suspicious point clusters and employs a reinforcement learning-based search to remove potential adversarial points [45]. While this method demonstrates strong detection performance and adaptability, it incurs significant computational overhead, often requiring several seconds to process each LiDAR frame, which limits its suitability for real-time autonomous driving scenarios where rapid response is critical.

To date, no existing defense offers both high effectiveness and real-time efficiency, despite the fact that AVs operate in time-sensitive environments where even minor delays in threat detection can result in catastrophic consequences. These limitations underscore the urgent need for a real-time, generalizable, and practical defense mechanism against object-based LiDAR attacks.

However, developing such a defense presents several key challenges. First, AVs may utilize a variety of LiDAR object detection systems, each with different architectures and models. Therefore, the defense must be sufficiently general to integrate seamlessly with diverse detection pipelines. Second, attacks typically occur while the vehicle is in motion and the distance between the AV and the attack location is usually short (e.g., tens of meters), leaving a narrow window for reaction. Any delay in attack identification or mitigation can lead to unsafe maneuvers or collisions. Third, the attack location is unpredictable, as adversaries may launch attacks at arbitrary points along the road. Moreover, existing object-based LiDAR attacks employ a wide range of adversarial object configurations, varying in number, size, shape, and placement. In practice, defenders lack prior knowledge of these parameters, and the defense must be robust against such variability to remain effective under diverse and unknown attack strategies.

To address the above challenges, we propose a novel real-time defense mechanism against object-based LiDAR attacks in autonomous driving. The proposed solution is both detection model-agnostic and attack-agnostic, making it compatible with a wide range of LiDAR perception systems and effective against diverse attack strategies. It is designed for seamless integration into existing autonomous vehicle software stacks without requiring significant modifications to system components. Positioned between the sensing and perception modules in the AV pipeline, the defense processes incoming LiDAR point clouds in real time before they are passed to the downstream perception model. At the core of our mechanism is a generative model that enables the identification and removal of LiDAR points introduced by adversarial objects. This results in a clean, attack-free point cloud that enhances the robustness of subsequent object detection. We validate the effectiveness of our approach through comprehensive evaluations in both simulated and real-world settings, demonstrating strong defense performance, high generalizability, and low-latency operation suitable for real-time deployment.

Contributions. Our contributions are summarized as follows:

- We present, to the best of our knowledge, the first real-time defense mechanism against object-based LiDAR attacks, specifically designed for practical deployment in autonomous driving systems.
- We introduce a novel generative model that enables the efficient and accurate identification and removal of LiDAR points introduced by adversarial objects.
- Our method is both detection model-agnostic and attack-agnostic, requiring no prior assumptions about the number, shape, or placement of adversarial objects.
- We conduct extensive evaluations in both simulated and real-world environments, demonstrating that our defense significantly mitigates object-based LiDAR attacks with minimal runtime overhead.

2 Background and Related Work

2.1 LiDAR Object Detection in Autonomous Driving

LiDAR is a core sensing modality in autonomous driving, providing high-resolution, three-dimensional information about the surrounding environment. A LiDAR sensor works by emitting laser pulses in multiple directions and measuring the time it takes for the signals to bounce back after hitting nearby objects. This time-of-flight data is converted into a set of 3D spatial coordinates, collectively forming a point cloud. Each point in this cloud represents a precise location in space, typically defined by its (x, y, z) coordinates. Compared to camera-based sensors, LiDAR offers superior depth perception and is robust to changes in lighting conditions, making it particularly useful for object detection in diverse driving environments [11, 16]. In modern autonomous driving systems, LiDAR serves as a critical component of the perception stack, enabling vehicles to recognize nearby obstacles and navigate safely [3].

LiDAR object detection aims to identify and localize objects (e.g., vehicles, pedestrians, and cyclists) within the raw 3D point cloud data collected by the sensor. The typical detection pipeline begins by preprocessing the raw point cloud to convert it into a more structured format, such as voxel grids, bird's-eye-view (BEV) images, or range maps, using techniques like voxelization or projection [11, 41–43, 47]. These structured representations are then fed into deep learning models, commonly 3D Convolutional Neural Networks (CNNs) or point-based architectures such as PointNet++ [30] and PointPillars [25], which extract spatial and semantic features from the data. The output of the detection model typically includes a set of 3D bounding boxes, each characterized by the object's position (center coordinates), size (length, width, height), and orientation (e.g., yaw angle). A confidence score is also often included to indicate the likelihood that the bounding box corresponds to a real object. These predictions are then refined using a score threshold

to filter out low-confidence detections, followed by non-maximum suppression to eliminate redundant bounding boxes with significant overlap. LiDAR object detection serves as a foundational capability for safe and reliable autonomous navigation.

2.2 Object-Based LiDAR Attacks

While LiDAR sensors have become integral to autonomous driving systems due to their precise 3D perception capabilities, recent research has shown that LiDAR-based perception pipelines, especially those powered by DNNs, are vulnerable to various adversarial attacks. Early work in this area focuses on laser-based attacks [7, 9, 17, 18, 23, 32, 35, 36], where attackers use directed laser beams to inject false points into the LiDAR data stream. Although these attacks can disrupt perception, their practical deployment is limited by strict line-of-sight and timing requirements, which are hard to maintain in dynamic driving environments. To overcome these limitations, a growing body of work has shifted toward object-based attacks [1, 8, 10, 28, 38, 46, 48, 49, 51], where the attacker places physical adversarial objects in the environment to manipulate LiDAR outputs. These objects range from specially fabricated adversarial shapes to common, everyday items, and they can cause the perception system to miss objects. Due to their low cost and high practicality, object-based attacks pose a more severe real-world threat to AV safety. They are generally categorized into two types: attacks using *objects with specific, crafted shapes*, and attacks using *common, easily accessible objects*.

For attacks using objects with specific shapes, they rely on physically realizable adversarial objects whose geometry is carefully optimized to deceive LiDAR detection models. The shapes are typically non-standard and are generated in simulation before being materialized with tools like 3D printers. One of the earliest demonstrations of this idea comes from Cao et al. [10], who design object geometries that can fool a LiDAR detection system when placed in the environment. However, the effectiveness of this approach is limited to specific scenes. To increase generalizability, Tu et al. [38] propose universal adversarial objects that remain effective across different scenes and vehicles. Despite these improvements, the objects still face challenges in physical robustness, as errors can occur during LiDAR sampling due to surface irregularities. Subsequent work by Zhu et al. [48] improves surface design to better conform to LiDAR's discrete scanning behavior.

For attacks involving common objects, researchers have shown that everyday items can be repurposed as effective adversarial objects. Zhu et al. [51] find that there are spatially sensitive regions in the driving environment where placing reflective objects, such as drones or cardboard, can successfully deceive the LiDAR perception models used by AVs. For example, drones can be made to hover at specific locations around a target vehicle to hide it from detection. These attacks have also been shown to be effective across different LiDAR-based tasks, such as semantic segmentation [49]. Beyond attacks that only target the inference stage, Zhang et al. [46] propose a backdoor attack in which the model is first trained on poisoned data and later triggered during inference by placing a common object, such as a carrier bag or box, in the environment. In addition, Lou et al. [28] extend the use of common adversarial objects to trajectory prediction.

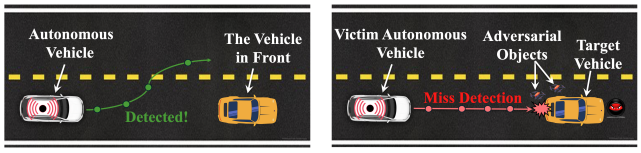
2.3 Countermeasures for Object-Based Attacks on LiDAR Perception

To defend against object-based LiDAR attacks in autonomous driving, a range of countermeasures have been proposed. However, to the best of our knowledge, no existing approach offers real-time attack detection with consistently high success rates. Most defenses suffer from limitations in either effectiveness, scalability, or practical deployment in real-world driving scenarios. Current defense strategies against object-based LiDAR attacks generally fall into three broad categories: adversarial training, multi-sensor fusion, and runtime LiDAR data analysis.

Adversarial Training. To enhance the robustness of LiDAR object detection models, several studies [8, 28, 38, 46, 49, 51] propose to use training-time defenses that incorporate adversarial examples. The central idea is to expose the model to adversarial point clouds, which are generated using known attack methods, during the offline training phase. These point clouds typically include perturbations caused by adversarial objects, which are added to the training data via data augmentation or adversarial training strategies. The goal is to increase the model's resilience to similar perturbations that may occur during deployment. However, a key limitation of these methods is their reliance on prior knowledge of specific attack types. In practice, it is often infeasible for defenders to anticipate the exact attack strategy that an attacker may employ. Different attack methods can lead to highly diverse configurations of adversarial objects, varying in shape, size, and placement. This variability makes it challenging for training-based defenses to generalize effectively. As a result, these approaches have shown only marginal effectiveness in reducing attack success rates [45].

Multi-Sensor Fusion. Another line of defense proposed in existing studies involves leveraging data from multiple sensors, such as cameras and radar, to compensate when LiDAR inputs are unreliable or manipulated [26, 33, 46, 49]. The intuition is that by cross-validating information across modalities, the victim AV may detect inconsistencies and maintain perception reliability under attack. While this approach can enhance system redundancy, it does not directly eliminate the underlying vulnerability of the LiDAR module itself. More critically, recent studies have shown that other sensing modalities, including cameras and radar, are also vulnerable to adversarial attacks [8, 24, 50]. A sophisticated attacker could exploit this weakness by launching coordinated attacks across multiple sensor types, rendering fusion-based defenses ineffective.

Runtime LiDAR Data Analysis. This category of defenses operates during runtime by analyzing and processing LiDAR point clouds collected as the AV navigates its environment. Hau et al. [19] propose identifying "shadow" regions, which are voids in the point cloud caused by occlusion, as indicators of hidden obstacles. Their method attempts to infer the physical sources of these voids by analyzing the geometric structure of the surrounding scene. While conceptually appealing, the approach suffers from high computational overhead, with processing times averaging tens of seconds per scene, making it impractical for real-time autonomous driving where low-latency response is essential. Zhu et al. [48] present an alternative defense that applies a smoothing algorithm to LiDAR scans in order to suppress adversarial patterns introduced by specially shaped physical objects. Although effective against the



(a) Without attack (b) With attack

Figure 1: An example of the vehicle hiding attack.

specific attack it targets, this method lacks robustness against more general attack scenarios. In particular, it fails when adversaries use common, easily obtainable objects instead of custom-designed shapes. Furthermore, the smoothing process may degrade the precision of detecting benign objects, thereby reducing its reliability in practice.

More recently, Zhang et al. [45] propose an online defense mechanism that continuously monitors the incoming LiDAR stream and extracts suspicious point clusters in front of the vehicle. This method uses a reinforcement learning-based search strategy to isolate and remove adversarial points from the identified cluster. While the approach demonstrates strong detection performance and adaptability, it imposes considerable runtime overhead due to the computationally intensive search process, which may take several seconds to process each LiDAR frame. Such latency limits its practicality in autonomous driving scenarios, particularly when the vehicle is traveling at high speeds. Moreover, the method assumes that attackers use only the minimal number of objects necessary for a successful attack. In practice, however, attackers may introduce redundant objects to increase attack robustness, potentially reducing the defense's effectiveness.

In contrast to these prior approaches, our work seeks to develop a real-time, attack-agnostic defense mechanism that imposes no assumptions on the number, size, or shape of adversarial objects, while remaining computationally efficient enough for deployment in practical autonomous driving systems.

3 Threat Model and Design Challenges

3.1 Threat Model

Attack Setting. This paper focuses on a widely studied class of object-based LiDAR attacks, referred to as *vehicle hiding attacks*, which have drawn increasing attention in the autonomous driving security literature [38, 46, 48, 49, 51]. As illustrated in Figure 1, such attacks aim to hide a target vehicle, which is often parked on the road, from the victim AV's object detection system by strategically placing adversarial objects around it. This can lead the AV to misinterpret its surroundings, potentially resulting in hazardous situations such as collisions. In line with prior work, we assume the attacker lacks access to the AV's real-time sensory data. However, the attacker can approximate the AV's view by collecting surrogate LiDAR scans from various angles and distances around the target scene. Additionally, the attacker may have either white-box or black-box access to the AV's LiDAR object detection model.

Defense Goal. Our objective is to design a real-time defense mechanism that can effectively mitigate object-based LiDAR attacks. A key design requirement is to ensure that the solution is agnostic to both the specific attack strategy and the underlying object detection model. The defense should integrate seamlessly into existing AV software stacks without requiring significant modifications to system components. We consider a practical setting where the defender has no prior knowledge of the adversarial objects (including their quantity, size, shape, or placement). Furthermore, the defender does not know in advance which road segment may be targeted by the attack.

3.2 Challenges and Design Rationale

Next, we outline the key challenges in achieving the defense goal and explain the design rationale behind our proposed solution.

Diverse LiDAR Object Detection Systems. Various LiDAR-based object detection systems have been developed for AVs, often adopting different detection models and architectures. Prior research has shown that many of these state-of-the-art LiDAR object detection models are vulnerable to object-based attacks. To ensure broad applicability and robustness, our defense mechanism is designed to be model-agnostic and compatible with diverse LiDAR object detection pipelines. Specifically, we propose a modular defense that can be integrated into the AV system as an independent module positioned between the sensing and perception modules. This module obtains raw LiDAR data from the sensing module, filters out potential adversarial threats, and outputs a cleaned point cloud to the perception module. Crucially, our approach does not rely on or require modification of the downstream object detection model, ensuring ease of integration with different AV platforms.

Real-Time Efficiency Requirement. Autonomous driving demands highly efficient, near-instantaneous decision-making, especially in response to adversarial threats. Attacks often occur when the AV is in motion, possibly at high speeds, and the distance between the victim AV and the target vehicle is typically short (e.g., tens of meters). As a result, the available reaction window is extremely limited. Any delay in identifying or mitigating an attack may lead to unsafe behavior or collisions. To meet these real-time constraints, our defense avoids computationally intensive search-based methods that aims to search out every adversarial object in the complex 3D space. Instead, we introduce a lightweight generative model, based on which we can directly generate a clean version of the point cloud from potentially contaminated input. By bypassing object-level analysis and instead generating a purified point cloud, our approach significantly reduces computational overhead and supports real-time response.

Unknown Attack Location. Another practical challenge is that the location of the attack is unknown in advance. An effective defense must therefore continuously monitor the driving environment without introducing unnecessary overhead or disruptions to the AV's normal operation. Naively applying the above mentioned generative model to every LiDAR frame would be inefficient, as attacks are rare in real-world driving scenarios. To address this, our system adopts a trigger-based activation strategy: the generative model-based defense is only engaged when a suspicious point cluster appears in front of the AV but is not reported by the object detection model. The rationale is that, during a successful attack, although the adversarial objects and the target vehicle are not detected, their corresponding LiDAR reflections still exist in the raw

point cloud. Therefore, if a point cluster exists in the scene but no corresponding detection is produced, it may indicate an ongoing attack, prompting the defense mechanism to perform further analysis and sanitize the data accordingly.

Unknown Attack Strategy. As discussed in Section 2.2, numerous object-based LiDAR attack methods have been proposed, each using different configurations of adversarial objects in terms of number, location, size, and shape. In practice, defenders have no prior knowledge of these properties, and the defense must remain effective regardless of the specific attack strategy. However, this variability makes it extremely difficult to train a model to directly identify and remove adversarial objects, because they can take virtually any form or placement in the environment. To overcome this, we shift our focus to the common element across all attacks: the target vehicle that the attacker aims to hide. Existing studies have shown that attackers typically choose regular-sized vehicles as targets to maintain stealth. Given the limited diversity of real-world vehicle shapes and sizes, it is more tractable to learn to extract the information about the hidden vehicle than to detect the surrounding adversarial artifacts. Therefore, our defense is designed to estimate and extract the surface points of the target vehicle from mixed point clusters containing both the hidden vehicle and nearby adversarial objects. These estimated surface points can then be used to guide the removal of LiDAR points associated with the adversarial objects.

4 Methodology

4.1 Overview

In this paper, we propose a novel real-time defense mechanism capable of removing LiDAR points generated by adversarial objects before the data is fed into downstream LiDAR perception models. As illustrated in Figure 2, the proposed defense can be seamlessly integrated into the pipeline of an autonomous driving system, positioned between the sensing and perception modules. The defense mechanism operates in three major steps: (1) suspicious point cluster extraction, (2) vehicle surface estimation, and (3) adversarial object removal.

Step 1: Suspicious Point Cluster Extraction. As discussed in Section 3.1, this paper considers vehicle hiding attacks, in which the attacker aims to hide a target vehicle from the LiDAR-based object detection system. Although these attacks may cause the detection model to miss the target vehicle and nearby adversarial objects, the corresponding LiDAR reflections from both still exist in the raw point cloud. Therefore, the first step of the proposed defense mechanism is to continuously monitor the incoming LiDAR point clouds and examine whether a suspicious cluster exists within a designated region of interest (for example, the road segment of a specified length directly in front of the victim AV). Such a point cluster can be extracted using segmentation and clustering algorithms. Intuitively, if a cluster is present in the scene but the object detection model fails to detect any object at that location, it is flagged as suspicious. The cluster is likely to contain both the hidden vehicle and the surrounding adversarial objects.

Step 2: Vehicle Surface Estimation. The second step is executed only when a suspicious point cluster has been identified in Step 1. The goal of this step is to estimate the surface points of

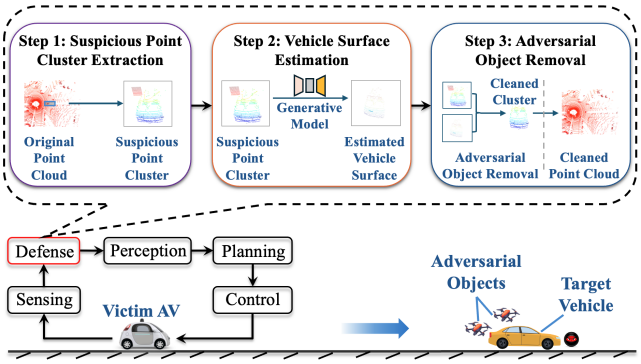


Figure 2: Overview of the proposed defense mechanism.

the target vehicle by leveraging the extracted suspicious cluster and applying a transformer-based point cloud generative model. The underlying intuition is that adversarial objects are typically positioned in the vicinity of the target vehicle but do not overlap with it. As a result, the estimated vehicle surface can act as a reliable reference for distinguishing genuine vehicle points from adversarial ones. More specifically, if a LiDAR point falls outside the estimated vehicle surface, there is a high likelihood that it originates from an adversarial object rather than from the vehicle itself.

Step 3: Adversarial Object Removal. In the third step, points generated by adversarial objects are identified and removed by comparing the extracted suspicious cluster with the estimated vehicle surface obtained in Step 2. This comparison is performed by measuring the distance between each point in the suspicious cluster and the nearest point on the estimated surface. If the distance exceeds a predefined threshold, the point is classified as originating from an adversarial object and is removed from the suspicious cluster. After this filtering process, the cleaned cluster, now free of adversarial points, is reinserted into the original point cloud by replacing the extracted suspicious region. The resulting cleaned point cloud can then be utilized to enable more robust and reliable object detection.

It is important to note that the estimated surface points obtained in Step 2 are not used directly for object detection. This is because the goal of Step 2 is not to precisely reconstruct the original point cluster of the target vehicle in the raw LiDAR point cloud, which would necessitate a more complex and computationally intensive model. Instead, we employ a relatively lightweight generative model that approximates the target vehicle's surface sufficiently well to serve as a reference for filtering out adversarial points. While this estimation may lack the level of detail required for precise vehicle detection, it provides adequate structural cues to enable reliable removal of adversarial objects. This design offers a balance between defense effectiveness and computational efficiency. Next, we discuss the detailed design of each step.

4.2 Suspicious Point Cluster Extraction

In this step, the collected LiDAR data is analyzed to identify the suspicious point cluster located in front of the victim AV. This cluster may correspond to the target vehicle hidden through adversarial attacks. The effectiveness of this extraction strategy has

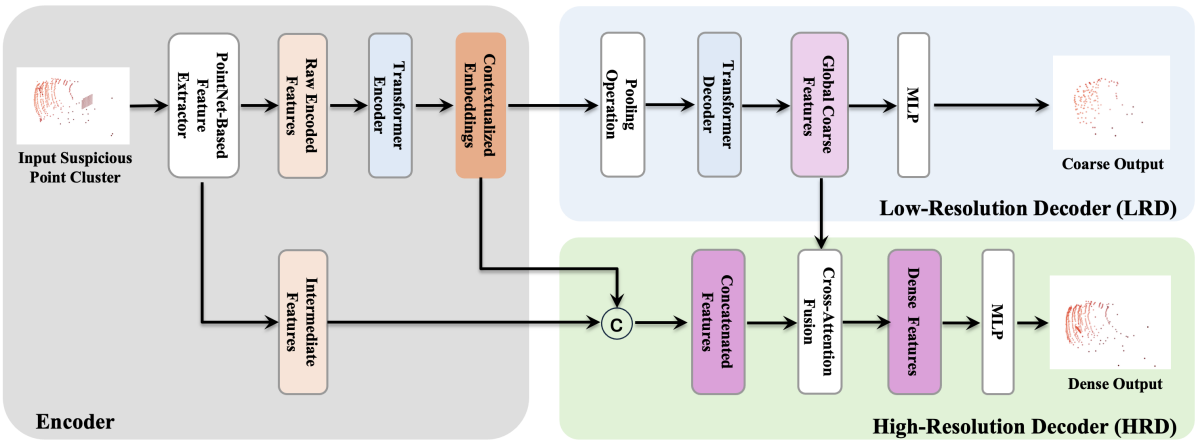


Figure 3: Overview of the vehicle surface estimation model.

been demonstrated in prior defense research against object-based LiDAR attacks [45], and we adopt a similar approach in this work. Specifically, we first apply the RANSAC (Random Sample Consensus) algorithm [13] to identify and remove points corresponding to the road surface. This enables isolation of LiDAR points above the ground, which are more likely to represent objects of interest. Next, we define a Region of Interest (ROI) based on typical attack scenarios. In many existing object-based LiDAR attacks, the attacker attempts to hide a vehicle positioned in front of the victim AV by strategically placing adversarial objects surrounding it. Accordingly, we define an ROI extending ahead of the AV to capture such potential threats. Within this ROI, the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm is applied to segment the remaining above-ground points into clusters. In some cases, multiple clusters may be segmented. We only focus on the cluster whose shape is most similar to a vehicle. If the cluster is present within the ROI but is not recognized as vehicle by the LiDAR perception model, it is flagged as a suspicious point cluster and passed to Step 2 for further analysis.

It is important to note that, due to the complexity of real-world environments, suspicious clusters can also arise from benign factors such as roadside billboards or environmental artifacts like snow or fog. Therefore, it is neither practical nor reliable to trigger defensive driving maneuvers based solely on the presence of such clusters. Additional processing is needed to verify whether the suspicious cluster corresponds to a hidden vehicle or represents a false alarm.

4.3 Vehicle Surface Estimation

The goal of this step is to estimate the surface of the target vehicle, which serves as a reference for identifying surrounding adversarial objects. To achieve this, we design a generative model that directly predicts the surface points of the target vehicle based on the suspicious point cluster extracted in Section 4.2. However, developing an effective generative model for this task is challenging due to the variability in LiDAR point density.

For a given vehicle, the number of LiDAR points captured on its surface depends heavily on the distance between the vehicle and the LiDAR sensor. Most LiDAR sensors operate with a fixed angular resolution, emitting beams at consistent angular intervals (e.g., 0.2° horizontally). As a result, objects that are closer to the sensor receive denser point coverage because the angular beams intersect with them at smaller spatial intervals. Conversely, objects farther away receive sparser coverage, as the same angular gaps cover larger physical areas and fewer beams strike the object. Therefore, the generative model must be trained to extract meaningful geometric features from both dense and sparse point clouds. It should also generalize well during inference to generate accurate surface point estimations for target vehicles observed at varying distances. This capability is crucial for real-time defense, as the victim AV is continuously approaching the target vehicle, and the defense must remain effective regardless of the relative distance during the detection process.

To meet this challenge, we propose a novel generative model with a dual-branch encoder-decoder architecture. As shown in Figure 3, the model includes a transformer-based encoder that captures both global and local geometric features, followed by two decoder branches: a Low-Resolution Decoder (LRD) and a High-Resolution Decoder (HRD). The LRD is designed to generate a coarse, sparse approximation of the vehicle's surface, while the HRD produces a dense and detailed reconstruction. This dual-decoder architecture improves the model's adaptability to varying input densities. When the target vehicle is far from the sensor, the suspicious point cluster is typically sparse and lacks detailed structure. In such cases, the LRD generates a coarse surface that offers more reliable estimation for identifying and filtering adversarial points. Conversely, when the vehicle is closer and the point cloud is denser, the HRD generates fine-grained surface details, enabling more accurate identification of adversarial objects. This design ensures robust surface estimation under diverse sensing conditions, enhancing the effectiveness of the subsequent defense step.

Encoder. The encoder is designed to extract both global and local features from the input suspicious point cluster. It begins with a PointNet-based feature extraction module, which processes the input point cloud using a series of multilayer perceptrons (MLPs), and each layer is followed by batch normalization and ReLU activation. The input to this module is a point cluster of shape $(N, 3)$,

where N is the number of points in the cluster, and each point is represented by its (x, y, z) coordinates. In our implementation, we use three MLP layers with output channels of 64, 128, and 1024, respectively. The output of this stage is a set of point-wise features with shape $(N, 1024)$, representing raw encoded features for each point. A subsequent linear layer followed by GELU activation is applied to reduce the dimensionality of these features, resulting in an embedding of shape $(N, 256)$. To preserve fine-grained geometric detail, we also retain intermediate low-level features extracted from earlier MLP layers. These features capture local structure and are later fused in the decoding stage to enhance surface detail estimation. Additionally, we compute sinusoidal positional encodings for each point and add them to the point-wise features to incorporate spatial information. The resulting features are then passed through a stack of transformer layers to model long-range dependencies and capture global contextual information. Each transformer layer consists of a multi-head self-attention mechanism (with 4 heads) and a feed-forward network with a hidden dimension of 512. We use a stack of four such layers, producing context-aware point-wise features of shape $(N, 256)$, which serve as the encoded representation for downstream processing.

Low-Resolution Decoder (LRD). The LRD aims to generate a coarse approximation of the vehicle surface, capturing the global structure of the vehicle surface. The input to the LRD originates from the encoded point-wise feature tensor of shape $(N, 256)$. To reduce redundancy and focus on high-level structure information, we apply a pooling operation to the encoded representation and downsample each sample to 128 representative tokens. This produces a token tensor of shape (128, 256), which retains essential global information while filtering out noise and minor variations. These tokens are then fed into a transformer-based decoder composed of four stacked transformer layers. Each layer applies a multi-head attention mechanism (with 4 heads) to refine the token embeddings and capture inter-point relationships. The layers also include feed-forward networks with hidden dimension 512, along with residual connections and layer normalization, to stabilize training and maintain feature integrity. This transformer-based decoding process enables the model to learn long-range dependencies and the overall surface topology of the vehicle. Such global understanding is essential for producing accurate surface estimates, which serve as a foundation for identifying and removing surrounding adversarial objects in the next step. Finally, the refined features are passed through a linear layer with GELU activation to regress the (x, y, z) coordinates of the predicted vehicle surface points. We set the number of output points to 128. The resulting output of the LRD is a sparse point cloud that provides a coarse yet structurally meaningful approximation of the vehicle surface.

High-Resolution Decoder (HRD). The HRD is designed to produce a detailed estimation of the vehicle surface with higher point density. In this branch, the global contextualized point-wise embeddings of shape $(N, 256)$ are concatenated with low-level intermediate features extracted from an earlier layer in the encoder. This fusion enables the decoder to leverage both high-level semantic information and fine-grained geometric details. The concatenated features are passed through a linear layer followed by GELU activation, yielding a fused representation of shape $(N, 256)$. These fused features are then transformed into a dense set of tokens of shape $(512, 256)$, representing a more fine-grained encoding of the surface structure. In parallel, the intermediate embeddings from the LRD branch are upsampled to match this resolution, producing another token set of shape $(512, 256)$. A multi-head cross-attention module (with 4 heads) is applied to refine the dense features. In this setup, the dense tokens act as queries, while the upsampled LRD tokens serve as keys and values. The cross-attention mechanism enables the HRD to incorporate coarse global cues from the LRD into its high-resolution surface estimation. A residual connection adds the attention output back to the original dense tokens, facilitating stable learning and information preservation. Finally, the refined dense tokens are passed through a linear layer to regress the (x, y, z) coordinates of the predicted vehicle surface points. In this branch, we set the number of output points to 512. The HRD generates a dense point cloud that offers a high-resolution, structurally detailed approximation of the vehicle surface, supporting precise adversarial object removal in downstream processing.

In practice, the appropriate decoder output is selected based on the density of the extracted suspicious point cluster. For sparse inputs, where the number of points falls below a predefined threshold, we use the surface points generated by the LRD. For denser inputs with richer structural detail, the output from the HRD is used instead.

4.4 Adversarial Object Removal

The goal of this step is to remove LiDAR points generated by adversarial objects using the estimated vehicle surface obtained in Section 4.3. After selecting the appropriate vehicle surface estimate, we compare it with the extracted suspicious cluster to identify points likely originating from adversarial objects. Specifically, for each point p_i in the suspicious cluster, we compute its distance d_i to the nearest point on the estimated vehicle surface $\mathbf{P}_v$. This distance is defined as:

$$d_i = \min_{p_j \in \mathbf{P}_v} \|p_i - p_j\|_2^2. \tag{1}$$

Here, p_i denotes the (x, y, z) coordinates of the i-th point in the suspicious cluster, and p_j represents the (x, y, z) coordinates of the j-th point in the generated vehicle surface point set $\mathbf{P}_v$.

If the distance d_i exceeds a predefined threshold $\mathcal{T}$, the point is considered an outlier relative to the estimated vehicle surface. This suggests a high likelihood that the point originates from an adversarial object rather than the target vehicle. All such outlier points are removed from the original collected LiDAR point cloud, resulting in a refined point cloud that contains only points that are not generated by adversarial objects. The cleaned point cloud is then passed to the downstream object detection module, allowing the system to generate robust detection results in the presence of adversarial attacks.

4.5 Model Training and Threshold Selection

To ensure the effectiveness of the proposed defense mechanism, two key challenges must be addressed: (1) how to train the parameters of the generative model so that the estimated vehicle surface can reliably support the adversarial object removal step, and (2) how to select an appropriate threshold value $\mathcal{T}$ to enable accurate separation of adversarial points from vehicle surface points.

To address these challenges, we propose to jointly optimize the generative model parameters and the threshold during the training stage in order to maximize overall defense performance. The underlying intuition is that the optimal threshold depends on the quality of the generated vehicle surface, while the optimal surface generation is also influenced by the threshold used for identifying adversarial points. Specifically, we formulate the following optimization problem:

$$\min_{\theta,\mathcal{T}} \ \mathcal{L}_c + \alpha \cdot \mathcal{L}_p + \beta \cdot \mathcal{L}_d, \tag{2}$$

where θ denotes the parameters of the generative model described in Section 4.3, and $\mathcal{T}$ is the distance threshold used during adversarial object removal. The objective function combines three loss components: a **point-wise classification loss ($\mathcal{L}_c$)**, which quantifies the accuracy of identifying adversarial points; a **Chamfer loss ($\mathcal{L}_p$)**, which measures the geometric similarity between the estimated and ground-truth vehicle surfaces; and a **structural shape loss ($\mathcal{L}_d$)**, which captures local surface geometry by comparing curvature patterns. Both decoder branches (LRD and HRD) are trained using this combined loss function. We describe the loss terms in more detail below.

Point-Wise Classification Loss $\mathcal{L}_c$. As described in Section 4.4, the object removal step classifies each point in the suspicious point cluster as either part of an adversarial object or the target vehicle surface using a distance threshold. In the training data, each point is labeled with a ground-truth binary value: 1 for adversarial points and 0 for vehicle surface points. The classification loss is defined as the mean squared error (MSE) between predicted and ground-truth labels:

$$\mathcal{L}_c = \frac{1}{N}\sum_{i=1}^{N}\Big(c_{\text{pred},i} - c_{\text{gt},i}\Big)^2, \tag{3}$$

where $c_{\text{gt},i} \in \{0, 1\}$ is the ground-truth label for the i-th point in the suspicious point cluster, and $c_{\text{pred},i}$ is the predicted label. Since the original threshold-based classification is non-differentiable, we approximate it using a sigmoid-based formulation:

$$c_{\text{pred},i} = \text{Sigmoid}\Big(\mu \cdot \big(d_i - \mathcal{T}\big)\Big), \tag{4}$$

where d_i is the distance from point p_i to the estimated vehicle surface (defined in Eq. (1)), $\mathcal{T}$ is the threshold for classification, and μ is a scaling factor that controls the steepness of the transition.

Chamfer Loss $\mathcal{L}_p$. To ensure accurate surface estimation, we adopt the Chamfer Distance [4, 15, 44] to measure the geometric similarity between the predicted and ground-truth vehicle surface point clouds. The Chamfer Distance computes the average nearest-neighbor distance between two point sets in both directions, penalizing points that are not well-aligned between the predicted and reference surfaces. This encourages the predicted surface to closely approximate the true geometry of the target vehicle. Formally, the Chamfer loss is defined as:

$$\begin{aligned}\mathcal{L}_p &= \mathcal{D}_{\text{Chamfer}}(\mathbf{P}_v, \mathbf{P}_v^*) \\ &= \frac{1}{|\mathbf{P}_v|}\sum_{p_i \in \mathbf{P}_v}\min_{p_j \in \mathbf{P}_v^*}\|p_i - p_j\|_2 + \frac{1}{|\mathbf{P}_v^*|}\sum_{p_{j'} \in \mathbf{P}_v^*}\min_{p_{i'} \in \mathbf{P}_v}\|p_{i'} - p_{j'}\|_2,\end{aligned} \tag{5}$$

where $\mathbf{P}_v$ and $\mathbf{P}_v^*$ represent the predicted and ground-truth surface point sets, respectively. The first term penalizes predicted points that are far from any ground-truth points, while the second term penalizes ground-truth points that are not well matched by the prediction. This bidirectional structure helps ensure both coverage and compactness, making Chamfer Distance a powerful tool for surface estimation tasks.

Structural Shape Loss $\mathcal{L}_d$. While the Chamfer loss $\mathcal{L}_p$ ensures global geometric alignment between the predicted and ground-truth vehicle surfaces, it primarily captures point-level proximity and may fail to enforce the preservation of fine-grained structural properties, such as surface curvature and continuity, that are critical for distinguishing vehicle surfaces from adversarial artifacts. To address this limitation, we introduce a structural shape loss that explicitly measures and penalizes discrepancies in local curvature between the predicted and reference surfaces. This loss encourages the generative model to estimate not only the overall shape but also the intrinsic geometric characteristics of the surface, thereby enhancing the robustness of subsequent adversarial object removal. Formally, the loss is defined as:

$$\mathcal{L}_d = \frac{1}{N}\sum_{i=1}^{N}\big(\kappa_{\text{pred},i} - \kappa_{\text{gt},i}\big)^2, \tag{6}$$

where $\kappa_{\text{pred},i}$ is the estimated curvature at the i-th point p_i on the predicted surface, and $\kappa_{\text{gt},i}$ is the curvature of the closest point on the ground-truth surface. Curvature is computed following the method described in [29]. For each point p_i, we first identify its k-nearest neighbors and compute the local covariance matrix:

$$\mathbf{C}_i = \sum_{j=1}^{k}(p_j - p_i)(p_j - p_i)^T, \tag{7}$$

where p_j denotes the j-th neighboring point of p_i. The eigenvalues $\lambda_1 \le \lambda_2 \le \lambda_3$ of $\mathbf{C}_i$ are then calculated, and the curvature at point p_i is defined as:

$$\kappa_i = \frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3}. \tag{8}$$

This formulation captures the flatness or sharpness of local surface regions, with lower curvature indicating flatter areas and higher curvature indicating sharper features. By minimizing the discrepancy between the predicted and ground-truth curvatures, the structural shape loss helps enforce local consistency.

When computing the Chamfer loss $\mathcal{L}_p$ and the structural shape loss $\mathcal{L}_d$, we normalize the ground-truth surface point cloud of the target vehicle to match the output size of each decoder branch. Specifically, for the LRD branch, the ground-truth surface is resampled to contain exactly 128 points. If the original point cloud has more than 128 points, the excess points are randomly downsampled; if it contains fewer than 128 points, additional points are added via padding (e.g., zero-padding or duplication). The HRD branch follows the same procedure, with the ground-truth surface adjusted to contain exactly 512 points. Both the LRD and HRD branches operate concurrently during training, and their respective loss values are computed independently. The final training loss is obtained by summing the loss contributions from both branches, ensuring that the model learns to reconstruct vehicle surfaces accurately at multiple levels of resolution.

Optimization. The optimization involves two interdependent components: the generative model parameters θ and the distance threshold $\mathcal{T}$ used for adversarial object identification. The optimal threshold $\mathcal{T}$ depends on the quality of the generated vehicle surface, which is governed by θ. Conversely, learning an optimal θ requires a reliable threshold to accurately distinguish adversarial points from true surface points. To address this mutual dependency, we adopt an alternating optimization strategy, which is well-suited for optimizing problems with coupled variables. In this framework, the parameters θ and $\mathcal{T}$ are updated alternately in two stages. In the first stage, we fix the threshold $\mathcal{T}$ and optimize the model parameters θ, ensuring that the generative model produces a surface estimation that not only approximates the target geometry but also supports effective adversarial object identification under the current threshold. In the second stage, we fix θ and optimize the threshold $\mathcal{T}$ to best separate adversarial points from vehicle surface points, given the current surface predictions. These two steps are repeated iteratively until a convergence criterion is satisfied, resulting in a jointly optimized surface generator and threshold tailored for robust defense performance.

5 Performance Evaluation

5.1 Experimental Setting

LiDAR Object Detection Models. We evaluate our defense using two state-of-the-art LiDAR object detection models: PIXOR [42] and PointPillars [25]. PIXOR performs real-time, anchor-free detection using a bird's-eye view (BEV) projection of the point cloud. PointPillars partitions the point cloud into vertical columns (pillars) and applies a PointNet-based encoder, followed by 2D convolutions in BEV space. Both models are configured with a detection confidence threshold of 0.5.

LiDAR Dataset. We conduct our evaluation on the KITTI dataset [16], which contains 7,481 training samples and 7,518 test samples collected from real-world driving scenarios. Since our focus is on object-based attacks that aim to hide a front vehicle, we select samples in which a vehicle appears directly ahead of the ego vehicle in the same lane.

Attack Methods. In our evaluation, we consider several state-of-the-art object-based LiDAR attack methods, covering both attacks that utilize specially shaped adversarial objects and those that exploit common items. The evaluated attacks include:

- *AdvObj* [38]: This attack is designed for hiding a target vehicle from the LiDAR detection system of an AV by placing a 3D-printed object with an adversarially designed shape on the rooftop of the target.
- *AE-Morpher* [48]: This attack is an enhanced version of AdvObj, in which the adversarial object is still placed on the target vehicle. This method further optimizes the adversarial object's geometric properties to improve physical robustness and increase attack effectiveness.
- *AdvLoc* [51]: In contrast to the above attack methods that rely on adversarial objects with specific shapes, AdvLoc uses an optimization process to identify vulnerable spatial locations around the target vehicle where common objects can be placed to mislead the LiDAR detection model adopted by the victim AV and hide the target.
- *BALiDAR* [46]: This method is designed as a backdoor attack that compromises both the training and inference phases by injecting poisoned point clouds into the training data of the LiDAR detection model. At inference time, the attacker can place a common object on the rooftop of a target vehicle to trigger the backdoor and mislead the detection model.

Baselines. We consider three runtime LiDAR data analysis methods as baselines: *RLDef* [45], *smoothing defense* [48], and *shadow detection*[19]. RLDef is an online defense mechanism that continuously monitors the incoming LiDAR stream and extracts suspicious point clusters in front of the AV. It employs a reinforcement learning-based search strategy to remove adversarial points from the identified cluster. The smoothing defense method applies a smoothing algorithm to LiDAR scans to regularize irregular boundaries of the adversarial object and suppress adversarial perturbations. The shadow detection method identifies "shadow" regions (voids in the point cloud caused by occlusion) as potential indicators of hidden obstacles. It infers the presence of such obstacles by analyzing the geometric structure of the surrounding scene.

Evaluation Metrics. We measure the performance of our defense using the following metrics:

- *Detection rate (DR)*: This metric is defined as the percentage of attacked LiDAR frames in which the hidden target vehicle is successfully detected by the defense, relative to the total number of attacked frames. A higher detection rate indicates stronger defense effectiveness.
- *Runtime (RT)*: This metric is defined as the average time required to process a single attacked LiDAR frame and detect the hidden vehicle. This includes the runtime of both the defense mechanism and the downstream LiDAR detection model, which may vary depending on the model used. Lower runtime reflects better computational efficiency.

Additional Settings. In our experiments, the threshold for selecting the appropriate decoder is set to 256, based on an analysis of typical per-vehicle point counts at various distances in the KITTI dataset. Specifically, if the number of points in the extracted suspicious point cluster exceeds 256, the output of the HRD is used for adversarial object removal; otherwise, the LRD output is applied. This ensures that dense clusters are processed with the high-resolution decoder, while sparser clusters are handled by the low-resolution decoder. In practice, the threshold can be tuned to accommodate different LiDAR sensors or data distributions. The experiments are conducted on a platform equipped with an Intel i9-10920X processor and an NVIDIA RTX 6000 GPU. Additionally, we evaluate the proposed defense on the NVIDIA Jetson AGX Orin to assess its real-time performance on resource-constrained edge computing platforms.

5.2 Overall Performance

Performance Under Different Attacks and Distance Ranges. We begin by evaluating the effectiveness and efficiency of our proposed defense mechanism under different attacks. All four attack methods are implemented following the settings described in their respective original papers. Specifically, for the AdvObj and AE-Morpher attacks, we follow the original configurations by generating a uniquely shaped adversarial object for each method and

Table 1: Defense performance under different attacks and distance ranges.

Attack	10-20 m				20-30 m				30-40 m				40-50 m			
	DR (%)		RT (s)		DR (%)		RT (s)		DR (%)		RT (s)		DR (%)		RT (s)	
	Ours	RLDef	Ours	RLDef	Ours	RLDef	Ours	RLDef	Ours	RLDef	Ours	RLDef	Ours	RLDef	Ours	RLDef
AdvLoc	**93.3**	81.7	**0.056**	2.9	**93.3**	80.0	**0.055**	2.5	**91.7**	83.3	**0.058**	2.6	**90.0**	86.7	**0.054**	2.1
BALiDAR	**90.0**	76.7	**0.054**	3.1	**88.3**	75.0	**0.056**	3.2	**93.3**	81.6	**0.056**	3.1	**91.7**	80.0	**0.056**	2.7
AdvObj	**91.7**	76.7	**0.055**	2.8	**93.3**	80.0	**0.056**	2.6	**93.3**	85.0	**0.056**	2.4	**88.3**	88.3	**0.055**	2.3
AE-Morpher	**95.0**	81.7	**0.056**	2.9	**90.0**	80.0	**0.055**	2.8	**91.7**	83.3	**0.055**	2.6	**93.3**	81.7	**0.056**	2.5

Table 2: Comparison with other baseline methods.

Defense	AdvLoc		BALiDAR	
	DR (%)	RT (s)	DR (%)	RT (s)
Smoothing Defense	11.7	0.28	4.0	0.29
Shadow Detection	83.3	17.1	83.3	18.2
Ours	**93.3**	**0.056**	**91.7**	**0.055**

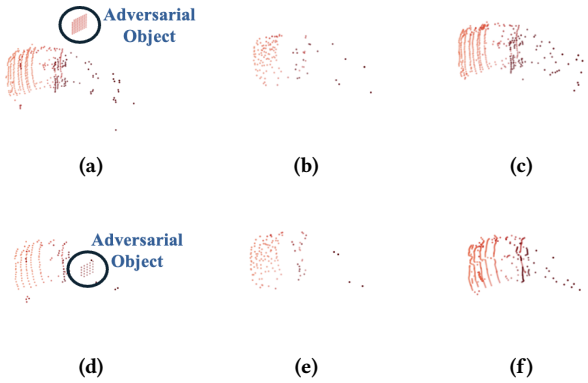


Figure 4: Visualization of generated vehicle surfaces for two test samples (one per row). (a) and (d): Input suspicious point clusters; (b) and (e): LRD outputs; (c) and (f): HRD outputs.

placing it on the rooftop of the target vehicle. For the AdvLoc attack, we use drone-suspended billboards as adversarial objects, positioning them at strategically identified adversarial locations near the target vehicle. In the case of the BALiDAR attack, we adopt a spherical object with a radius of 0.4 meters as the backdoor trigger. Unless otherwise specified, PIXOR is used as the default object detection model in all subsequent experiments. To train the generative model, we construct a set of adversarial training samples by injecting adversarial objects (generated using the above attack methods) into clean point clouds collected from the KITTI dataset.

Table 1 reports both the detection rate (DR) and runtime (RT) of our defense and the baseline RLDef across varying attack scenarios. We consider four distance ranges between the target vehicle and the victim AV: 10–20 m, 20–30 m, 30–40 m, and 40–50 m. For each distance range, we randomly select 60 successfully attacked LiDAR frames for evaluation. Note that the average recalls achieved by the detection model without any defense under the four attacks (AdvLoc, BALiDAR, AdvObj, and AE-Morpher) are 14.0%, 11.0%, 25.6%, and 16.0%, respectively. These results demonstrate that these attacks are effective and can significantly degrade the performance of the detection model. Table 1 show that our defense consistently outperforms RLDef across all attack methods and distance ranges. In most cases, the detection rate of our approach exceeds 90%, demonstrating high effectiveness in detecting hidden vehicles. Moreover, our method also achieves much lower runtime across all settings, with an average processing time of less than 0.06 seconds per LiDAR frame, which is well within the real-time constraints of autonomous driving systems. These findings highlight the advantages of our approach in terms of both robustness and efficiency, making it a practical solution for defending against object-based LiDAR attacks in real-world AV deployments

We also compare our defense with two additional baselines: the smoothing defense method and the shadow detection method. For each of these baseline methods, we evaluate performance under both the AdvLoc and BALiDAR attacks. In this experiment, we use the same generative model as described above. For each scenario, we randomly select 60 LiDAR frames from the 10–50 meter distance range between the target vehicle and the victim AV for evaluation. The results are presented in Table 2, and they further demonstrate the advantages of our proposed defense. The smoothing defense performs poorly in terms of detection rate because it is primarily designed to suppress adversarial objects with specific geometric patterns. In contrast, both the AdvLoc and BALiDAR attacks leverage common objects to achieve their goals, making them more difficult to counter using geometric smoothing alone. Although the shadow detection method is effective at identifying obstacles in front of the victim AV, its high processing latency makes it unsuitable for real-time autonomous driving applications. Moreover, shadow detection can only determine the presence of an obstacle and cannot work in conjunction with existing object detection models to infer additional object properties such as class labels, as our defense does.

Visualization of Generated Vehicle Surfaces. In Figure 4, we present some visual examples of the vehicle surface points generated by our proposed generative model. The first and second rows correspond to two different test samples. In each row, the first image (Figure 4a or 4d) shows the input suspicious point cluster; the second (Figure 4b or 4e) and third (Figure 4c or 4f) images display the outputs from the LRD and HRD, respectively. For the two examples, we can see that when the target vehicle is close to

Table 3: Defense performance with PointPillars.

Distance range	10-20 m	20-30 m	30-40 m
DR (%)	91.7	86.7	88.3
RT (s)	0.031	0.030	0.031

the LiDAR sensor, which results in higher point density in the suspicious cluster, the HRD is able to generate a detailed and accurate surface closely aligned with the ground truth, as seen in the first row. However, when the vehicle is farther away and the input is sparser, the HRD may introduce inaccuracies, such as generating extraneous surface points outside the vehicle's actual geometry (e.g., Figure 4f). In contrast, the LRD produces more reliable surface estimates under sparse input conditions. These examples demonstrate the effectiveness of our generative model and highlight the importance of using dual decoder branches. Dynamically selecting the decoder output based on the input cluster's density ensures more accurate surface estimation, which is essential for reliable adversarial object removal.

5.3 Defense Analysis

Impact of LiDAR Object Detection Model. The proposed defense mechanism is inherently detection model-agnostic, meaning it operates independently of any specific LiDAR object detection architecture. As it does not rely on internal information from the detection model, it can be seamlessly integrated with a wide range of existing LiDAR perception systems. To validate the generalizability of our approach, we evaluate its performance using another widely adopted detection model—PointPillars. Table 3 reports the defense results under the AdvLoc attack, using PointPillars for object detection. We consider three distance ranges between the target vehicle and the victim AV and randomly sample approximately 50 successfully attacked LiDAR frames per range. The results show that our defense maintains high detection rates across all cases while achieving low runtime. The runtime is notably shorter than that observed with PIXOR, owing to the greater computational efficiency of the PointPillars model.

Defense Transferability. In practice, it is often difficult for defenders to anticipate the specific types of attacks that a victim AV may encounter. To assess the generalizability of our defense, we evaluate its performance in scenarios where the attack types used during training and testing differ. Specifically, we train the generative model on adversarial examples generated by two combined attack methods (either AdvLoc+BALiDAR or AdvObj+AE-Morpher) and then test the defense against the remaining two attacks. Detection rates for these cross-attack evaluations are reported in Table 4. The first column show the attack methods used for training, and the first row presents those used for testing. The results demonstrate that our proposed defense maintains strong performance across all cases, achieving detection rates above 85% even when faced with previously unseen attack types.

Defense Performance When Training with Randomly Generated Point Clusters. To further assess the generalizability of our proposed defense, we also explore a more challenging setting

Table 4: Detection rate (%) for defense transferability.

Attack	AdvLoc	BALiDAR	AdvObj	AE-Morpher
AdvLoc+BALiDAR	-	-	91.7	88.3
AdvObj+AE-Morpher	88.3	86.7	-	-

Table 5: Detection rate (%) when training with randomly selected point clusters.

	10-20 m	20-30 m	30-40 m	40-50 m
Random cluster 1	90.0	88.3	86.7	76.7
Random cluster 2	85.0	86.7	78.3	71.6

in which the generative model is trained without using any adversarial examples generated by known object-based LiDAR attacks. Instead, we construct a training set by inserting randomly generated point clusters into clean LiDAR point clouds and treating them as adversarial objects. Specifically, we create two random clusters, with one (cluster 1 in Table 5) having a relatively larger size, and use each to train a separate generative model. In this experiment, we evaluate defense performance against the AdvLoc attack. Table 5 presents the detection rates within different distance ranges between the target vehicle and the victim AV. Despite not being exposed to any actual adversarial objects during training, the model achieves respectable detection performance, with detection rates generally ranging from 70% to 90%. While there is a slight drop in performance compared to models trained on real attack examples, particularly at longer distances, these results indicate that learning to distinguish clean vehicle surfaces from random point clusters can still offer meaningful robustness against unseen threats. This suggests that the proposed defense mechanism can retain practical effectiveness even in scenarios where the attacker's strategy is completely unknown.

Impact of Adversarial Object Removal Threshold $\mathcal{T}$. To evaluate the effectiveness of the threshold $\mathcal{T}$ derived from the formulated optimization problem, we compare the defense performance using the optimized value against several alternative threshold values. In this experiment, we adopt the setup described in Section 5.2, where the generative model is trained using adversarial examples from all four attack methods: AdvLoc, BALiDAR, AdvObj, and AE-Morpher. The optimized threshold value derived from training is $\mathcal{T} = 0.08$. Figure 5 shows the impact of varying $\mathcal{T}$ on detection performance across different attacks. Each curve represents an individual attack, with detection rate (y-axis) plotted against threshold values (x-axis). As shown, the value $\mathcal{T} = 0.08$ consistently yields the best overall performance, while both lower and higher thresholds lead to reduced effectiveness. This demonstrates the value of our joint optimization strategy for threshold selection. Moreover, the results highlight the importance of tuning $\mathcal{T}$ to balance the trade-off between removing adversarial points and preserving the geometric integrity of legitimate vehicle surfaces.

False Positives Generated by the Defense. As described in Section 4.2, the second step of our defense is triggered when a point cluster appears in front of the victim AV but no corresponding

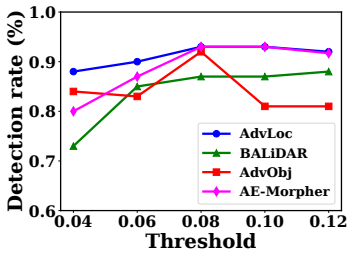


Figure 5: Defense performance under different values of threshold $\mathcal{T}$.

bounding box is reported by the perception model. In practice, benign objects such as roadside billboards may produce suspicious-looking clusters and activate the defense. In this experiment, we evaluate the false positives that may be introduced by our defense. In our context, a false positive occurs when a non-vehicle object (originally not detected by the perception model) is incorrectly identified as a vehicle due to the defense mechanism. Specifically, we use the CARLA simulator [14] to model two representative driving scenarios: a T-junction road and a curved road, where common roadside structures may trigger false detections (as illustrated in Figure 6). CARLA is a widely used open-source simulation platform for developing and testing autonomous driving algorithms. It offers a high fidelity simulation environment with realistic road layouts and dynamic traffic scenarios. For each scenario in Figure 6, we examine three distinct street scenes with different surrounding layouts and evaluate four types of roadside objects: advertising billboard, bus shelter, traffic drum, and traffic barricade (as shown in Figure 7). The AV's speed is set to 25 km/h, and the initial distance between the object and the AV is approximately 35 meters. We employ PIXOR as the detection model, and the generative model used is the same as that described in Section 5.2. Table 6 reports the average false positive rate for each object type, defined as the percentage of LiDAR frames in which the perception system based on our defense incorrectly detects a non-existent front vehicle. The results in Table 6 indicate that the defense successfully avoids misidentifying benign objects in most LiDAR frames. While some objects occasionally trigger false positives, the overall rate remains low. Notably, such objects may also be misidentified as vehicles even without the defense, depending on the LiDAR object detection model, likely due to their LiDAR data resembling those of vehicles when viewed from certain angles.

Runtime on Resource-Constrained Devices. We also evaluate the real-time performance of our defense on a resource-constrained edge computing platform, specifically the NVIDIA Jetson AGX Orin. In this experiment, we consider two LiDAR object detection models: PIXOR and PointPillars. For each model, we evaluate the defense against both the AdvLoc and BALiDAR attacks, using the generative model trained in Section 5.2. The results show that the average runtime (RT) for the complete detection pipeline, which includes both our defense mechanism and the object detection model, is 0.12 seconds per frame with PIXOR and 0.09 seconds

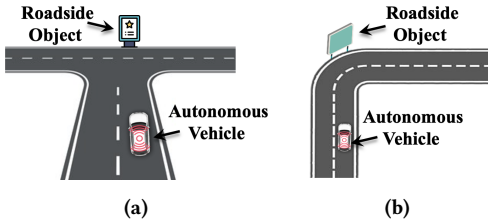


Figure 6: Two driving scenarios where common roadside objects may trigger false detections. (a) T-junction; (b) Curved road.

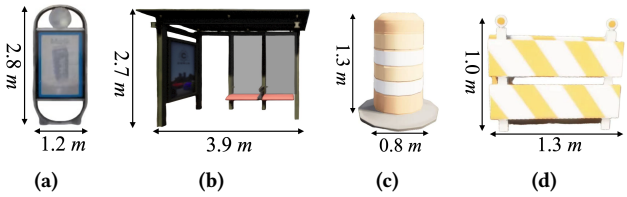


Figure 7: Roadside objects used in the false positive evaluation. (a) Advertising billboard; (b) Bus shelter; (c) Traffic drum; (d) Traffic barricade.

Table 6: False positive rate generated by the defense.

	Advertising billboard	Bus shelter	Traffic drum	Traffic barricade
False positive rate (%)	1.3	2.1	0.0	1.1

with PointPillars. These results demonstrate that the proposed defense can meet real-time constraints, even on resource-constrained edge computing platforms.

In addition to the above experiments, we conduct a case study to evaluate the impact of the proposed defense on vehicle behavior. Specifically, we integrate the defense mechanism into a full autonomous driving stack within the CARLA simulator. The defense module continuously monitors incoming LiDAR data, removes adversarial points, and forwards the cleaned point cloud to the perception module. The resulting perception outputs are then passed to the downstream planning and control modules. In this case study, we simulate an AdvLoc attack using PIXOR as the detection model. The results show that the attack is detected with low latency, enabling the AV to respond promptly, such as by initiating a lane change, to maintain safe navigation.

6 Real-World Evaluation

In this section, we assess the effectiveness of our proposed defense using a real-world LiDAR perception testbed, as shown in Figure 8. The setup features a Velodyne VLP-32C LiDAR sensor (widely used in commercial AVs) mounted 1.8 meters above the ground on the vehicle's roof. The sensor offers 32 channels and a 40° vertical field of view. For object detection, we adopt the PIXOR model, and evaluate the defense under two representative attack methods: AdvLoc and

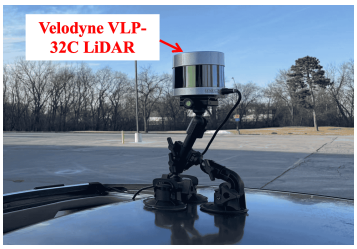


Figure 8: Real-world LiDAR perception testbed.

Table 7: Defense performance in different driving scenarios.

Attack	Scenario 1		Scenario 2		Scenario 3	
	DR(%)	RT(s)	DR(%)	RT(s)	DR(%)	RT(s)
AdvLoc	93.3	0.048	90.0	0.044	86.7	0.045
BALiDAR	90.0	0.051	86.7	0.046	90.0	0.045

BALiDAR, which have been demonstrated effective in physical-world environments. In both attack scenarios, a sedan is parked in front of the testbed to act as the target vehicle. For the AdvLoc attack, common cardboard pieces are deployed at specific positions around the target vehicle to serve as adversarial objects. In the BALiDAR setting, we follow the setup described in the original study by placing a 0.4-meter-radius exercise ball on the vehicle to act as the backdoor trigger.

Impact of the Driving Scenario. To assess the robustness of our defense across diverse driving environments, we conduct evaluations in three real-world scenarios, as illustrated in Figure 9. In this evaluation, we apply the generative model and threshold $\mathcal{T}$ learned in Section 5.2 using the KITTI dataset. It is worth noting that the KITTI dataset does not include LiDAR point clouds of the specific sedan used as the target vehicle in this experiment. In each scenario, the sedan is parked along the roadway while the LiDAR-equipped victim vehicle approaches it. The placement of adversarial objects is configured according to the respective attack strategy being tested. For evaluation, we randomly select 30 successfully attacked LiDAR frames per attack scenario, with the distance between the two vehicles ranging from 10 to 40 meters. Table 7 reports the average detection rates and runtime performance for both the AdvLoc and BALiDAR attacks across the tested scenarios. Results show that our defense consistently achieves high detection rates across different physical-world conditions and adversarial configurations. Furthermore, the runtime remains low, demonstrating that our system maintains real-time performance comparable to that achieved in simulation environments.

Impact of the Target Vehicle Type. To further evaluate the effectiveness of our defense in real-world scenarios, we examine its performance across different types of target vehicles. Importantly, our defense mechanism is designed to be applicable to any vehicle type, provided the vehicle can be detected by the underlying LiDAR object detection model (e.g., PIXOR or PointPillars). Since the defense is typically deployed by the same manufacturer responsible for training the detection model, both models can be trained using

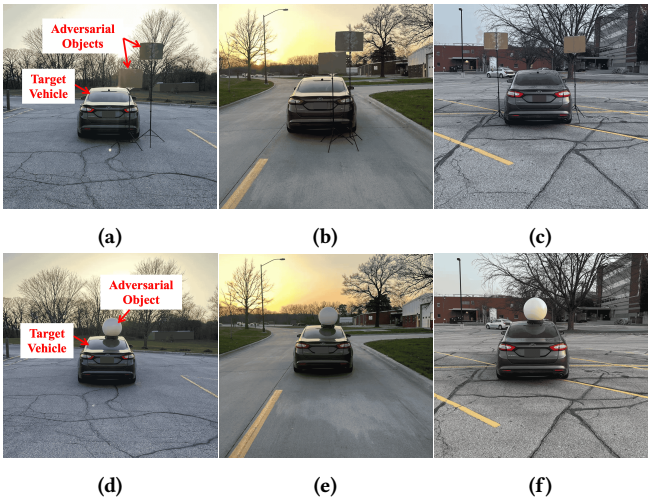


Figure 9: Different real-world scenarios for the AdvLoc and BALiADR attacks. (a) and (d) show scenario 1; (b) and (e) show scenario 2; (c) and (f) show scenario 3.

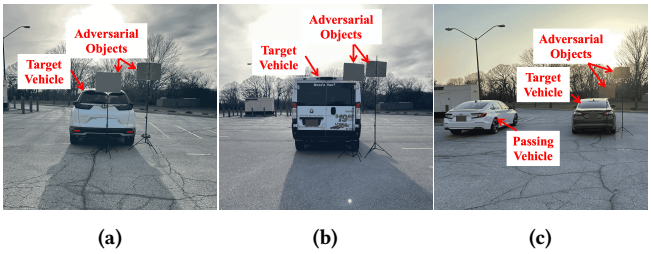


Figure 10: Real-world scenarios with different evaluation settings. (a) Target vehicle is an SUV; (b) Target vehicle is a van; (c) Scenario includes a passing vehicle near the target.

the same dataset. To validate this generalizability, we consider two additional target vehicle types: a sport utility vehicle (SUV) and a van. The real-world setups for these experiments are shown in Figure 10a and Figure 10b, respectively. Here we consider the AdvLoc attack. For training, the generative model incorporates data from both the KITTI dataset and supplementary data of the SUV and van, collected in real-world environments that are different from those in Figure 10. For each vehicle type, the target is parked on the road while the LiDAR-equipped victim AV approaches. We randomly select 30 successfully attacked LiDAR frames within a distance range of 10 to 40 meters for evaluation. The defense achieves average detection rates of 93.3% for the SUV and 80.0% for the van, with corresponding runtimes of 0.046 and 0.047 seconds, thereby demonstrating strong performance across varied vehicle types with real-time efficiency.

Impact of the Passing Vehicle. We next examine how the presence of additional road objects affects the performance of our defense. In this experiment, we consider a scenario involving a passing vehicle, as illustrated in Figure 10c. The same generative model used in the evaluation for Table 7 is employed here. We evaluate

Table 8: Detection rate (%) when the target is in motion.

Speed	Scenario 1	Scenario 2	Scenario 3
25 km/h	91.1	90.0	87.5
40 km/h	88.9	87.8	90.0

the defense against the AdvLoc attack by randomly selecting 30 successfully attacked LiDAR frames in which the distance between the target vehicle and the victim AV ranges from 10 to 40 meters. Despite the added complexity from the passing vehicle, our defense achieves a detection rate of 90.0% with an average runtime of 0.442 seconds, demonstrating its robustness in dynamic real-world environments.

Performance with Moving Vehicle. We also evaluate the performance of our defense when both the victim AV and the target vehicle are in motion. Specifically, we consider the BALiDAR attack and examine three driving scenarios similar to those shown in Figure 9. For each scenario, we evaluate two cases in which the target vehicle moves at approximately 25 km/h and 40 km/h, respectively, while the victim AV travels slightly faster in each case. We use PIXOR as the detection model and apply the generative model trained in Section 5.2. For each case, we repeatedly select 30 successfully attacked LiDAR frames and compute the average detection rate. Table 8 presents the average detection rate for each case. The results indicate that target vehicle motion has little impact on the defense performance, primarily because each LiDAR frame is processed independently by the defense mechanism.

7 Discussion

Other Attack Types. Although this paper primarily focuses on vehicle hiding attacks, the proposed defense mechanism has the potential to be adapted to other types of attacks by training the generative model to estimate corresponding object surfaces. To demonstrate its generality and adaptability, we consider the attack proposed in [39], which can cause a LiDAR classification model to misclassify pedestrians by placing adversarial objects around them. To adapt the generative model in our defense mechanism to this setting, we construct a set of training samples by extracting pedestrian point clusters from the KITTI dataset and injecting adversarial objects generated using the method in [39] into these clean samples. These examples enable the generative model to learn how to estimate pedestrian surfaces. Experimental results show that our defense achieves an 88.9% success rate in this setting, highlighting the generality and adaptability of the proposed approach.

Locations of Adversarial Objects. In our experiments, we implement existing object-based LiDAR attacks by following the settings described in their original papers. The derived location of the adversarial object is typically around or on the target vehicle, but not necessarily between the victim AV and the target. However, placing an adversarial object between the two vehicles can obstruct laser signals, resulting in missing points in the LiDAR returns for the target vehicle and posing a more challenging scenario for defense. To evaluate the defense performance under this condition, we consider the AdvLoc attack, which derives adversarial object locations through an optimization process. While the original formulation does not explicitly place objects between the victim AV and the target, we modify the optimization by introducing a constraint that limits the search space to the region between the two vehicles. For this evaluation, we use the generative model trained under the setting described in Section 5.2 and employ PIXOR as the detection model. We randomly select 60 LiDAR frames where the distance between the two vehicles ranges from 10 to 50 meters. The results show that our defense achieves an 86.7% detection rate in this scenario, demonstrating its robustness even when the target vehicle is partially occluded by the adversarial object.

8 Conclusion

This paper presents the first real-time defense mechanism against object-based LiDAR attacks in autonomous driving. The proposed mechanism is both detection model-agnostic and attack-agnostic. By introducing a lightweight generative model with a dual-decoder architecture and integrating it into a trigger-based defense pipeline, our approach enables reliable vehicle surface estimation and effective adversarial object removal with minimal runtime overhead. Unlike prior defenses that suffer from high latency or limited generalizability, our solution operates efficiently and remains robust across a wide range of attack strategies and detection models. Extensive evaluations in both simulated and real-world environments demonstrate that our defense achieves high detection accuracy across multiple attack methods, while maintaining low false positive rates and minimal latency.

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 *Proceedings of the 2021 IEEE International Conference on Image Processing (ICIP)*. 3592–3596.

[2] Simegnew Yihunie Alaba and John E Ball. 2022. A survey on deep-learning-based LiDAR 3D object detection for autonomous driving. *Sensors* 22, 24 (2022), 9577.

[3] Claudine Badue, Rânik Guidolini, Raphael Vivacqua Carneiro, Pedro Azevedo, Vinicius B Cardoso, Avelino Forechi, Luan Jesus, Rodrigo Berriel, Thiago M Paixao, Filipe Mutz, et al. 2021. Self-driving cars: A survey. *Expert Systems with Applications* 165 (2021), 113816.

[4] Harry G. Barrow, Jay M. Tenenbaum, Amy R. Hanson, and Elaine H. Crowley. 1977. Parametric correspondence and Chamfer matching: Two new techniques for image matching. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence (IJCAI)*. 659–663.

[5] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. 2020. nuScenes: A Multimodal Dataset for Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 11621–11631.

[6] 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 (AAAI)*, Vol. 38. 20949–20957.

[7] 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 *Proceedings of the 32nd USENIX Security Symposium (USENIX Security)*. 2993–3010.

[8] 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 *Proceedings of the 2021 IEEE Symposium on Security and Privacy (SP)*. 176–194.

[9] 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 (CCS)*. 2267–2281.

[10] 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).

[11] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. 2017. Multi-view 3D object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1907–1915.

[12] 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 *Proceedings of the 21st USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. 1831–1845.

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

[14] 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 (CoRL)*. 1–16.

[15] Haoqiang Fan, Hao Su, and Leonidas J Guibas. 2017. A point set generation network for 3D object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 605–613.

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

[17] 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 *Proceedings of the 31st USENIX Security Symposium (USENIX Security)*. 1903–1920.

[18] 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).

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

[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 (MobiCom)*. 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 (SenSys)*. 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 (MobiCom)*. 573–586.

[23] Zizhi Jin, Xiaoyu Ji, Yushi Cheng, Bo Yang, Chen Yan, and Wenyuan Xu. 2023. PLA-LiDAR: Physical laser attacks against LiDAR-based 3D object detection in autonomous vehicle. In *Proceedings of the 2023 IEEE Symposium on Security and Privacy (SP)*. 1822–1839.

[24] Zizhi Jin, Xuancun Lu, Bo Yang, Yushi Cheng, Chen Yan, Xiaoyu Ji, and Wenyuan Xu. 2024. Unity is Strength? Benchmarking the Robustness of Fusion-based 3D Object Detection against Physical Sensor Attack. In *Proceedings of the ACM Web Conference 2024 (The Web Conference)*. 3031–3042.

[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 (CVPR)*. 12697–12705.

[26] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. 2022. BEVFusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. *arXiv preprint arXiv:2205.13542* (2022).

[27] 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 *Proceedings of the 26th European Conference on Artificial Intelligence (ECAI)*. 1593–1600.

[28] 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 *Proceedings of the 33rd USENIX Security Symposium (USENIX Security)*.

[29] Mark Pauly, Richard Keiser, Leif P Kobbelt, and Markus Gross. 2003. Shape modeling with point-sampled geometry. In *Proceedings of the Premier Conference and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH)*. 641–650.

[30] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. 2017. PointNet++: Deep hierarchical feature learning on point sets in a metric space. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)* 30 (2017).

[31] Raul Quinonez, Jairo Giraldo, Luis Salazar, Erick Bauman, Alvaro Cardenas, and Zhiqiang Lin. 2020. SAVIOR: Securing autonomous vehicles with robust physical invariants. In *Proceedings of the 29th USENIX Security Symposium (USENIX Security)*. 895–912.

[32] 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 (CCS)*. 3459–3461.

[33] 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 (MobiCom)*. 133–146.

[34] 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 (MobiCom)*. 139–154.

[35] Hocheol Shin, Dohyun Kim, Yujin Kwon, and Yongdae Kim. 2017. Illusion and dazzle: Adversarial optical channel exploits against LiDARs for automotive applications. In *Proceedings of the 19th International Conference on Cryptographic Hardware and Embedded Systems (CHES)*. 445–467.

[36] 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 *Proceedings of the 29th USENIX Security Symposium (USENIX Security)*. 877–894.

[37] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. 2020. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2446–2454.

[38] 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 (CVPR)*. 13716–13725.

[39] Zhen Xiang, David J Miller, Siheng Chen, Xi Li, and George Kesidis. 2021. A backdoor attack against 3D point cloud classifiers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 7597–7607.

[40] 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).

[41] Yan Yan, Yuxing Mao, and Bo Li. 2018. SECOND: Sparsely Embedded Convolutional Detection. *Sensors* 18, 10 (2018), 3337.

[42] 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 (CVPR)*. 7652–7660.

[43] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. 2021. Center-based 3D object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 11784–11793.

[44] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. 2018. PCN: Point completion network. In *Proceedings of the 2018 International Conference on 3D Vision (3DV)*. 728–737.

[45] Yan Zhang, Zihao Liu, Chongliu Jia, Yi Zhu, and Chenglin Miao. 2024. An Online Defense against Object-based LiDAR Attacks in Autonomous Driving. In *Proceedings of the 22nd ACM Conference on Embedded Networked Sensor Systems (SenSys)*. 380–393.

[46] 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 (SenSys)*. 533–547.

[47] Yin Zhou and Oncel Tuzel. 2018. VoxelNet: End-to-end learning for point cloud based 3D object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4490–4499.

[48] 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 *Proceedings of the 33rd USENIX Security Symposium (USENIX Security)*.

[49] 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 (SenSys)*. 329–342.

[50] Yi Zhu, Chenglin Miao, Hongfei Xue, Yunnan Yu, Lu Su, and Chunming Qiao. 2024. Malicious attacks against multi-sensor fusion in autonomous driving. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking (MobiCom)*. 436–451.

[51] 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 (CCS)*. 1945–1960.