Towards Robust LiDAR-based Perception in Autonomous Driving: General Black-box Adversarial Sensor Attack and Countermeasures

Jiachen Sun
University of Michigan

Yulong Cao
University of Michigan

Qi Alfred Chen
UC Irvine

Z. Morley Mao
University of Michigan

Abstract

Perception plays a pivotal role in autonomous driving systems, which utilizes onboard sensors like cameras and LiDARs (Light Detection and Ranging) to assess surroundings. Recent studies have demonstrated that LiDAR-based perception is vulnerable to spoofing attacks, in which adversaries spoof a fake vehicle in front of a victim self-driving car by strategically transmitting laser signals to the victim's LiDAR sensor. However, existing attacks suffer from effectiveness and generality limitations. In this work, we perform the first study to explore the general vulnerability of current LiDAR-based perception architectures and discover that the ignored occlusion patterns in LiDAR point clouds make self-driving cars vulnerable to spoofing attacks. We construct the first black-box spoofing attack based on our identified vulnerability, which universally achieves around 80% mean success rates on all target models. We perform the first defense study, proposing CARLO to mitigate LiDAR spoofing attacks. CARLO detects spoofed data by treating ignored occlusion patterns as invariant physical features, which reduces the mean attack success rate to 5.5%. Meanwhile, we take the first step towards exploring a general architecture for robust LiDAR-based perception, and propose SVF that embeds the neglected physical features into end-to-end learning. SVF further reduces the mean attack success rate to around 2.3%.

1 Introduction

Today, self-driving cars, or autonomous vehicles (AV), are undergoing rapid development, and some are already operating on public roads, *e.g.,* self-driving taxis from Google's Waymo One [3] and Baidu's Apollo Go [1], and self-driving trucks from TuSimple [2] used by UPS. To enable self-driving, AVs rely on autonomous driving (AD) software, in which *perception* is a fundamental pillar that detects surrounding obstacles using sensors like cameras and LiDARs (Light Detection and Ranging). Since perception directly impacts safety-critical driving decisions such as collision avoidance, it is imperative to ensure its security under potential attacks.

In AD perception, 3D object detection is indispensable for ensuring safe and correct autonomous driving. To achieve this, almost all AV makers [4, 5, 7] adopt LiDAR sensors, since they capture high-resolution 360° 3D information called point clouds and are more reliable in challenging weather and lighting conditions than other sensors such as cameras. Due to such heavy reliance on LiDAR, a few prior studies have explored the security of LiDAR and its usage in AD systems [17, 48, 55]. Among them, Cao *et al.* are the first to discover that the deep learning model for LiDAR-based perception used in a real-world AD system can be fooled to detect a fake vehicle by strategically injecting a small number of spoofed LiDAR points [17]. Such LiDAR spoofing attacks could lead to severe safety consequences (*e.g.,* emergency brake operations that may injure passengers). However, the attack proposed was evaluated on only one specific model (*i.e.,* Baidu Apollo 2.5), assuming white-box access, which may be unrealistic. Moreover, it is unclear 1) whether the attack generalizes to other machine learning models, and 2) how to mitigate such spoofing attacks.

In this work, we perform the first study to systematically explore, discover, and defend against a *general* vulnerability existing among three state-of-the-art LiDAR-based 3D object detection model designs: bird's-eye view-based, voxel-based, and point-wise (introduced in §2). More specifically, we first demonstrate that existing LiDAR spoofing attacks [17, 55] cannot directly generalize to all three model designs (§4). Meanwhile, we find that in these prior works the required sensor attack capabilities to succeed in fooling AD perception are quite intriguing: Cao *et al.* [17] found that an attack trace with merely *60 points* is sufficient to spoof a front-near vehicle in Apollo 2.5, while a valid one should have *∼2000 points* [31], which is almost two magnitudes more. Thus, there must exist certain LiDAR-related physical invariants that are not correctly learned in the model, which could also be generalizable to other state-of-the-art 3D object detection model designs.

To explore the cause, we perform experiments based on hypotheses formed by empirical observations of deep learning models and unique physical features of LiDAR, and discover that *all* the three state-of-the-art 3D object detection model designs above generally ignore the *occlusion patterns* in LiDAR point clouds, a set of physical invariants for LiDAR (§5.2).

For example, when a vehicle is behind another vehicle, its point cloud can legitimately have much fewer points due to the front vehicle's occlusion of the LiDAR beams. *However, such point cloud with much fewer points should not be detected as a vehicle at front-near locations with no occlusions, due to the physical law.* Unfortunately, all three model designs today fail to differentiate these two cases. This allows an adversary to spoof almost two magnitudes fewer points into the victim's LiDAR but can still fool the perception model into detecting a fake front-near vehicle (§5.3).

Based on this general vulnerability, we design the first black-box (i.e., without any knowledge about the models) adversarial sensor attack on LiDAR-based perception models to spoof a front-near vehicle to a victim AV that can alter its driving decisions (§6). To realize this, we enumerate different occlusion patterns of a 3D vehicle mesh (*e.g.,* different occluded postures) to fit the sensor attack capability, and leverage ray-casting techniques [18] to render the attack traces. We perform large-scale experiments on the three target model designs with around 15,000 point cloud samples from the KITTI [31] dataset. Evaluations show that with the same sensor attack capability as prior works [17] (i.e., 60 spoofed points), adversaries can generally achieve over 80% success rates on all three model designs.

Since these spoofed point clouds directly violate the physical laws of the LiDAR occlusion patterns mentioned above, we then leverage them as physical invariants to defend against this class of LiDAR spoofing attacks. First, we design a model-agnostic defense solution, CARLO: oCclusion-Aware hieRarchy anomaLy detectiOn, which can be applied to LiDAR-based perception immediately without changing the existing models. CARLO exploits two occlusion-related characteristics: 1) the free space inside a detected bounding box, and 2) the locations of points inside the frustum corresponding to a detected bounding box. Large-scale evaluations of CARLO show that it can efficiently and effectively defend both white- and black-box LiDAR spoofing attacks [17]. CARLO is also found to have high resilience to adaptive attacks since it exploits physical invariants that are highly difficult, if not impossible, for attackers to break.

While the model-agnostic defense is already useful, it is also beneficial if we can improve the robustness of the model designs themselves. Thus, we further design a general architecture for robust LiDAR-based perception in AVs. We observe that LiDAR measures range data by nature; hence the front view (FV) of the LiDAR sensor preserves the physical features as well as the occlusion information [38]. Recent studies present view fusion-based models that combines the FV and 3D representations [24, 35, 72]. However, our experiment results show that current designs are still vulnerable to LiDAR spoofing attacks since features from the 3D representation dominate the fusion process. To address such limitations, we propose sequential view fusion (SVF), a novel view fusion-based model design that sequentially fuses the FV and 3D representations to ensure that the end-to-end learning makes sufficient use of the features from FV (§8.2). Evaluations show that SVF can effectively reduce the attack success rate to 2.3% without sacrificing the original performance, which is a 2.2× improvement compared to CARLO. We find that SVF is also resilient to white-box attacks and adaptive attacks.

Overall, our key contributions are summarized as follows:

- We perform the first study to explore the general vulnerability of current LiDAR-based perception architectures. We discover that current LiDAR-based perception models do not learn occlusion information in the LiDAR point clouds, which enables a class of spoofing attacks. We construct the first black-box spoofing attack based on this vulnerability. Large-scale evaluations show that attackers can achieve around 80% mean success rates on all target models.
- To defend against LiDAR spoofing attacks, we design a model-agnostic defense CARLO that leverages the ignored occlusion patterns as invariant physical features to detect spoofed fake vehicles. We also perform large-scale evaluations on CARLO, and demonstrate that CARLO can effectively reduce the mean attack success rate to 5.5% on all target models without sacrificing the original performance.
- We design a general architecture for robust LiDAR-based perception in AVs by embedding the front view (FV) representation of LiDAR point clouds. We find that existing view fusion-based models are dominated by features from the 3D representation, meaning they are still vulnerable to LiDAR spoofing attacks. To address such limitations, we propose sequential view fusion (SVF). SVF leverages a semantic segmentation module to better utilize FV features. Evaluations show that SVF can further reduce the mean attack success rate to 2.3%.

2 Background

2.1 LiDAR-based Perception in AVs

LiDAR-based perception leverages 3D object detection models to understand driving environments, in which the models output 3D bounding boxes for detected objects. Deep learning has achieved great success in computer vision tasks for 2D images. However, standard convolutional pipelines cannot digest point clouds due to their sparsity and irregularity. To this end, significant research efforts have been made for 3D object detection recently [36, 53, 54, 73], among which the state-of-the-art models can be grouped into three classes:

1. Bird's-eye view (BEV)-based 3D object detection. Due to the remarkable progress made in 2D image recognition tasks, a large number of existing works [7, 40, 43, 68] attempt to transform LiDAR point clouds into the 2D structure for 3D object detection in AD systems. Most state-of-the-art methods [7, 40, 68] conduct the transformation by projecting point clouds into the top-down view, also known as the BEV,

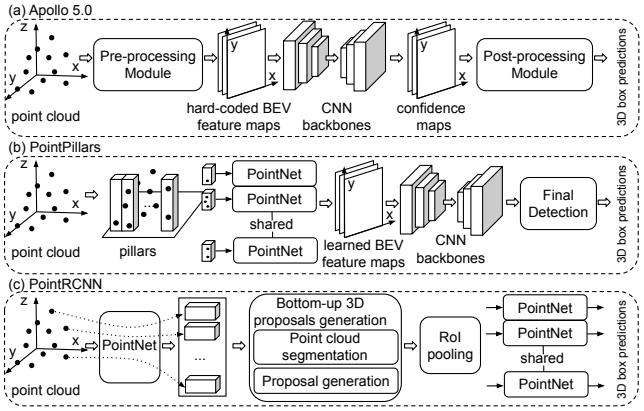


Figure 1: State-of-the-art LiDAR-based perception models.

and utilize convolutional neural networks (CNNs) to perform the final detection. Figure 1 (a) shows the architecture of *Apollo 5.0*[1], an industry-level BEV-based model, that has six hard-coded feature maps in the BEV and follows a UNet-like [52] pipeline to output the grid-level confidence score. The final stage heuristically clusters the grids that belong to the same object.

2. Voxel-based 3D object detection. VoxelNet [73] is the first model that slices the point clouds into voxels and extracts learnable features by applying a PointNet [49] to each voxel, after which a 2D convolutional detection layer is applied in the final stage. Many recent works [36, 37, 58, 67] adopt this voxel-based architecture and achieve state-of-the-art performance [9]. Figure 1 (b) shows the architecture of *PointPillars* that creatively voxelizes the point cloud into pillars (a representation of point clouds organized in vertical columns) to enhance the efficiency and follows the general design of voxel-based detection architectures. Notably, PointPillars is adopted by Autoware [6], an industry-level AV platform.

3. Point-wise 3D object detection. Instead of transforming point clouds to regular 2D structures or voxels for feature extraction, recent studies propose to directly operate on point clouds for 3D object detection [25, 53, 54, 71] and achieve the state-of-the-art performance. Most existing works in this category use a classic two-stage architecture similar to Faster RCNN [51] in 2D object detection. The first stage is responsible for generating high-quality region proposals in the 3D space. Based on these proposals, the second stage regresses the bounding box parameters and classifies the detected objects. As shown in Figure 1 (c), *PointRCNN* adopts a bottom-up method that generates point-wise region proposals in the first stage and regresses these proposals in the later stage.

2.1.1 KITTI Benchmark

KITTI [31] is a popular dataset for benchmarking AD research, of which the point cloud data are by design divided into a **trainval set** containing 7481 samples and a **test set** containing 7518 samples. We follow the methodology by Chen *et al.* to split the trainval set to a **training set** (3712 samples) and a **validation set** (3769 samples) for better experimental studies [23]. KITTI evaluates 3D object detection performance by average precision (AP) using the PASCAL [27] criteria and requires a 3D bounding box overlap (IoU) over 70% for car detection. KITTI also defines objects into three difficulty classes: Easy, Moderate, and Hard [9]. The difficulties correspond to different occlusion and truncation levels. We train PointPillars and PointRCNN on the training set, and Table 1 shows their APs evaluated on the validation set. We utilize the publicly released Apollo 5.0 model since it has its own labeling, which is incompatible with KITTI. In this work, we target car detection on the KITTI benchmark as the APs of pedestrian and cyclist detection are not yet satisfactory. However, our methodology can be generalized to other categories.

2.2 LiDAR Sensor and Spoofing Attacks

LiDAR sensor. A LiDAR instrument measures the distance to surroundings by firing rapid laser pulses and obtaining the reflected light with a sensor. Since the speed of light is constant, the accurate distance measurements can be derived from the time difference between laser fires and returns. By firing laser pulses at many predetermined vertical and horizontal angles, a LiDAR generates a point cloud that can be used to make digital 3D representations of surroundings. Each point in a point cloud contains its *xyz-i* information, corresponding to its location and the intensity of the captured laser return.

2.2.1 Sensor-level LiDAR Spoofing Attack

In the context of sensors, a spoofing attack is the injection of a deceiving physical signal into a victim sensor [46]. Since they share the same physical channels, the victim sensor accepts the malicious signal, trusting it as legitimate. Prior works [48, 55] have shown that LiDAR is vulnerable to laser spoofing attacks. Specifically, Petit *et al.* showed the feasibility to relay LiDAR laser pulses from other locations to inject fake points into the point cloud [48]. Shin *et al.* further improved the attack to control fake points at different locations in the point cloud, even very close to the victim vehicles [55].

2.2.2 Adv-LiDAR: Model-level LiDAR Spoofing Attack

Besides directly spoofing fake points into LiDAR point clouds, a recent study proposes Adv-LiDAR that uses adversarial machine learning to not only spoof a set of fake points into the point cloud but also manage to deceive the LiDAR-based perception model [17]. The authors formulate the attack on Apollo 2.5[2] as an optimization problem:

$$\begin{aligned} \min \quad & \mathcal{L}(x \oplus t'; \mathcal{M}) \\ \text{s.t.} \quad & t' \in \{\Phi(T') \,|\, T' \in \mathcal{A}\} \;\; \& \;\; x = \Phi(X) \end{aligned} \tag{1}$$

where X is the pristine point cloud and x represents the hard-coded feature maps in Apollo (§2.1). $\Phi(\cdot)$ is the preprocessing function for crafting the feature maps. T' and t'

[1] In this paper, we use "Apollo 5.0" to denote the Baidu Apollo 5.0 model.

[2] Apollo 2.5 was the latest version when Adv-LiDAR [17] was published. In this work, we target Apollo 5.0, the latest version at the time of writing.

are the spoofed point cloud and its corresponding feature maps, respectively. $\mathcal{A}$ stands for the sensor attack capability, and $\oplus$ merges the pristine and adversarial feature maps.

The attack goal is to spoof a fake vehicle right in front of the victim AV that leads to safety issues, and the success condition is that the confidence score of the optimized spoofed points (T') exceeds the default threshold so that Apollo 2.5 ($\mathcal{M}$) will detect T' as a valid vehicle. The authors formulate the sensor attack capability ($\mathcal{A}$) for general LiDAR spoofing attacks and design a specific loss function ($\mathcal{L}$) and a merging function ($\oplus$) for Apollo 2.5 ($\mathcal{M}$). By strategically controlling the spoofed points, Adv-LiDAR achieves around 75% attack success rate towards Apollo 2.5 and is considered as the state-of-the-art LiDAR spoofing attack.

3 Threat Model

Sensor attack capability. We perform the sensor-level spoofing attack experiments towards a Velodyne VLP-16 PUCK LiDAR [32]. The attack setup is the same as Cao *et al.* [17], and the utilized devices are detailed in Appendix A.

We adopt the formulation in Adv-LiDAR [17] to describe the sensor attack capability ($\mathcal{A}$):

- *Number of spoofed points.* Compared to Adv-LiDAR, we fine-tune the comparator circuit that bridges the photodiode and delay components to calculate the time delay more accurately. Moreover, we also use a better COTS lens put in front of the attack laser to refract the laser beams to a slightly wider azimuth range. Based on these improvements, we can stably spoof at most 200 points. Thus, we assume that attackers can spoof at most 200 points in the pristine point cloud. Such a capability is constrained by the attack hardware devices.
- *Location of spoofed points.* Similar to Adv-LiDAR, we assume that attackers are able to modify the *distance*, *altitude*, and *azimuth* of a spoofed point to the victim LiDAR by changing the delay intervals of the attack devices. Especially, the *azimuth* of a spoofed point can be modified within a horizontal viewing angle of 10°.

Black-box model-level spoofing attack. We consider LiDAR spoofing attacks as our threat model, which has been shown as a practical attack vector for LiDAR sensors [48, 55]. We adopt the attack goal of Adv-LiDAR: to spoof a *front-near* vehicle located 5-8 meters in front of the victim AV. To perform the attack, adversaries can place an attack device at roadsides to shoot malicious laser pulses to AVs passing by, or launch attacks in another vehicle in front of the victim car (*e.g.,* on the adjacent lane) [17]. LiDAR spoofing attack has been demonstrated to cause severe safety consequences in *Sim-control*, an AV simulator provided by Baidu Apollo [7]. For example, spoofing a front-near vehicle to a high-speed AV will make it trigger a hard brake, which may injure the passengers. Adversaries can also launch a spoofing attack on an AV waiting for the traffic lights to freeze the local transportation system [17]. We assume that attackers can control the spoofed points within the observed sensor attack capability ($\mathcal{A}$). Note that attackers do not have access to the machine learning model nor the perception system. We deem such an attack model realistic since we adopt the demonstrated sensor attack settings by Shin *et al.* [55] and relax the white-box assumptions in Adv-LiDAR [17].

Defense against general spoofing attacks. We also consider defending such LiDAR spoofing attacks. We assume a stronger attack model that adversaries have white-box access to the machine learning model and the perception systems. We also assume that defenders can only strengthen the software-level design, but cannot modify the AV hardware (*e.g.,* sensors) due to cost concerns. We deem it a realistic setting since we propose to defend state-of-the-art spoofing attacks, and software-level countermeasures can be easily adopted in current AD systems.

4 Limitations of Existing Attacks

In this section, we first study whether existing LiDAR spoofing attacks can realize the attack goal on three target models, and further discuss their limitations accordingly.

Limitations of sensor-level LiDAR spoofing attacks:

1. Blind attack limitation. The sensor-level spoofing attack suffers from the effectiveness issue due to no control strategies for the spoofed points. We apply the reproduced sensor attack traces to three target models and further explore whether they will be detected as vehicles at target locations. The results (detailed in §6.1.1) show that blindly spoofing cannot effectively achieve the attack goal other than Apollo 5.0, which also confirms the findings by Cao *et al.* [17].

Limitations of Adv-LiDAR:

1. White-box attack limitation. Adv-LiDAR, the state-of-the-art spoofing attack by Cao *et al.*, demonstrates the feasibility of leveraging adversarial machine learning techniques to enhance its effectiveness [17]. However, it suffers from the white-box limitation. Adv-LiDAR assumes that attackers have access to the deep learning model parameters and its pre- and post-processing modules. However, very few AV companies publicly release their perception systems, making Adv-LiDAR challenging to launch in the real world.

2. Attack generality limitation. Adv-LiDAR cannot be easily generalized. First, as introduced in §2.2.2, Adv-LiDAR only targets Apollo 2.5 and utilizes a specific pre-processing function ($\Phi(\cdot)$) and merging function ($\oplus$) which are not applicable to other models. Constructing such functions is non-trivial since they need to be differentiable so that the optimization problem can be solved by gradient descent-based methods [19]. For example, the $\Phi(\cdot)$ and $\oplus$ correspond to the voxelization and stacking processes, respectively, in PointPillars. It is still unknown whether such processes can be properly approximated differentiablely. Second, adversarial examples generated by Adv-LiDAR cannot transfer between models. We construct 20 optimized attack traces using Adv-LiDAR that successfully fool Apollo 5.0, and apply them to

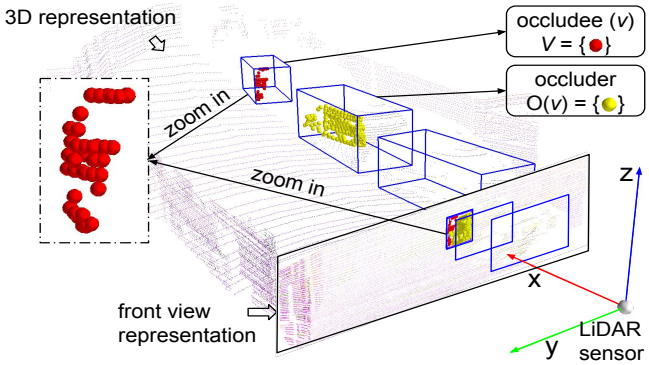


Figure 2: Illustration of an occluded vehicle (**C1**) in LiDAR point clouds. The yellow points from another vehicle occlude the vehicle v from the perspective of the LiDAR sensor. The blue 3D cubes are the bounding boxes of detected vehicles.

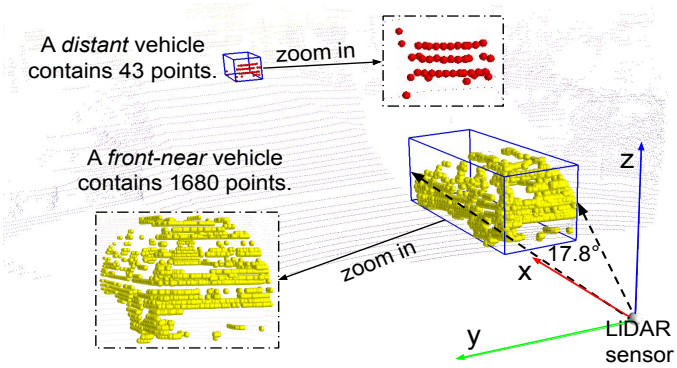


Figure 3: Illustration of a distant vehicle (**C2**) and a front-near vehicle in LiDAR point clouds, where the front-vehicle occupies 17.8° in azimuth from the perspective of the LiDAR sensor. The blue 3D cubes are the bounding boxes of detected vehicles.

the other two models. However, none can achieve the attack goal in either PointPillars or PointRCNN. Third, the attack trace T' is optimized with one specific point cloud at a time (Equation 1), which indicates that T' may not succeed in attacking other point cloud samples. The robustness analysis by Cao *et al.* also validates that the attack success rate consistently drops with the change of the pristine point cloud [17].

Overall, existing spoofing attacks cannot easily achieve the attack goal on all target models. Though Adv-LiDAR shows the feasibility to attack Apollo 2.5, more work is needed to understand the potential reasons that lead to its success.

5 A General Design-level Vulnerability

Motivated by the limitations of existing attacks, in this section, we leverage an in-depth understanding of the intrinsic physical nature of LiDAR to identify a general design-level vulnerability for LiDAR-based perception in AD systems.

5.1 Behind the Scenes of Adv-LiDAR

Despite a lack of generality, Adv-LiDAR was able to spoof a fake front-near vehicle by injecting much fewer points than required for a valid vehicle representation. For example, Cao *et al.* have demonstrated that an attack trace with merely 60 points and 8° of horizontal angles is sufficient to deceive Apollo 2.5 [17]. However, a valid front-near vehicle (§3) contains around 2000 points and occupies about 15° of horizontal angles in KITTI point clouds [31]. It remains unclear why such spoofing attacks can succeed despite a massive gap in the number of points between that of a fake and a valid vehicle. To answer this question and comprehend the general vulnerability exposed by Adv-LiDAR, it is necessary to consider the distinct physical features of LiDAR. In particular, we identify two situations where a valid vehicle contains a small number of points in LiDAR point clouds: 1) **an occluded vehicle** and 2) **a distant vehicle**, each corresponding to a unique characteristic (**C**) of LiDAR.

C1: Occlusions between objects will make occluded objects partially visible in the LiDAR point cloud. As introduced in §2.2, a LiDAR sensor functions by firing laser pulses and capturing their returns. As a result, each point in a point cloud represents the distance to the *nearest* solid object along the laser ray. Similar to human eyes, a LiDAR sensor can only perceive parts of an object (*e.g.,* a vehicle) if other obstacles, that obstruct the laser beams, are standing between the LiDAR and the object. Consequently, an occluded vehicle contains significantly fewer points than a fully exposed one since only a portion of it is visible.

In this paper, we name such occluded objects as *occludees* and the obstacles that occlude others as *occluders*. Particularly, as shown in Figure 2, we use $\mathsf{O}(v)$ to represent the point set that occludes a vehicle v, and V to denote the point set that belongs to the vehicle v in a point cloud F.

C2: The density of data decreases with increasing distance from the LiDAR sensor, due to the working principles of LiDAR sensors (§2.2). Since the generated point clouds are collected uniformly in vertical and horizontal angles, the density of point clouds varies in the 3D space. Similar to human eyes in which a far object occupies much fewer pixels than a near one with identical size, a distant vehicle contains significantly fewer points since its point set is much sparser than that of a front-near vehicle in LiDAR point clouds (Figure 3).

Based upon these observations, we propose two hypotheses of potential false positive (**FP**) conditions for current LiDAR-based perception models, which could contribute to the success of Adv-LiDAR:

FP1: *If an occluded vehicle can be detected in the pristine point cloud by the model, its **point set** will still be detected as a vehicle when directly moved to a front-near location.*

FP2: *If a distant vehicle can be detected in the pristine point cloud by the model, its **point set** will still be detected as a vehicle when directly moved to a front-near location.*

5.2 Experimental Validation

We design experiments (**E**) to test the existence of such potential erroneous predictions (*i.e.,* **FP**) on three target models using the KITTI validation set.

E1: To validate **FP1**, we first randomly pick 100 point

cloud samples $\mathcal{F} = \{F_i\}_{i=1}^{100}$ that contain 100 target *occluded* vehicles $\{v_i\}_{i=1}^{100}$ with their point sets $V_i \subseteq F_i$. We then feed $\mathcal{F}$ into three target models and record the confidence scores (*i.e.,* outputs of models to represent the confidence of detection) of the occluded vehicles as s_i for each v_i.

Second, we leverage a global translation matrix $H(\theta, \tau)$ (Equation 2) to move every V_i to a front-near location (*i.e.,* 5-8 meters in front of the victim AV) in the point cloud F_i as V_i', where θ and τ correspond to the *azimuth* and *distance* of the translation, respectively:

$$V'_{i\,\mathbf{w_i}} = V_{i\mathbf{w_i}}$$

$$\begin{bmatrix} V'_{i\,\mathbf{w_x}} \\ V'_{i\,\mathbf{w_y}} \\ V'_{i\,\mathbf{w_z}} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & \tau\cos(\theta+\alpha) \\ \sin\theta & \cos\theta & 0 & \tau\sin(\theta+\alpha) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} V_{i\mathbf{w_x}} \\ V_{i\mathbf{w_y}} \\ V_{i\mathbf{w_z}} \\ 1 \end{bmatrix} \tag{2}$$

$(V_{i\mathbf{w_x}}, V_{i\mathbf{w_y}}, V_{i\mathbf{w_z}}, V_{i\mathbf{w_i}})$ denotes the *xyz-i* feature vectors (introduced in §2.2) of all points in V_i, and $\alpha = arctan(V_{i\mathbf{w_y}}/V_{i\mathbf{w_x}})$. We make sure that there are no other objects standing between the LiDAR and each V_i'. By doing so, we construct a new set, where the points belong to the target occluded vehicles are moved to a front-near location by Equation 2. We further feed the new point cloud set $\mathcal{F}'$ into three target models and record the confidence scores of the translated points V_i' as s_i'.

Experimental results show that all of the translated points V_i' are detected by three target models, and we calculate the relative errors $e = \frac{|s_i' - s_i|}{s_i}$. Figure 4 shows the CDF of e for three target models. As shown, 99.5% of the picked occluded vehicles only have below 10% fluctuations of their confidence scores, which successfully validate **FP1**.

The success of **E1** comes from the fact that LiDAR-based 3D object detection models perform *amodal* perception, where given only the visible portions of a vehicle v, the model attempts to reason about occlusions and predict the bounding box for the complete vehicle (Figure 2). However, convolutional operations exploit spatial locality by enforcing a local connectivity pattern between neurons of adjacent layers. Such architecture thus ensures to produce the strongest response to a spatially local input pattern. Since the *occludee*'s and *occluder*'s point sets V and $O(v)$ stand apart from each other in the 3D space, deep learning models may fail to identify the causality between V and $O(v)$ and thus learns to regress the bounding box for v by V only.

E2: To validate **FP2**, similarly, we first randomly pick 100 point cloud samples that contain 100 target *distant* vehicles $\{v_i\}_{i=1}^{100}$ that locate farther than 30 meters away from the AV, and follow the same procedure with **E1** to record the confidence score changes. Experimental results show that all of the translated points V_i' are detected by three target models, and we calculate the relative errors $e' = \frac{|s_i' - s_i|}{s_i}$. Figure 4 shows the CDF of e' for three target models. As shown, 99.5% of the picked distant vehicles only have below 7.5% fluctuations

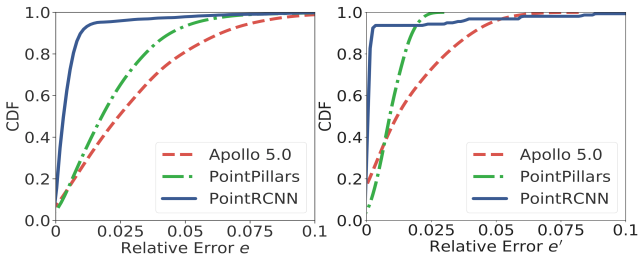


Figure 4: Left: CDF of the relative errors e in **E1**. Right: CDF of the relative errors e' in **E2**.

of their confidence scores, which successfully validate **FP2**.

The success of **E2** comes from that 3D object detection models are designed to be non-sensitive to the locations of objects. For example, Apollo 5.0 does not incorporate location information in its hard-coded feature maps, and PointRCNN regards the centers of each bounding box as the origins of their coordinates. Hence the global locations of objects are not valued by the 3D object detection models in AD systems.

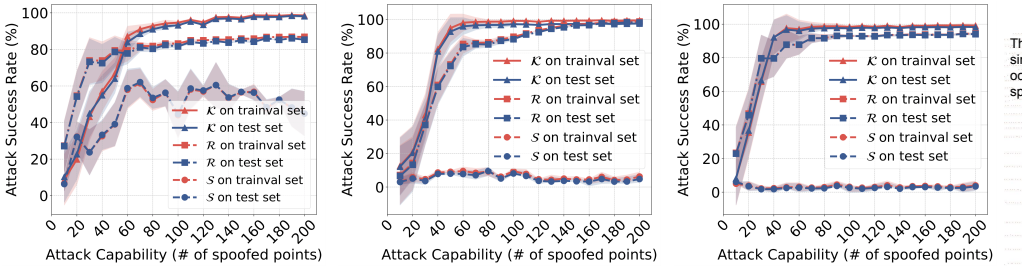
5.3 Vulnerability Identification

As mentioned earlier, the sensor attack capability $\mathcal{A}$ is far from spoofing a fully exposed front-near vehicle's point set. However, **E1** and **E2** provide two strategies for adversaries to launch spoofing attacks with fewer points and horizontal angles. As a result, attackers can directly spoof a vehicle imitating various occlusion and sparsity patterns that satisfy the sensor attack capability $\mathcal{A}$ to fool the state-of-the-art models. For example, the V (red points) in Figure 2 only contains 38 points and occupies 4.92° horizontally when translated to 6 meters in front of the AV. We confirm that it can deceive all three target models successfully, as visualized in Appendix E.

The vulnerability comes from the observation that the state-of-the-art 3D object detection architectures ignore the distinct physical features of LiDAR. Therefore, they leave a gap, as well as an attack surface, between the model capacity and LiDAR point clouds. We further abstract the neglected physical features as two occlusion patterns inside the LiDAR point clouds, described below.

Inter-occlusion. We abstract the typical occlusion introduced in §5.1 as inter-occlusion. As its name indicates, inter-occlusion describes a causal relationship between *occludee* and the corresponding *occluders* (*i.e.,* the *occluders* cause the *occludee* partially visible). **FP1** violates the physical law of inter-occlusion since a translated "occluded" vehicle's point set V' no longer has its valid *occluder* $O(v)$. However, **E1** demonstrates that state-of-the-art LiDAR-based perception models overlook such inter-occlusions in the point clouds.

Intra-occlusion. We abstract the other occlusion pattern hidden inside an object as intra-occlusion. The facing surface of a solid object (*e.g.,* a vehicle) occludes itself in the point cloud, which indicates that the LiDAR cannot perceive the interior of the object (Figure 9). **FP2** violates the physical law of intra-occlusion since the abnormal sparseness of a translated "distant" vehicle's point set V' can no longer fully



(a) *ASR* of Apollo 5.0. (b) *ASR* of PointPillars. (c) *ASR* of PointRCNN.

Figure 5: Attack success rates (*ASR*s) of proposed black-box spoofing attacks on target state-of-the-art models.

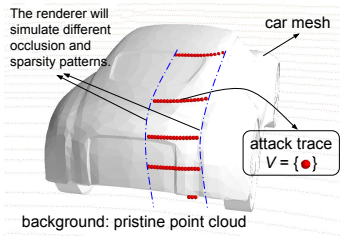


Figure 6: The process of generating attack traces for $\mathcal{R}$ from the implemented renderer.

occlude a valid vehicle since other laser pulses could penetrate its "surface". However, **E2** demonstrates that state-of-the-art LiDAR-based perception models are unable to differentiate reflected points of real solid objects from sparse injected points of the same overall shape so that they also overlook the intra-occlusions in the LiDAR point clouds.

To demonstrate the potential real-world impacts of this identified vulnerability, we construct the first black-box spoofing attack on state-of-the-art LiDAR-based perception models in §6. We find the violations of the physical law of occlusion generally enable LiDAR spoofing attacks. Therefore, we perform the first defense study, exploiting the occlusion patterns as physical invariants to detect spoofing attacks in §7. Lastly, in §8, we present a general architecture for robust LiDAR-based perception that embeds occlusion patterns as robust features into end-to-end learning.

6 Black-box Spoofing Attack

Constructing black-box attacks on deep learning models is non-trivial. Prior works have studied black-box attacks on image classification [45] and speech recognition models [8]. However, none explored LiDAR-based perception models, and their approaches usually suffer from efficiency limitations (*e.g.,* building a local substitute model). In this section, we present the first black-box LiDAR spoofing attack based on our identified vulnerability (§5.3) that achieves both high efficiency and success rates.

1. Constructing original attack traces. As demonstrated in §5.3, occluded or distant vehicles' point sets that meet the sensor attack capability can be utilized to spoof front-near vehicles. Therefore, our methodology attempts to closely represent realistic physical attacks using traces from real-world datasets (*e.g.,* KITTI). In order to test different sensor attack capability, we extract occluded vehicles' point sets with varying numbers of points (5-200 points) from the KITTI validation set. Furthermore, we take 10 points as interval, and divide the extracted point sets into 20 groups per their number of points (The first group contains traces with the number of points from 0 to 10, and the second group contains traces with the number of points from 10 to 20, *etc.*). We then randomly pick five traces in each group forming a small dataset $\mathcal{K}$ containing 100 point sets.

Besides collecting existing real-world traces, the identified vulnerability also supports adversaries in generating customized attack traces, which are more efficient for pipelining the attack process. We leverage ray-casting techniques to generate customized attack traces. More specifically, we utilize a 3D car mesh and implement a renderer [18] simulating the function of a LiDAR sensor that probes the car mesh by casting lasers. By doing so, we can render the car mesh's point cloud. We further simulate different occlusion and sparsity patterns on the car mesh to fit the sensor attack capability, as shown in Figure 6. Similar to $\mathcal{K}$, we collect rendered point clouds with different numbers of points by using different postures and occlusion patterns. We also follow the same procedure to build a small dataset $\mathcal{R}$ containing 100 rendered point sets. More figures of $\mathcal{R}$ are shown in Appendix E.

2. Spoofing original attack traces at target locations. To trigger severe security and safety consequences, adversaries need to inject the constructed attack traces at target locations in the point cloud. We consider spoofing $\mathcal{K}$ and $\mathcal{R}$ in both digital and physical environments. For digital spoofing, we make sure the injection of attack traces meets the sensor attack capability $\mathcal{A}$ and real-world requirements. We follow the high-level formulation in Adv-LiDAR [17] utilizing a global transformation matrix $H(\theta,\tau)$ (Equation 2) to translate the attack traces (*i.e.,* $V'^T = H(\theta,\tau)\cdot V^T$, where $V \in \mathcal{K}\cup\mathcal{R}$). Here the translation interprets the attack capability ($\mathcal{A}$) in terms of modifying the *azimuth* and *distance* of attack traces. We further calibrate each point in the translated attack trace to its nearest laser ray's direction and prune the translated attack trace to fit the attack capability (*i.e.,* $V' \in \mathcal{A}$). Finally, we merge the attack trace with the pristine point cloud according to the physics of LiDAR. We feed the modified point cloud samples containing the attack traces into three target models. For physical spoofing, we program attack traces from $\mathcal{R}$ as input to the function generator so that we can control the spoofed points and launch the spoofing attack [55] in our lab. We further collect the physical attack traces and feed them into target models. Due to the limitation of our attack devices, we only conduct preliminary physical spoofing experiments. More details of physical spoofing can be found in §9.1.2. It is worth noting that such limitations do not hurt the validity of

our attack model (§3) since the attack capability $\mathcal{A}$ is adopted from Adv-LiDAR [17], in which has been demonstrated in the real world.

6.1 Attack Evaluation and Analysis

We perform large-scale evaluations on our proposed black-box attack in terms of effectiveness and robustness.

Experimental setup. The evaluations are performed on the KITTI trainval and test sets (introduced in §2.1.1), which are collected in the physical world. As mentioned before, limited by our attack devices, we leverage $\mathcal{K}, \mathcal{R}$ to launch *digital spoofing* attacks. We also utilize attack traces ($\mathcal{S}$) generated by the sensor-level spoofing attack (§4) as a baseline. $\mathcal{S}$ is collected from blindly *physical spoofing* attacks on a real Velodyne VLP-16 PUCK LiDAR [32]. We further inject all the attack traces from above three constructed datasets into the KITTI point clouds at front-near locations (*i.e.,* 5-8 meters in front of the victim AV) to test their effectiveness.

Evaluation metrics. Object detection models often have default thresholds for confidence scores to filter out detected objects with low confidence (potential false positives). We leverage the default thresholds used by three target models to measure the attack success rate (*ASR*). We label an attack successful as long as the model detects a vehicle at the target location whose confidence score exceeds the default threshold:

$$ASR = \frac{\text{\# of successful attacks}}{\text{\# of total point cloud samples}} \tag{3}$$

Besides the default threshold, we also define a new metric that leverages multiple thresholds to evaluate LiDAR spoofing attacks. The corresponding definitions and evaluations are described in Appendix B, which provide insights that point-wise features appear to be more robust than voxel-based features.

6.1.1 Attack Effectiveness

Figure 5 shows the *ASR* of the digital spoofing attack with different attack capabilities (*i.e.,* number of points). As expected, the *ASR* increases with more spoofed points. The *ASR*s are able to universally achieve higher than 80% in all target models with more than 60 points spoofed, and it also stabilizes to around 85% with more than 80 points spoofed. Notably, the attack traces from $\mathcal{R}$ can achieve comparable *ASR* with $\mathcal{K}$ on all target models, which demonstrates that adversaries can efficiently leverage a customized renderer to generate attack traces (Figure 6). Such rendered traces can be directly programmed into hardware for physical spoofing attacks (Appendix A). Interestingly, $\mathcal{S}$ achieves much higher *ASR* on Apollo 5.0, indicating that BEV-based features are less robust to spoofing attacks than the other two categories, which could be attributed to the information loss of feature encoding from BEV.

6.1.2 Robustness Analysis

We analyze the robustness of the proposed attack to variations of attack traces V' and the average precision (AP) of target models $\mathcal{M}$. We also evaluate the attack robustness against state-of-the-art defense strategies [66, 70] designed for image-based adversarial attacks. We find that spoofed traces with around 60 points to trigger major changes in *ASR*. Note that Cao *et al.* also utilized spoofed traces with 60 points for analysis [17]. Therefore, we use attack traces with $(60, 70]$ points from $\mathcal{R}$ for the robustness analysis.

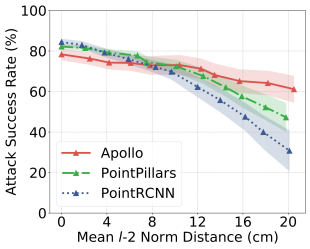


Figure 7: Attack robustness to variations in generated attack traces from $\mathcal{R}$.

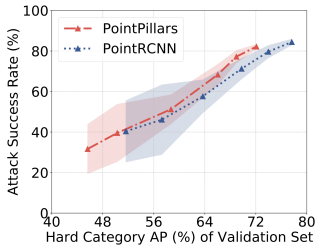


Figure 8: Attack robustness to variations in target models' performance.

Robustness to variations in attack traces. First, we apply a scaling matrix S to the attack traces V' with different-level randomness to simulate the inaccuracy of sensor attack:

$$V''_{\mathbf{w_i}} = V'_{\mathbf{w_i}}$$
$$\begin{bmatrix} V''_{\mathbf{w_x}} \\ V''_{\mathbf{w_y}} \\ V''_{\mathbf{w_z}} \end{bmatrix} = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix} \cdot \begin{bmatrix} V'_{\mathbf{w_x}} \\ V'_{\mathbf{w_y}} \\ V'_{\mathbf{w_z}} \end{bmatrix} \tag{4}$$

where s subjects to a uniform distribution $U(1-\varepsilon, 1+\varepsilon)$. We use the mean l-2 norm to measure the distance between V'' and V'. Figure 7 shows the *ASR* drops with larger l-2 distances which is expected. However, as shown, the *ASR* still reaches around 70% while the distance is around 10 cm. We also observe that the *ASR* for PointRCNN drops faster than for PointPillars and Apollo 5.0, which also validates that point-wise features are arguably more robust than voxel-based and BEV-based features (detailed in Appendix B).

Robustness to variations in model performance. To understand the relationship between *ASR* and the original performance of models (*i.e.,* AP), we first extract the intermediate models when we trained PointPillars and PointRCNN. We then try to launch attacks on these models. Surprisingly, we find that the *ASR* increases with higher AP[3] (Figure 8), which implies that a model with better performance could be more vulnerable to such attacks. Our results indirectly demonstrate that the identified vulnerability could be attributed to an ignored dimension (*i.e.,* occlusion patterns) by current models. Since the models do not notice such a hidden dimension, they will be overfitted to be more vulnerable during training.

Robustness to adversarial training. Adversarial training is not rigorously applicable because it targets classification models, and requires norm-bounded perturbations to make

[3]We evaluate the *ASR*s until the training procedures (*i.e.,* APs) converge on both models.

the optimization problem tractable [42]. In contrast, our study targets 3D object detection models, and the proposed attack is constrained by the sensor attack capability ($\mathcal{A}$), which does not fit any existing norm-bounded formulations. Thus, we perform this robust analysis in an empirical setting. Specifically, we generate another 100 attack traces with 60 points using the customized renderer and randomly inject two of them into each point cloud sample in the KITTI training set at areas without occlusions. We further train PointPillars and PointRCNN on this modified dataset and evaluate our proposed attack using the same 60-point attack traces with §6.1.1 on them. We observe that the *ASR*s drop from 83.6% to 70.1% and 88.3% to 79.7% on PointPillars and PointRCNN, respectively, on the KITTI validation set. However, the "Hard" category's original detection performance has significant degradation of over 10% on both models. Our results empirically show that current LiDAR-based perception model designs cannot learn the occlusion information correctly. The slight drop of the *ASR*s comes from the under-fitting effect of existing occluded vehicles (*i.e.,* significant AP degradation), which is not acceptable in real AD systems.

Robustness to randomization-based defenses. We leverage state-of-the-art image-based defenses: feature squeezing [66] and ME-Net [70] to test the attack robustness on Apollo 5.0 since it has similar pipelines with image-based models. We demonstrate that none of them can defend the black-box spoofing attack without hurting the original AP. More details can be found in Appendix B.

7 Physics-Informed Anomaly Detection

Our results show that a lack of awareness for occlusion patterns enables the proposed black-box attack in §6. Since adversaries exploit an ignored hidden dimension, such attacks can succeed universally in target models and appear to be robust to existing defenses (§6.1). Since anomaly detection methods have been widely adopted in different areas [20, 39], one intuitive and immediate mitigation is to detect such violations of physics. We find that no existing open-source AV platforms enable such physical checking [6, 7]. In this section, we present CARLO: oCclusion-Aware hieRarchy anomaLy detectiOn, that harnesses occlusion patterns as invariant physical features to accurately detect such spoofed fake vehicles.

7.1 CARLO Design

CARLO consists of two building blocks: free space detection and laser penetration detection.

7.1.1 Free Space Detection

Free space detection (FSD) integrates both inter- and intra-occlusions (§5.3) to detect spoofed fake vehicles. As introduced in §2.2, each laser in a LiDAR sensor is responsible for perceiving a direction in the spherical coordinates. Due to resolution limits, such a laser direction actually corresponds to a thin frustum in the 3D space. As shown in Figure 9, the

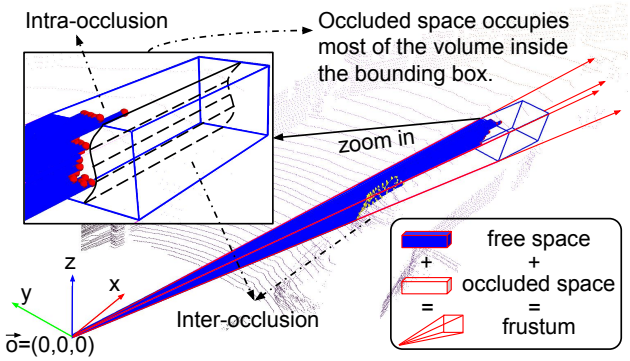


Figure 9: Illustration of free space (**FS**) and occluded space (**OS**) in a frustum corresponding to a detected bounding box.

frustum (as well as the straight-line path $\vec{p}-\vec{o}$) from the LiDAR sensor ($\vec{o}=(0,0,0)$) and any point in the point cloud ($\forall\vec{p}=(x,y,z)$) is considered as free space (drivable space occupied by air only). Therefore, combined with all laser beams of the LiDAR, the entire 3D space is divided into free space (**FS**) and occluded space (**OS**) (*i.e.,* space behind the hit point from the LiDAR sensor's perspective). **FS** information is embedded at the point level. Occlusions, on the other hand, exist at the object level. **FS**, thus, is more fine-grained and incorporates occlusion information since the **OS** of an object directly reflects its occlusion status (Figure 9).

Due to inter-occlusion and intra-occlusion, we observe that the ratio f of the volume of **FS** over the volume of a detected bounding box should be subject to some distribution and upper-bounded by $\exists b\in(0,1)$, implying $f\in(0,b]$ (Figure 9). Nevertheless, since the fake vehicles do not obey the two occlusion patterns, their ratio f' should be large enough and lower-bounded by $\exists a\in(0,1)$ such that $f'\in[a,1)$. Clearly, as long as $a>b$, we have opportunities to distinguish valid vehicles with the spoofed fake vehicles statistically. To estimate the ratio f, we grid the 3D space into cells and calculate:

$$f_{\mathsf{B}}=\frac{\sum_{c\in\mathsf{B}}\mathbb{1}\cdot FS(c)}{|\mathsf{B}|} \tag{5}$$

where $FS(c)$ indicates whether the cell c is free or not, and $|\mathsf{B}|$ denotes the total number of cells in the bounding box B. The algorithm to derive $FS(c)$ can be found in Appendix C.

We then estimate the distributions of valid and fake vehicles.We empirically set the cell size to 0.25^3 m^3, and utilize the KITTI training set and 600 new attack traces generated by the implemented renderer (§6) for estimation. Figure 10 shows that the CDF of f and f' clearly separate from each other. We further take the models' error into considerations (0.7 IoU), and estimate the distributions again. The two distributions still do not overlap with each other, as shown in Figure 10, which demonstrate the feasibility to leverage the ratio f as an invariant indicator for detecting anomalies.

However, though FSD provides a statistically significant method to detect adversarial examples, it is too time-consuming to perform ray-casting to all the detected bounding

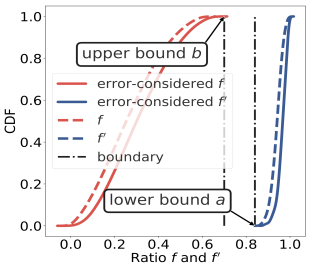


Figure 10: CDF of f and f', and the two distributions are clearly separate.

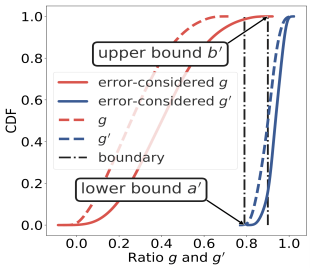


Figure 11: CDF of g and g', but the two distributions overlap with each other.

boxes in real-time. The mean processing time of one vehicle is around 100 ms in our implementation using C++ on a commodity Intel i7-6700K CPU @ 4.00GHz, which is already comparable to the inference time of deep learning models.

7.1.2 Laser Penetration Detection

Laser penetration detection (LPD) is a variant of FSD that aims to provide better efficiency for CARLO. As introduced in §7.1, each point in the point cloud represents one laser ray and the boundary between free space and occluded space. Given a vehicle's point set, its bounding box B also divides the corresponding frustum into three spaces which are: 1) the space between the LiDAR sensor and the bounding box $\mathsf{B}\uparrow$, 2) the space inside the bounding box B, and 3) the space behind the bounding box $\mathsf{B}\downarrow$. Intuitively, only a small number of laser rays can *penetrate* the bounding box (Figure 9). As a result, from the perspective of the LiDAR sensor, the ratio g of the number of points located in the space behind the bounding box $\mathsf{B}\downarrow$ over the total number of points in the whole frustum should be upper bounded by $\exists b' \in (0,1)$. For the same reason in §7.1, the ratio g' of the spoofed vehicles is supposed to be large enough and lower bounded by $\exists a' \in (0,1)$.

Therefore, the ratio g is derived from:

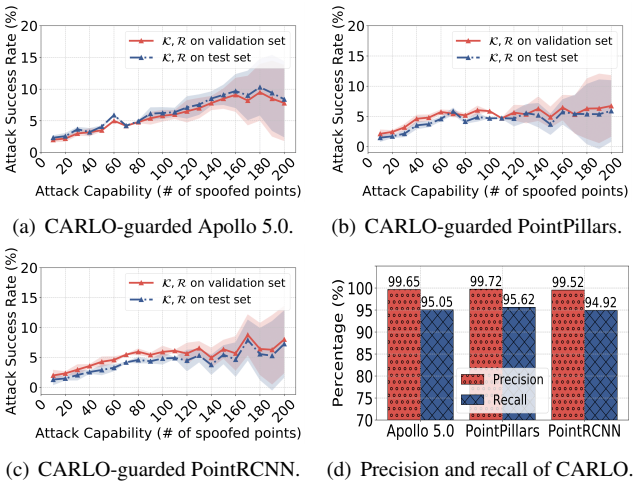
$$g_{\mathsf{B}} = \frac{\sum_{\vec{p}\in \mathsf{B}\downarrow} \mathbb{1}(\vec{p})}{\sum_{\vec{p}\in \mathsf{B}\cup\mathsf{B}\downarrow\cup\mathsf{B}\uparrow} \mathbb{1}(\vec{p})} \tag{6}$$

Since LPD leverages information directly from the output of models, it is a good fit for parallel acceleration. The mean processing time of LPD is around 5 ms for each bounding box using Python on a commodity GeForce RTX 2080 GPU.

Similarly, Figure 11 shows the CDF of g and g' for valid vehicles from the KITTI training set and the 600 generated attack traces, respectively. As shown, though the distributions of ground-truth are separate, the error-considered distributions overlap with each other (*i.e.,* $b' > a'$). We verify that the overlap comes from the noise introduced by points of the ground plane. As a result, LPD will cause erroneous detection of potential anomalies.

7.1.3 Hierarchy Design

To achieve both robustness and efficiency, CARLO hierarchically integrates FSD and LPD. In the first stage, CARLO



(a) CARLO-guarded Apollo 5.0. (b) CARLO-guarded PointPillars.

(c) CARLO-guarded PointRCNN. (d) Precision and recall of CARLO.

Figure 12: Attack success rates (*ASR*s) of proposed black-box spoofing attacks on three CARLO-guarded models.

accepts the detected bounding boxes and leverages LPD to filter the unquestionably fake and valid vehicles by two thresholds (§7.1.2). The rest bounding boxes are uncertain and will be further fed into FSD for final checking. CARLO achieves around 8.5 ms mean processing time for each vehicle. The entire algorithm of CARLO is detailed in Appendix C.

7.2 CARLO Evaluation

Experimental setup. We evaluate the defense performance of CARLO on the KITTI trainval and test sets. We apply all the attack traces from $\mathcal{K}, \mathcal{R}$ to all point cloud samples at target locations (5-8 meters in front of the victim), and feed them into three CARLO-guarded models $\mathsf{CARLO}(\mathcal{M}(\cdot))$. We also evaluate CARLO against Adv-LiDAR [17] on Apollo 5.0. The defense goal is to successfully detect the spoofed fake vehicles from the output bounding boxes without hurting the original performance (*i.e.,* AP) of the target models.

Evaluation metrics. We evaluate the performance of CARLO in two aspects, which are the *ASR* of the CARLO-guarded models, and the precision and recall of CARLO itself. *ASR* directly relates to the defense performance, while the precision and recall of CARLO reflect whether it will harm the original AP of target models. We test the *ASR* on the validation set and test set since the distributions are estimated from the training set. We only evaluate the precision and recall of CARLO on the validation set as we do not have the ground-truth for the test set.

Figure 12 (a-c) shows the *ASR* of three CARLO-guarded models. As shown, CARLO reduces the *ASR* from more than 95% to below 9.5% with the maximum attack capability, and reduce the mean *ASR* to around 5.5%. We observe that the remaining 5.5% comes from the detection errors (*i.e.,* the detected bounding box of the fake vehicle cannot match well with the ground-truth) that shift the f' and g' to the distribution of valid vehicles. The errors occur randomly in the point clouds so that it is hard for adversaries to utilize. The recall in Figure 12 (d) reaches around 95% in all targets

Table 1: PointPillars' and PointRCNN's APs (%) of 3D car detection on the KITTI validation set. "Mod." refers to the Moderate category introduced in §2.1.1; "Original" refers to the original performance of two models; "Attack" refers to the performance after spoofing attacks; "CARLO" refers to the performance after CARLO applied.

Model	PointPillars			PointRCNN		
	Easy	Mod.	Hard	Easy	Mod.	Hard
Original	86.56	76.87	72.09	88.80	78.58	77.64
Attack	74.06	56.69	53.98	84.51	71.17	68.06
CARLO	86.57	78.60	73.55	88.91	78.61	77.63

models which also validate the results in Figure 12 (a-c).

Besides delivering satisfactory defense performance, CARLO barely introduces misdetections (*i.e.,* false negatives) to the models. Figure 12 (d) shows that the precision of anomaly detection reaches at least 99.5% for all target models. We manually verify the 0.5% misdetections, and find they are all vehicles at least 40 meters away from the AV, which will not affect its immediate driving behavior. Table 1 also shows that the AP will slightly increase after CARLO being applied to the original model because the original model has internal false positives such as detecting a flower bed as a vehicle. CARLO detects some of those false positives generated by the original model.

7.2.1 Defense against White-box and Adaptive Attacks

To further evaluate CARLO against white-box attacks, we first leverage Adv-LiDAR to generate adversarial examples that fool Apollo 5.0 and test whether they can succeed in attacking CARLO-guarded Apollo 5.0. Figure 13 demonstrates that CARLO can effectively defend Adv-LiDAR, where the *ASR* drops from more than 95% to below 5% consistently. We observe that the defense effects are better than the results shown in Figure 12 (a). We find out that Adv-LiDAR tends to translate the attack traces to a slightly higher place along z axis. Such translations will isolate the adversarial examples in the point cloud, making themselves easier to be detected by CARLO.

We also try our best efforts to evaluate CARLO on the adaptive attacks. We assume attackers are aware of the CARLO pipelines and utilize Adv-LiDAR to break CARLO's defense. Due to the sensor attack capability, adversaries have limited ability to modify the absolute free space ($\sum_{c\in\mathsf{B}} \mathbb{1}\cdot FS(c)$) in Equation 5. However, attackers can try to shrink the volume of the bounding box ($|\mathsf{B}|$) to shift the distribution of f', since it is controlled purely by models. Therefore, the attack goal is to spoof a vehicle at target locations at the same time, minimize the size of the bounding box. We formulate the loss function and follow Adv-LiDAR [17] to utilize a global transformation matrix $H(\theta,\tau)$ (Equation 2) for solving the optimization problem:

$$\min_{\theta,\tau} \quad \mathcal{L}(x\oplus V\cdot H(\theta,\tau)^T)+\lambda\cdot\mathcal{V}_B(V\cdot H(\theta,\tau)^T) \qquad (7)$$

where $\mathcal{L}(\cdot)$ is the loss function defined in Adv-LiDAR [17], $\mathcal{V}_B(\cdot)$ is the volume of the target bounding box B, and λ is a hyper-parameter. Figure 14 shows that such adaptive attacks cannot break CARLO, either. We attribute the reason to $H(\theta,\tau)$ that holistically modifies the spoofed points so that it can barely change the size of the bounding box.

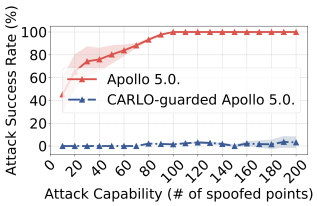


Figure 13: Attack success rates (*ASR*s) of Adv-LiDAR on Apollo 5.0 and CARLO-guarded model.

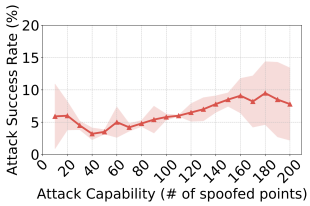


Figure 14: Attack success rate (*ASR*) of the adaptive attack on CARLO-guarded Apollo 5.0.

8 Physics-Embedded Perception Architecture

In this section, we take a step further to explore the feasibility of embedding physical features into end-to-end learning that provides better robustness for AD systems. We find that, despite BEV or 3D representations, which are used by most models, the front view (FV) is a better representation for learning occlusion features by nature. However, prior works adopting FV are still vulnerable to the proposed attacks due to their model architecture designs' fundamental limitations. To improve the design and further enforce the learning of occlusion features, we propose sequential view fusion (SVF), a general architecture for robust LiDAR-based perception.

8.1 Why should FV Representations help?

We observe that LiDAR natively measures range data (§2.2). Thus, projecting the LiDAR point cloud into the perspective of the LiDAR sensor will naturally preserve the physical features of LiDAR. Such projecting is also known as the FV of LiDAR point clouds [38]. Given a 3D point $\vec{p}=(x,y,z)$, we can compute its coordinates in FV $\vec{p}_{FV}=(r,c)$ by:

$$\begin{aligned} c &= \lfloor \arctan(y,x)/\Delta\theta \rfloor \\ r &= \lfloor \arctan(z,\sqrt{x^2+y^2})/\Delta\phi \rfloor \end{aligned} \qquad (8)$$

where $\Delta\theta$ and $\Delta\phi$ are the horizontal and vertical fire angle intervals (§2.2). As shown in Figure 2, since the *occluder* $\mathsf{O}(v)$ and *occludee* V neighbor with each other in the FV, deep learning models have opportunities to identify the inter-occlusion. The abnormal sparseness of a fake vehicle will also be exposed, as valid vehicles' points are clustered, while the spoofed points scatter in the FV (§5.3). Therefore, the FV representation of point clouds embeds both ignored occlusion patterns.

Although prior works have utilized FV for object detection, little is known about its robustness to LiDAR spoofing attacks. LaserNet [43] is the latest model that only takes the FV representation of point clouds as input for 3D object detection. However, LaserNet cannot achieve state-of-the-art performance compared to models in the three classes introduced in §2. Other studies [38] also confirm that only by

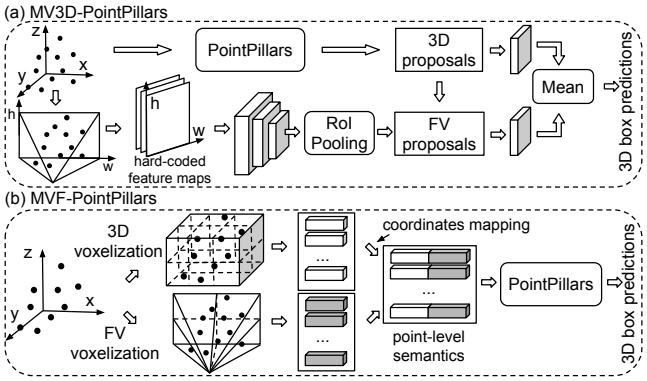


Figure 15: Existing view fusion-based architectures.

leveraging the FV representation, models cannot provide satisfactory detection results. The failure of FV-based models comes from the scale variation of objects as well as occlusions between objects in a cluttered scene [72].

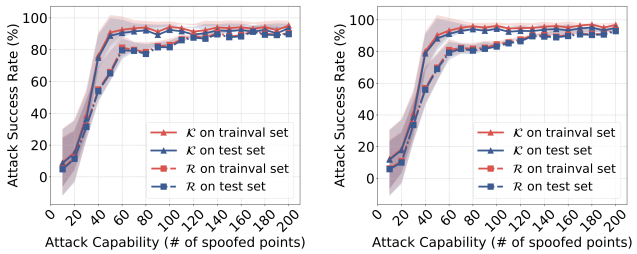
Besides the models that only take FV as input, several studies [24, 35, 72] present fusion-based architectures for LiDAR-based perception that utilize the combinations of data from different sensors and views as input. MV3D [24] is a classic fusion-based design that takes both LiDAR point clouds and RGB images as input and predicts 3D bounding boxes, where the point cloud is projected into multi-views (*i.e.,* FV and BEV) for feature encoding. Zhou *et al.* recently proposed multi-view fusion (MVF), which combines FV with 3D representations [72]. MVF builds on top of PointPillars. Instead of only voxelizing points in 3D, MVF also voxelizes the point cloud into FV frustums and integrates the two voxels' features based on coordination mapping in the 3D space.

To better understand the robustness of fusion-based architectures, we reproduce MV3D and MVF based on PointPillars. For MV3D, we ignore the RGB images, and take the FV and BEV as the model input since we focus on LiDAR-based perception. We use a VGG-16 [56] for FV feature learning in MV3D. Figure 15 shows the architectures we adopt and reproduce. We train the two reproduced models on the KITTI training set and evaluate them on the KITTI validation set. As Table 2 shows, the FV-augmented models can achieve comparable performance than the original PointPillars. The reproduced results also align well with the evaluations in [24, 72].

Table 2: MV3D-PointPillars' and MVF-PointPillars' APs (%) of 3D car detection on the KITTI validation set.

Model	Car Detection		
	Easy	Moderate	Hard
MV3D-PointPillars	85.67	77.12	71.65
MVF-PointPillars	86.77	79.15	75.72

We then evaluate their robustness against our proposed black-box attack. The experimental setups are identical to the settings in §6.1. Figure 16 shows that the *ASR* of reproduced models are as high as the original PointPillars. It indicates that existing view fusion-based architectures both cannot help with defending against LiDAR spoofing attacks, although they provide marginally gain on the AP. We further perform ablation studies and find that the BEV (or 3D) features dominate the model decisions (elaborated in Appendix D). Since the identified vulnerability exists in 3D space, the models are still vulnerable to LiDAR spoofing attacks.



(a) *ASR* of the reproduced MV3D. (b) *ASR* of the reproduced MVF.

Figure 16: Attack success rates (*ASR*s) of proposed black-box spoofing attacks on MV3D and MVF models.

8.2 Sequential View Fusion

The insights drawn from existing view fusion schemes show that existing fusion designs cannot provide better robustness compared to the original models. The 3D (or BEV) representation dominates the model leaving the FV representation not critical in the end-to-end architectures.

Based on the above understandings, we propose a new view fusion schema called sequential view fusion (SVF). SVF comprises of three modules (Figure 17), which are: 1) semantic segmentation: a semantic segmentation network that utilizes the FV representation to computes the point-wise confidence scores (*i.e.,* the probability that one point belongs to a vehicle). 2) view fusion: the 3D representation is augmented with semantic segmentation scores. 3) 3D object detection: a LiDAR-based object detection network that takes the augmented point clouds to predict bounding boxes. Instead of leaving the models to learn the importance of different representations by themselves, we attach a semantic segmentation network to the raw FV data. By doing so, we enforce the end-to-end learning to appreciate the FV features, so that the trained model will be resilient to LiDAR spoofing attacks.

Semantic segmentation. The semantic segmentation networks accept the FV represented point clouds and associate each point in FV with a probability score that it belongs to a vehicle. These scores provide aggregated information on the FV representation. Semantic segmentation over FV has several strengths. First, as mentioned before, the FV representation is noisy because of the nature of LiDAR. Compared to 3D object detection or instance segmentation, which is intractable over FV, semantic segmentation is an easier task as it does not need to estimate object-level output. Second, there are extensive studies on semantic segmentation over FV represented point clouds [15, 59, 63], and the segmentation networks achieve much more satisfactory results than the 3D object detection task over FV.

In our implementation, we adopt the high-level design in LU-Net [15]. It is worth noting that the end-to-end SVF architecture is agnostic to the semantic segmentation module.

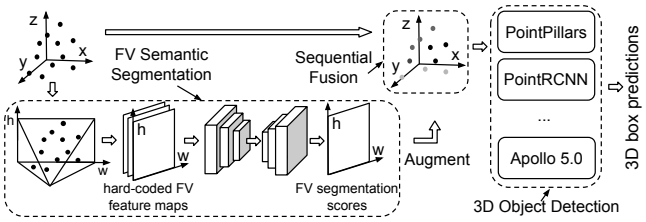


Figure 17: Sequential view fusion (SVF) architecture.

View fusion. The fusion module re-architects existing symmetric designs which integrate the 3D representation with the confidence scores generated by the semantic segmentation module. Specifically, we use Equation 8 for mapping between $\vec{p} = (x, y, z)$ and $\vec{p}_{FV}(r, c)$, and augment each $\vec{p}$ with the pointwise confidence score from its corresponding $\vec{p}_{FV}$.

3D object detection. SVF is also agnostic to the 3D object detection module. In this paper, we utilize PointPillars and PointRCNN in our implementation. Most of the models introduced in §2 can fit into the end-to-end SVF architecture.

8.3 SVF Evaluation

Experimental setup. We train SVF-PointPillars and SVF-PointRCNN on the KITTI training set. The setup of robustness analysis against LiDAR spoofing attacks is identical to the settings in §6.1. We also try to evaluate SVF against Adv-LiDAR [17] on Apollo 5.0 and the adaptive attacks.

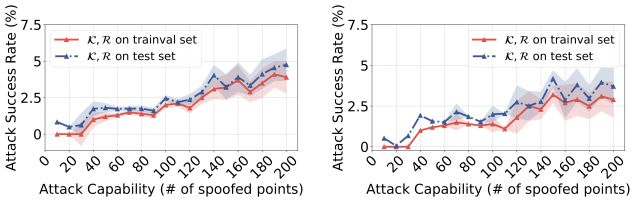
Evaluation metrics. We evaluate the AP of SVF-PointPillars and SVF-PointRCNN on the KITTI validation set, and leverage *ASR* to test their robustness against LiDAR spoofing attacks.

As shown in Table 3, both SVF models achieve comparable AP compared to the original models. The marginal degradation comes from two-state training. More specifically, the distributions of the semantic segmentation outputs in the training and validation sets do not align well with each other. We find that the drop of AP will indeed cause a tiny amount of false negatives but will not influence the driving behaviors. Moreover, such degradation could be compensated with better training strategies (*e.g.,* finer-tuning of the parameters) since the capacity of SVF is larger than the original models.

Table 3: SVF-PointPillars' and SVF-PointRCNN's APs (%) of 3D car detection on the KITTI validation set.

Model	Car Detection		
	Easy	Moderate	Hard
SVF-PointPillars	85.93	74.12	70.19
SVF-PointRCNN	88.12	76.56	74.81

We then perform the robustness evaluation of SVF models. Figure 18 shows the *ASR* of our proposed spoofing attacks. As shown, the attacks are no longer effective in SVF models. The *ASR* reduces from more than 95% (original models) to less than 4.5% on both models with the maximum attack capability, which is also an around 2.2× improvement compared to CARLO-guarded models. The mean *ASR* also drops from 80% to around 2.3%. We also perform ablation study on SVF,



(a) *ASR* of SVF-PointPillars. (b) *ASR* of SVF-PointRCNN.

Figure 18: Attack success rates (*ASR*s) of proposed black-box spoofing attack on SVF models.

and demonstrate that the FV features are more important in SVF models (detailed in Appendix D).

8.3.1 Defense against White-box and Adaptive Attacks

Since SVF requires re-training for the model, we cannot directly evaluate Adv-LiDAR on SVF-Apollo (§2.1.1). As a result, we decouple the problem to whether Adv-LiDAR can fool both the semantic segmentation and 3D object detection modules. We first directly apply the attack traces that successfully fool Apollo 5.0 to the segmentation network and record the mean confidence score of all the points belonging to the attack trace. Figure 19 shows that the mean confidence scores are consistently below 0.08, which is too low to be classified as a valid vehicle with mean confidence scores of around 0.73 in our trained model.

Model-level defenses are usually vulnerable to simple adaptive attacks [13, 19]. To demonstrate the effectiveness of SVF against adaptive attack, we assume that the adversaries are aware of the SVF architecture. The attack goal is to both fool the semantic segmentation and 3D object detection modules. We also leverage the formulation in [17] to utilize the global transformation matrix $H(\theta, \tau)$ to control the spoofed points.

$$\min_{\theta,\tau} \quad -\mathcal{L}_{seg}(x \odot V \cdot H(\theta,\tau)^T) \tag{9}$$

where $\odot$ represents the point cloud merge in the front view and $\mathcal{L}_{seg}(\cdot)$ defines the average confidence score of the attack trace (*i.e.,* $V \cdot H(\theta,\tau)^T$). Figure 20 shows that none of the attack traces' average confidence score reaches 0.2 in the segmentation module, which is still far from the mean average confidence score of valid vehicle 0.73. Therefore, the adaptive attacks also cannot break the robustness of SVF.

9 Discussion and Future Work

In this section, we discuss the distinct features of our proposed black-box attack along with its practicality and completeness. We further discuss the comparisons between the presented defense strategies and their limitations, accordingly.

9.1 Attack Discussion

9.1.1 Comparison with Physical Adversarial Attacks

First, we distinguish the spoofing attacks on LiDAR with extensive prior work on physical-world adversarial machine learning attacks in mainly three aspects:

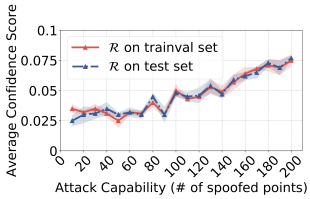


Figure 19: Average confidence score of Adv-LiDAR on the segmentation network.

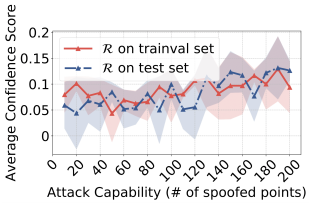


Figure 20: Average confidence score of the adaptive attack on the segmentation network.

1. Different perturbation methods. Images and point clouds have different data structures, which further lead to different perturbation methods applied. Images have compact and ordered structures. In contrast, point clouds are irregular, represented as $N \times C$, where N is the number of points, and C contains the location and intensity information (*i.e., xyz-i*) [9]. Attackers are able to generate adversarial examples by modifying the RGB values in images. For attacks on LiDAR, however, attackers can directly shift the point in the 3D Euclidean space as long as it obeys the physics of LiDAR.

2. Different perturbation capabilities. Prior attacks on 2D images treat the whole target area as the attack surface since the threat model assumes that attackers have full controls over the target object (*e.g.,* attackers can potentially modify any area of a stop sign in [28]). However, due to the characteristics of LiDAR spoofing attacks, the attack surface is limited by the sensor attack capability ($\mathcal{A}$) in §3. Such a small attack surface also introduces difficulties in launching the attack.

3. Different perturbation constraints. Prior attacks on 2D images leverage L_p norms as the main constraints for the formulated optimization problem [28] whose goal is to minimize the perturbation to be stealthy. Such constraints do not apply to attack LiDAR because point clouds are not perceived by humans. Thus, stealthiness is not a focus in such attacks. Instead, the optimized attack traces must not exceed the sensor attack capability ($\mathcal{A}$) boundary, in which case $\mathcal{A}$ naturally becomes the primary constraint for attacking LiDAR.

Second, the high-level methodology of our proposed attack is similar to replay attacks, in which adversaries playback the intercepted data to deceive target systems [11]. However, different from existing replay attacks [44] that retransmit *logically correct* data to launch attacks, the limited sensor attack capability ($\mathcal{A}$) cannot support the injection of a *physically valid* vehicle's trace into the LiDAR point cloud [17]. Thus, the success of our black-box attack indeed relies on the identified vulnerability.

9.1.2 Attack Practicality and Completeness

One major limitation of our proposed attack is that the presented results cannot directly demonstrate attack practicality in the physical world. First, due to the limitations of our delay component (*i.e.,* the function generator in our implementation), we can only control spoofed points at 10cm-level precision. Therefore, we only construct two fine-controlled physical attack traces for a proof-of-concept demonstration. The two attack traces contain 140 and 47 points. We evaluated them on the KITTI trainval set, and they achieve 87.68%, 98.12%, and 74.91% *ASR*s on Apollo 5.0, PointPillars, and PointRCNN, respectively. Second, launching our black-box attack on a real AV requires accurate aiming of attack lasers at a target LiDAR, which is challenging to perform without real-world road tests and precision instruments [17]. Since this paper aims to explore and expose the underlying vulnerability, we leave real-world testing as future work.

Although we demonstrate that our proposed black-box attack achieves high attack success rates, the identified vulnerability does not provide *completeness*. This means that there may exist other potential vulnerabilities hidden in the AD systems to be discovered and exploited. Future research may include verification of the AD models and comprehensive empirical studies to explore the underlying vulnerabilities.

9.2 Defense Discussion

CARLO vs. SVF. Both CARLO and SVF achieve satisfactory defense performance while maintaining comparable AP with the original model. In addition, both of them are model-agnostic so that they can be incorporated into most existing LiDAR-based perception systems. CARLO is a practical post-detection module. It does not require re-training the model, which can be quite labor-intensive. CARLO is also realistic because it does not assume that users have white-box access to the model. SVF, on the other hand, is a general architecture for ensuring robust LiDAR-based perception. SVF embeds physical information into model learning, which requires re-training. Compared to CARLO, SVF achieves better defense performance but suffers from a slight drop in AP, indicating that it may require more training efforts.

Limitations. The main limitation of our mitigation strategies is the lack of guarantees. First, although both defenses can effectively defend against LiDAR spoofing attacks under the current sensor attack capability, our countermeasures may not work at some point with the increasing capability of sensor attacks. We argue that if attackers can spoof a set of points located in the distribution of physical invariants for valid vehicles (*e.g.,* injecting around 1500 points into the point cloud), there is arguably no way to distinguish them at the model level and it is safer for AVs to engage emergency brakes in that situation. Second, both defenses have a small portion of false alarms (*i.e.,* the 0.5% false negatives in CARLO and the slight AP drop of SVF). However, we manually verify that they are not front-near vehicles; hence they would not impact the AV's driving behaviors, as mentioned before. Third, since the adaptive attacks are formulated with our efforts, future research may present more powerful attacks or advanced perturbation methods to break our defenses. In the future, we plan to improve SVF to provide guaranteed robustness by combining multiple sensors' inputs.

10 Related Work

Vehicular system security. Extensive prior works explore security problems in vehicular systems and have identified

vulnerabilities in in-vehicle networks of modern automobiles [10, 21, 26, 34], in-vehicle cache side channels [12], and Connected Vehicle (CV)-based systems [22, 29, 61]. Other studies try to provide robustness vehicular systems, such as secured in-vehicle communications [14, 47, 62], secured in-vehicle payment transactions [30], and secured CV communications [50]. In comparison, our work focuses on the emerging autonomous vehicle systems and specifically targets the robustness of LiDAR-based perception in AVs, which are under-explored in previous studies.

3D adversarial machine learning. Adversarial attacks and defenses towards 3D deep learning have been increasingly explored recently. Point cloud classification models have been demonstrated vulnerable to adversarial perturbations [57, 60, 64]. Xiao *et al.* generate adversarial examples for 3D mesh classification [65]. Liu *et al.* and Yang *et al.*, on the other hand, leverage heuristics to detect the adversarial examples for point cloud classification [41, 69]. In comparison, our work targets LiDAR-based 3D object detection in AVs. As introduced in §2.2, LiDAR point clouds only have measurements of the object's facing surface, which are different from full 3D point cloud data or meshes. Our attack method is motivated to generate adversarial examples in a black-box manner based on the discovered vulnerability. The mitigation strategies are designed to defend against current sensor attack capability, thus provide better robustness against both white- and black-box LiDAR spoofing attacks.

11 Conclusion

In this paper, we perform the first study to explore the general vulnerability of LiDAR-based perception architectures. We construct the first black-box spoofing attack based on the identified vulnerability, which universally achieves an 80% mean success rate on target models. We further perform the first defense study, proposing CARLO to accurately detect spoofing attacks which reduce their success rate to 5.5%. Lastly, we present SVF, the first general architecture for robust LiDAR-based perception that reduces the mean spoofing attack success rate to 2.3%.

12 Acknowledgements

We appreciate our shepherds, Xiaoyu Ji and Wenyuan Xu, and the anonymous reviewers for their insightful comments. We thank Xiao Zhu, Shengtuo Hu, Jiwon Joung, and Xumiao Zhang for proofreading our paper. This project is partially supported by Mcity and NSF under the grants CNS-1930041, CNS-1850533, CNS-1929771, and CNS-1932464.

References

[1] Baidu debuts robotaxi ride hailing service in China, using self-driving electric taxis. https://www.marketwatch.com/story/baidu-debuts-robotaxi-ride-hailing-service-in-china-using-self-driving-electric-taxis-2019-09-26.

[2] UPS joins race for future of delivery services by investing in self-driving trucks. https://abcnews.go.com/Business/ups-joins-race-future-delivery-services-investing-driving/story?id=65014414.

[3] Waymo has launched its commercial self-driving service in Phoenix - and it's called 'Waymo One'. https://www.businessinsider.com/waymo-one-driverless-car-service-launches-in-phoenix-arizona-2018-12.

[4] GM Advances Self-Driving Vehicle Deployment With Acquisition of LIDAR Developer. https://media.gm.com/media/us/en/gm/news.detail.html/content/Pages/news/us/en/2017/oct/1009-lidar1.html, 2017.

[5] Introducing Laser Bear Honeycomb by Waymo. https://waymo.com/lidar/, 2019.

[6] Autoware.AI. https://www.autoware.ai/, 2020.

[7] Baidu Apollo. http://apollo.auto, 2020.

[8] Devil's whisper: A general approach for physical adversarial attacks against commercial black-box speech recognition devices. In *29th USENIX Security Symposium (USENIX Security 20)*, Boston, MA, Aug. 2020. USENIX Association.

[9] KITTI Vision Benchmark: 3D Object Detection. http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d, 2020.

[10] Plug-n-pwned: Comprehensive vulnerability analysis of obd-ii dongles as a new over-the-air attack surface in automotive iot. In *29th USENIX Security Symposium (USENIX Security 20)*, Boston, MA, Aug. 2020. USENIX Association.

[11] Replay attack. https://www.tek.com/signal-generator/afg2021-software-0, 2020.

[12] Stealthy tracking of autonomous vehicles with cache side channels. In *29th USENIX Security Symposium (USENIX Security 20)*, Boston, MA, Aug. 2020. USENIX Association.

[13] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

[14] O. Avatefipour, A. S. Al-Sumaiti, A. M. El-Sherbeeny, E. M. Awwad, M. A. Elmeligy, M. A. Mohamed, and H. Malik. An intelligent secured framework for cyberattack detection in electric vehicles' can bus using machine learning. *IEEE Access*, 7:127580–127592, 2019.

[15] P. Biasutti, V. Lepetit, J.-F. Aujol, M. Brédif, and A. Bugeau. Lu-net: An efficient network for 3d lidar point cloud semantic segmentation based on end-to-end-learned 3d features and u-net. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[16] J. Bresenham. A linear algorithm for incremental digital display of circular arcs. *Communications of the ACM*, 20(2):100–106, 1977.

[17] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu, and Z. M. Mao. Adversarial sensor attack on lidar-based perception in autonomous driving. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 2267–2281. ACM, 2019.

[18] Y. Cao, C. Xiao, D. Yang, J. Fang, R. Yang, M. Liu, and B. Li. Adversarial objects against lidar-based autonomous driving systems. *arXiv preprint arXiv:1907.05418*, 2019.

[19] N. Carlini and D. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14, 2017.

[20] A. Chaman, J. Wang, J. Sun, H. Hassanieh, and R. Roy Choudhury. Ghostbuster: Detecting the presence of hidden eavesdroppers. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 337–351, 2018.

[21] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno. Comprehensive Experimental Analyses of Automotive Attack Surfaces. In *Proceedings of the 20th USENIX Conference on Security*, SEC'11, 2011.

[22] Q. A. Chen, Y. Yin, Y. Feng, Z. M. Mao, and H. X. L. Liu. Exposing Congestion Attack on Emerging Connected Vehicle based Traffic Signal Control. In *Proceedings of the 25th Annual Network and Distributed System Security Symposium*, NDSS '18, 2018.

[23] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun. 3d object proposals for accurate object class detection. In *Advances in Neural Information Processing Systems*, pages 424–432, 2015.

[24] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017.

[25] Y. Chen, S. Liu, X. Shen, and J. Jia. Fast point r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9775–9784, 2019.

[26] K.-T. Cho and K. G. Shin. Error handling of in-vehicle networks makes them vulnerable. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS'16, 2016.

[27] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[28] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018.

[29] Y. Feng, S. Huang, Q. A. Chen, H. X. Liu, and Z. M. Mao. Vulnerability of Traffic Control System Under Cyber-Attacks Using Falsified Data. In *Transportation Research Board 2018 Annual Meeting (TRB)*, 2018.

[30] A. Gaddam, G. Prakash, and S. Aissi. Mechanism for secure in-vehicle payment transaction, Feb. 26 2015. US Patent App. 14/466,405.

[31] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[32] V. L. Inc. VLP-16 User Manual, 2018.

[33] T. Kim and J. Ghosh. On single source robustness in deep fusion models. *arXiv preprint arXiv:1906.04691*, 2019.

[34] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage. Experimental Security Analysis of a Modern Automobile. In *Proceedings of the 2010 IEEE Symposium on Security and Privacy*, SP'10, 2010.

[35] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018.

[36] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019.

[37] J. Lehner, A. Mitterecker, T. Adler, M. Hofmarcher, B. Nessler, and S. Hochreiter. Patch refinement–localized 3d object detection. *arXiv preprint arXiv:1910.04093*, 2019.

[38] B. Li, T. Zhang, and T. Xia. Vehicle detection from 3d lidar using fully convolutional network. *arXiv preprint arXiv:1608.07916*, 2016.

[39] Y. Li, J. Sun, W. Huang, and X. Tian. Detecting anomaly in large-scale network using mobile crowdsourcing. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 2179–2187. IEEE, 2019.

[40] M. Liang, B. Yang, S. Wang, and R. Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 641–656, 2018.

[41] D. Liu, R. Yu, and H. Su. Extending adversarial attacks and defenses to deep 3d point cloud classifiers. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 2279–2283. IEEE, 2019.

[42] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[43] G. P. Meyer, A. Laddha, E. Kee, C. Vallespi-Gonzalez, and C. K. Wellington. Lasernet: An efficient probabilistic 3d object detector for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12677–12686, 2019.

[44] B. Nassi, D. Nassi, R. Ben-Netanel, Y. Mirsky, O. Drokin, and Y. Elovici. Phantom of the adas: Phantom attacks on driver-assistance systems, 2020.

[45] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017.

[46] Y. Park, Y. Son, H. Shin, D. Kim, and Y. Kim. This ain't your dose: Sensor spoofing attack on medical infusion pump. In *10th USENIX Workshop on Offensive Technologies (WOOT 16)*, Austin, TX, Aug. 2016. USENIX Association.

[47] C. Patsakis, K. Dellios, and M. Bouroche. Towards a distributed secure in-vehicle communication architecture for modern vehicles. *Computers & security*, 40:60–74, 2014.

[48] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl. Remote attacks on automated vehicles sensors: Experiments on camera and lidar. *Black Hat Europe*, 11:2015, 2015.

[49] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[50] A. Rao, A. Sangwan, A. A. Kherani, A. Varghese, B. Bellur, and R. Shorey. Secure v2v communication with certificate revocations. In *2007 Mobile Networking for Vehicular Environments*, pages 127–132. IEEE, 2007.

[51] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[52] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[53] S. Shi, X. Wang, and H. Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019.

[54] S. Shi, Z. Wang, X. Wang, and H. Li. Part-aˆ 2 net: 3d part-aware and aggregation neural network for object detection from point cloud. *arXiv preprint arXiv:1907.03670*, 2019.

[55] H. Shin, D. Kim, Y. Kwon, and Y. Kim. Illusion and dazzle: Adversarial optical channel exploits against lidars for automotive applications. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 445–467. Springer, 2017.

[56] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[57] T. Tsai, K. Yang, T.-Y. Ho, and Y. Jin. Robust adversarial objects against deep learning models.

[58] B. Wang, J. An, and J. Cao. Voxel-fpn: multi-scale voxel feature aggregation in 3d object detection from point clouds. *arXiv preprint arXiv:1907.05286*, 2019.

[59] Y. Wang, T. Shi, P. Yun, L. Tai, and M. Liu. Pointseg: Real-time semantic segmentation based on 3d lidar point cloud. *arXiv preprint arXiv:1807.06288*, 2018.

[60] Y. Wen, J. Lin, K. Chen, and K. Jia. Geometry-aware generation of adversarial and cooperative point clouds. *arXiv preprint*

arXiv:1912.11171, 2019.

[61] W. Wong, S. Huang, Y. Feng, Q. A. Chen, Z. M. Mao, and H. X. Liu. Trajectory-Based Hierarchical Defense Model to Detect Cyber-Attacks on Transportation Infrastructure. In *Transportation Research Board 2019 Annual Meeting (TRB)*, 2019.

[62] S. Woo, H. J. Jo, I. S. Kim, and D. H. Lee. A practical security architecture for in-vehicle can-fd. *IEEE Transactions on Intelligent Transportation Systems*, 17(8):2248–2261, 2016.

[63] B. Wu, A. Wan, X. Yue, and K. Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1887–1893. IEEE, 2018.

[64] C. Xiang, C. R. Qi, and B. Li. Generating 3d adversarial point clouds. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[65] C. Xiao, D. Yang, B. Li, J. Deng, and M. Liu. Meshadv: Adversarial meshes for visual recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[66] W. Xu, D. Evans, and Y. Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

[67] Y. Yan, Y. Mao, and B. Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.

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

[69] J. Yang, Q. Zhang, R. Fang, B. Ni, J. Liu, and Q. Tian. Adversarial attack and defense on point sets. *arXiv preprint arXiv:1902.10899*, 2019.

[70] Y. Yang, G. Zhang, D. Katabi, and Z. Xu. ME-Net: Towards effective adversarial robustness with matrix estimation. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

[71] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1951–1960, 2019.

[72] Y. Zhou, P. Sun, Y. Zhang, D. Anguelov, J. Gao, T. Ouyang, J. Guo, J. Ngiam, and V. Vasudevan. End-to-end multi-view fusion for 3d object detection in lidar point clouds. *arXiv preprint arXiv:1910.06528*, 2019.

[73] Y. Zhou and O. Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018.

Appendices

A Spoofing Attack Details

The attack consists of three modules: a photodiode, a delay component, and an infrared laser [55]. The photodiode functions as a synchronizer that triggers the delay component whenever it captures laser signals from the victim LiDAR sensor. The delay component triggers the laser module after a configurable time delay to attack the following firing cycles of the victim LiDAR sensor. The attack can be programmatically controlled so that an adversary can target different locations and angles in the point cloud. Specifically, we use an OSRAM SFH 213 FA as the photodiode, a Tektronix AFG3251 function generator as the delay component, and a PCO-7114 laser driver that drives the attack laser OSRAM SPL PL90 in our setups. Figure 21 shows the physical spoofing attack conducted in a controlled environment.

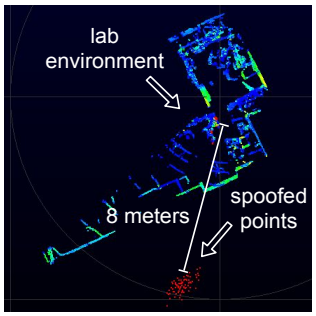


Figure 21: Physical spoofing in in-lab environments.

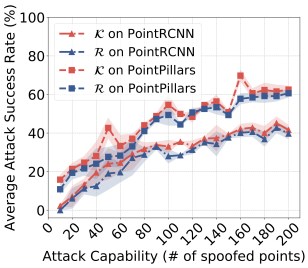


Figure 22: Average attack success rates (A^2SRs) of proposed black-box attack on PointPillars and PointRCNN.

B Supplementary Attack Evaluation

We define a new metric for general evaluations on the object detection-based attacks called average attack success rate (A^2SR). As mentioned before, the default thresholds are empirically set. Thus, evaluations of *ASR* provide limited insights. Similar to AP defined in PASCAL [27] criteria, we average the *ASR* in 11 recall intervals to better understand the impact of the proposed attacks and the characteristics of different architectures:

$$A^2SR = \frac{1}{11} \sum_{r \in \{0.0, 0.1, \ldots, 1.0\}} ASR_{t_r} \tag{10}$$

where t_r represents the threshold that makes the recall of the target model at r. The evaluation of recall follows the description of the Moderate category in §2.1.1. In this paper, we test A^2SR on PointPillars and PointRCNN since Apollo model is not designed to be evaluated on KITTI (§2.1.1).

Figure 22 shows that the A^2SR of PointPillars is generally higher than PointRCNN, which means the spoofed points can achieve higher confidences in PointPillars. Such results are expected since point-wise features contain more detailed information than voxel-based features; hence point-wise features could be more resilient to spoofing attacks.

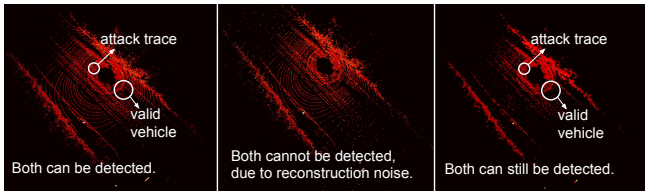


Figure 23: Illustrative example: the left figure is the original feature map of a point cloud sample from Apollo 5.0; the middle one is the feature map after ME reconstruction; and the right one is the feature map after squeezing.

We also leverage feature squeezing [66] and ME-Net [70] to evaluate our proposed attack on Apollo 5.0. We utilize median smoothing as the method for feature squeezing, and follow general settings in [70] for matrix estimation. We perform evaluations on 100 samples from the KITTI validation set. Results show that ME-Net can eliminate the fake vehicle but introduce new false negatives, which will lead to more

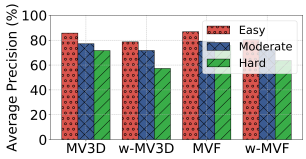


Figure 24: APs of weakened view fusion-based models (w-: weakened models).

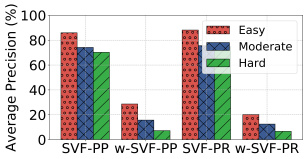


Figure 25: APs of weakened SVF models (PP: PointPillars; PR: PointRCNN).

severe safety issues. In contrast, feature squeezing cannot effectively eliminate fake vehicles, as shown in Figure 23.

C CARLO Algorithm Details

Algorithm 1 shows the detailed CARLO algorithm combined with its two building blocks: FSD and LPD. Especially, to estimate the free space inside a bounding box B, we first extract all the laser fires that have chances to hit B, which form a frustum in the 3D space. We then grid the 3D Euclidean space of the frustum into small 3D cells and initialize all the cells as occluded cells in the beginning. For each laser, we use 3D Bresenham's line algorithm [16] to compute the cells it traversed from the origin of the laser beam (*i.e.,* the LiDAR sensor) to the end (*i.e.,* the hit point). If a cell is traversed by a laser beam, we label it as a free cell because it does not belong to a solid object. Finally, we union the free cells for all the possible laser rays to get the total free cells in the frustum.

D Ablation Study of View Fusion Models

We perform ablation studies to explore the reasons behind the results shown in Figure 16. In particular, we find most of the existing fusion-based models utilize a symmetric design where the FV and 3D (or BEV) features are fed into similar modules for learning, and the learned features are simply stacked or averaged for later stages (Figure 15). We design experiments to study the effectiveness of such a design empirically. Explicitly, we zero out the features from FV to measure how much the FV representation contributes to the final detection. As shown in Figure 24, the APs only have relatively small degradation, which implies that BEV (or 3D) features dominate the model decisions. Kim *et al.* also empirically demonstrates that current sensor fusion-based models are vulnerable to single-source perturbations [33]. Similarly, we showcase that current view fusion-based models are vulnerable to the perturbation represented in the dominated view.

To better understand why SVF models can provide better robustness, we analyze how the FV representation helps in SVF models. Similarly, we zero out the augmented confidence score features and evaluate the AP. Figure 25 shows the weakened models' performance, which empirically demonstrates that the features from FV account more in SVF models.

E Supplementary Figures

Figure 26 shows an illustrative example that translated points from Figure 2 can be detected as a valid vehicle in PointRCNN. Figure 27 shows more rendered original attack traces.

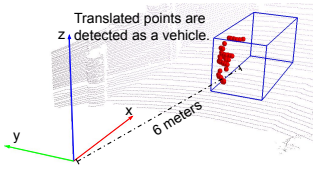


Figure 26: Translated points from Figure 2 are detected as a valid vehicle by PointRCNN.

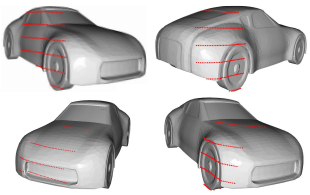


Figure 27: More examples of our rendered traces with occlusions.

Algorithm 1: CARLO

input: Detected bounding boxes $\boldsymbol{B} = \{\mathsf{B}\}$;
LiDAR laser ray directions $\boldsymbol{L} = \{\mathsf{L}\}$;
3D point cloud $\boldsymbol{X} = \{\vec{p}\}$;
Threshold of FSD $\frac{a+b}{2}$;
Thresholds of LPD $b'+\varepsilon$, $a'-\varepsilon$;
output: Valid bounding boxes $\boldsymbol{B}_{\text{valid}} = \{\mathsf{B}\}$;
Adversarial bounding boxes $\boldsymbol{B}_{\text{adv}} = \{\mathsf{B}\}$;

Initialization : $\boldsymbol{B}_{\text{valid}} \leftarrow \emptyset$, $\boldsymbol{B}_{\text{adv}} \leftarrow \emptyset$, $g \leftarrow 0$, $f \leftarrow 0$;
/* Initiate parameters. */
for $\mathsf{B} \in \boldsymbol{B}$ **do**
  /* Initiate parameters, where $FS(\cdot)$ is the free space and F_B is the frstum of B. */
  $\mathsf{F}_\mathsf{B} \leftarrow \emptyset$, $FS(\cdot) \leftarrow \emptyset$;
  for $\mathsf{L} \in \boldsymbol{L}$ **do**
    /* Predict whether L will intersect with B. */
    if $\mathsf{L} \cap \mathsf{B}$ **then**
      $\vec{p}_\mathsf{L} \leftarrow \mathsf{L}$;
      F_B.append($[\mathsf{L}, \vec{p}_\mathsf{L}]$);
    /* Extract the frustum F_B of B. */
  end
  $g \leftarrow$ Equation 6;
  /* Calculate g by F_B for B (LPD). */
  if $g < a' - \varepsilon$ **then**
    $\boldsymbol{B}_{\text{valid}}$.append($\mathsf{B}$);
    /* Certainly valid vehicles. */
  else if $g > b' + \varepsilon$ **then**
    $\boldsymbol{B}_{\text{adv}}$.append($\mathsf{B}$);
    /* Certainly spoofed vehicles. */
  else
    /* Calculate$f$ by F_B for B (FSD). */
    for $[\mathsf{L}, \vec{p}_\mathsf{L}] \in \mathsf{F}_\mathsf{B}$ **do**
      $FS(\mathsf{L}) \leftarrow$ Bresenham($[\mathsf{L}, \vec{p}_\mathsf{L}]$)[16];
      $FS(\mathsf{B}) \leftarrow FS(\mathsf{B}) \cup FS(\mathsf{L})$;
    end
    $f \leftarrow$ Equation 5;
    if $f < \frac{a+b}{2}$ **then**
      $\boldsymbol{B}_{\text{valid}}$.append($\mathsf{B}$);
    else
      $\boldsymbol{B}_{\text{adv}}$.append($\mathsf{B}$);
end
Return : $\boldsymbol{B}_{\text{valid}}$, $\boldsymbol{B}_{\text{adv}}$;