Security Analysis of Camera-LiDAR Fusion Against Black-Box Attacks on Autonomous Vehicles

R. Spencer Hallyburton
Duke University

Yupei Liu
Duke University

Yulong Cao
University of Michigan

Z. Morley Mao
University of Michigan

Miroslav Pajic
Duke University

Abstract

To enable safe and reliable decision-making, autonomous vehicles (AVs) feed sensor data to *perception* algorithms to understand the environment. Sensor fusion with multi-frame tracking is becoming increasingly popular for detecting 3D objects. Thus, in this work, we perform an analysis of camera-LiDAR fusion, in the AV context, under LiDAR spoofing attacks. Recently, LiDAR-only perception was shown vulnerable to LiDAR spoofing attacks; however, we demonstrate these attacks are not capable of disrupting camera-LiDAR fusion. We then define a novel, context-aware attack: *frustum attack*, and show that out of 8 widely used perception algorithms – across 3 architectures of LiDAR-only and 3 architectures of camera-LiDAR fusion – *all* are significantly vulnerable to the frustum attack. In addition, we demonstrate that the frustum attack is stealthy to existing defenses against LiDAR spoofing as it preserves consistencies between camera and LiDAR semantics. Finally, we show that the frustum attack can be exercised consistently over time to form stealthy longitudinal attack sequences, compromising the tracking module and creating adverse outcomes on end-to-end AV control.

1 Introduction

Autonomous vehicles (AVs) have enjoyed millions of miles of partially automated road travel [1, 2]. This has been enabled by advances in *perception*, the foundation for safe and reliable decision-making in AVs. Sensors including cameras and light detection and ranging (LiDAR) collect data so perception can provide AVs enough awareness of surroundings to make informed decisions in safety-critical tasks such as obstacle/pedestrian avoidance and traffic sign detection.

The camera and LiDAR are the most used AV sensors [3–6]. Inexpensive, high-quality cameras can provide high resolution, dense 2D outputs on limited fields of view. LiDAR is complementary to the camera, providing up to 360° view of the surroundings and fully resolving the 3D position of objects with a sparse set of points (i.e., point clouds).

Due to AVs' safety-critical nature, misinformation or wrong decisions can quickly lead to severe adverse outcomes [7, 8]. The high-impact outcomes underscore the need for security research in the domain. In particular, the increasing reliance of AVs on deep neural networks (DNNs) for real-time perception has sparked security questions at the algorithm level. There is a growing body of AV perception security work: an attacker can perturb sensor data to change object classification (misclassification) [9], introduce fake objects (false positives) [10, 11], and remove existing objects (false negative) [12, 13], each with devastating consequences at the driving decision and control levels.

Initially, security analysis of perception focused on the image domain with LiDAR only recently emerging as the target for security research. Spoofing attacks against LiDAR have since been demonstrated [10, 11, 14–16], and applied to LiDAR-only perception [10, 11]. The use of physical adversarial objects has also been explored [12, 13, 17], demonstrating outcomes against end-to-end AV pipelines [17].

However, existing security analyses of LiDAR-based perception have several limitations. Reported physically-realizable attacks mainly consider single-sensor (e.g., LiDAR-only, camera-only) perception. On the other hand, deployed AV architectures such as Waymo's One [1], Baidu's Apollo [6], and NVIDIA's DRIVE [5] employ multi-sensor perception with multi-frame tracking. Security analysis of multi-sensor fusion has been recently considered [12, 13, 17]; e.g., [17] focuses on the impact of adversarial physical objects on camera-LiDAR perception. Yet, these approaches require highly representative models of the deployed perception algorithms to design attacks with white-box optimization online or a-priori. To the best of our knowledge, there is no analysis of black-box (i.e., when the perception model is not known to the attackers) attacks against sensor-fusion perception.

Consequently, in this work, *we present security analysis for camera-LiDAR sensor fusion under physically-demonstrated black-box LiDAR spoofing attacks*. Using the LiDAR-spoofing threat model from [10, 11], we first show that *camera-LiDAR fusion confers additional robustness against general black-box (i.e., naive) LiDAR attacks*; this is because the naive spoofing does not retain consistency be-

tween camera data and thus can be filtered. That attack success may be greatly reduced with sensor fusion when not all sensors are compromised has been suggested in prior works [10–12, 18, 19], and is systematically evaluated for the first time in this work.

Unlike the recent work of [18] that restricts analysis to naive LiDAR attacks, we introduce a new class of perception attacks, the *frustum attack*, which compromises camera-LiDAR fusion by preserving semantic consistencies between the camera and LiDAR data. To achieve this, the attacker only needs to know approximate locations of true objects in the scene. We experimentally demonstrate that the frustum attack can be executed in the physical world with limited knowledge. We describe five scenarios where an adversary can use contextual information to *launch spoofing attacks relative to existing objects in the scene*. This expands upon prior works [10, 11] that focused only on naive, isolated placement at 5-8 m range.

We then evaluate the frustum attack against state-of-the art defenses against LiDAR spoofing [11, 18, 20] using a diverse set of eight perception algorithms across three distinct LiDAR-only and three distinct camera-LiDAR fusion architectures, including cascaded-semantic, feature-level, and tracking-level fusions (Fig. 1) on over 75 million attack scenarios. To the best of our knowledge, this constitutes the largest analysis of LiDAR spoofing to date and the first that extensively evaluates multiple architectures of multi-sensor fusion for perception.

In addition to false positives (FPs), we demonstrate that the frustum attack is successful in generating false negative (FN) and translation attack outcomes, as defined in Section 3.1, which is a novel discovery for LiDAR spoofing attacks.

We also show that a key assumption about the required attacker's capabilities from prior work can be relaxed. Existing spoofing attacks have only had success at creating FPs or FNs with precise (cm-level) placement of points; furthermore, existing LiDAR spoofing attacks have required either white-box model access [10] or carefully-crafted point placements in the outline of real vehicles (e.g., adversary pre-captures samples and replays them [11]). We establish that *inserting a random sample of normally-distributed points is comparably as successful as inserting points in the outline of a car.* This confers inherent attack robustness to small perturbations, facilitating attack deployment with a physical spoofing device such as in (e.g., [10, 14–16]), and as demonstrated in Section 5.3.1.

Finally, to assess the impact of LiDAR attacks on AVs equipped with multi-frame tracking, we present frustum attack case studies using longitudinal sequences of perception data. First, we explicitly analyze the multi-frame fusion and tracking module, which is employed by all industry AVs, using representative algorithms. Then, we test the frustum attack on Baidu Apollo [6] using the LGSVL simulator [21]. The case studies illuminate that high-impact adversarial situations that endanger vehicle and passenger safety occur under the frustum attack when attacking over multiple time points, effectively deceiving the host vehicle's tracking and control.

In summary, we make the following main contributions:

- We show that several sensor-fusion algorithms are robust to naive LiDAR spoofing at some of the highest defensive rates yet observed (e.g., $< 1\%$ for some algorithms), suggesting sensor fusion is inherently secure against naive attacks.
- We introduce a novel class of LiDAR spoofing attacks on AVs, the *frustum attack*, and experimentally validate frustum attack feasibility with existing hardware.
- We perform a thorough analysis of LiDAR-only and camera-LiDAR perception and show the frustum attack's first-of-a-kind ability to compromise 8 high-performing perception algorithms across 3 LiDAR-only and 3 camera-LiDAR fusion architectures. We also show that the *frustum attack is stealthy even against existing defenses of LiDAR spoofing*.
- We perform longitudinal studies of security against perception attacks. We show that, on an end-to-end AV driving software, by using frustum attacks to fool the AV's tracking and control, the attacker has high levels of attack success attacking at short and long range, expanding on previous short range attack cases.

2 Background and Related Work

2.1 Perception

AVs interact in complex environments with active agents and dynamic weather and terrain situations. To accomplish desired tasks while retaining consistent situational awareness, deployed AVs are equipped with multiple sensors of multiple modalities as well as with perception algorithms to translate sensor data into meaningful semantic information (e.g., vehicle tracking for situational awareness).

2.1.1 Camera and LiDAR Sensing

AVs are equipped with multiple cameras spaced around the vehicle. Individual cameras provide monocular vision which resolve azimuth and elevation angles to targets. Cameras are inexpensive compared to LiDAR and radar and thus are the preferred sensing modality for many AVs [3, 5, 6, 22, 23].

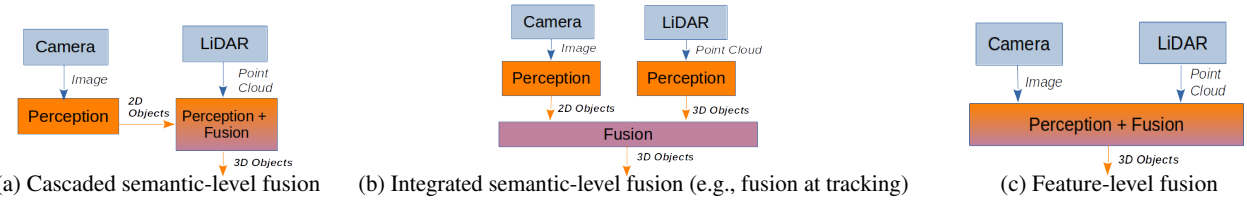
A central scanning LiDAR is commonly mounted on the roof of AVs for maximum viewing opportunity. LiDAR is complementary to the camera; it is an active sensor that sends primarily infrared light and constructs transmit-receive time differences to resolve the full 3D position of point returns [3]. LiDAR has demonstrated enhanced robustness compared to cameras in situations including adverse weather [24].

2.1.2 AV Benchmarks

We use KITTI [22] and the LGSVL Simulator [21] to test our algorithms and attacks. KITTI is composed of synchronized camera and LiDAR captures with ground truth 2D and 3D bounding boxes. We use perception algorithms with publicly available models pretrained on KITTI (see Sec. 2.1.3) as well as Baidu Apollo's open source end-to-end AV stack [6].

2.1.3 Perception Algorithms

Recently, novel DNN-based methods have been proposed for processing point cloud data from LiDAR. Three general



(a) Cascaded semantic-level fusion (b) Integrated semantic-level fusion (e.g., fusion at tracking) (c) Feature-level fusion

Figure 1: The three widely-used classes of multi-sensor fusion in perception.

Table 1: LiDAR-only and camera-LiDAR fusion perception algorithms of multiple architectures are all evaluated under the naive and frustum attacks and with defenses.

Algorithm	Type	Architecture
PointPillars [27]	LiDAR	Voxel-Based
PointRCNN [28]	LiDAR	Point-Based
PIXOR [33]	LiDAR	BEV
Frust.-ConvNet [30]	Camera-LiDAR	Casc. Semantic
Frust.-PointNet [29]	Camera-LiDAR	Casc. Semantic
AVOD [31]	Camera-LiDAR	Feature-Level
EPNET [32]	Camera-LiDAR	Feature-Level
Baidu Apollo [6]	Camera-LiDAR	Integ. Semantic

classes of LiDAR-only perception were reported in [11] and include the bird's-eye view (BEV) (e.g., [6, 25]), voxelization of the 3D space (e.g., [26, 27]), and direct ingesting of points (e.g., [28]). These early works focused on single-sensor perception pipelines without considering multi-sensor fusion.

We consider three broad classes of multi-sensor fusion for perception, illustrated in Fig. 1: (1) cascaded semantic fusion (e.g., [29, 30]) using the output of perception on one or more sensors to augment the input of other single-sensor perception, (2) integrated semantic fusion (e.g., [6]) that runs isolated perception for each sensor and fuses semantic outputs (e.g., in tracking), and (3) feature-level fusion (e.g., [31, 32]) that combines low-level (machine-learned) features from multiple perception sources to produce a unified output. Specifically, we analyze state-of-the-art LiDAR-only (PointPillars [27], PointRCNN [28], PIXOR [33]) and camera-LiDAR fusion (Frustum-ConvNet [30], Frustum-PoinNet [29], AVOD [31], EPNET [32], Baidu Apollo [6]) perception algorithms, as summarized in Table 1.

2.2 Attacks on Perception

Attacks on camera-based perception. Camera-based perception algorithms that use DNN models have been shown vulnerable to black-box attacks (e.g., [34]). Attacks on camera-based perception have been extended to AV-specific contexts [9, 35], showing that object detection and classification are vulnerable when using only camera data.

Demonstrations of LiDAR spoofing attacks. Recently, [10, 11, 14–16] have demonstrated feasibility of LiDAR spoofing devices. A relay system where LiDAR pulses were received by a photodiode and relayed through an attack laser was introduced in [14]; the system was expanded to control the 3D positioning of spoof points with a delay component [15], capitalizing on the regular patterning of LiDAR emissions. With this foundation, [10] established a 60 point stable spoofing baseline on a per-frame basis, subsequently improved to 200 points [11].

Attacks on point cloud detection. Spoofing attacks have motivated security studies of LiDAR-based perception. The placement of spoof points is considered as a white-box optimization problem in [10]. In [11], black-box attacks are introduced, exploiting that DNNs may not encode causality about the data (e.g., occluded objects). To date, only mild success is seen in obtaining FPs with spoofed points from a real laser due to engineering limitations [10, 11]; thus, many security studies use simulated spoofing models [10, 11, 18] while engineering is improved [16].

Further, [17] develops physical adversarial objects capable of compromising sensor fusion using gradient-based shape and texture optimization. The model is an expansion on single-sensor adversarial objects, as both camera and LiDAR perception model gradients are used to update shape of the adversarial object. Physical-adversarial-object approaches, such as [17], require white-box access to the deployed or highly representative perception models for training offline. Additionally, [12, 13] introduce attacks with adversarial patches and physical objects that are optimized for color, shape, and texture. Each attack performs optimization over training data [12, 13]. It has not been studied whether these attacks can generalize across perception algorithms.

2.3 Perception Defenses

Several defenses have been proposed to counteract LiDAR spoofing attacks, including model-agnostic defenses independent of the perception model (e.g., CARLO [11], ShadowCatcher [20]) and model-based defenses that fortify the perception architecture (e.g., SVF [11], LIFE [18]).

CARLO [11] is a detection-centric defense, guarding LiDAR-only perception against naive spoofing in front-near positions. The exploits the intuition that, if there are many LiDAR points appearing to pass through a detected object, the object is likely a false positive (FP). **ShadowCatcher** [20] is a detection-centric defense and uses a similar line of reasoning to CARLO: if a detection has a highly anomalous shadow region – defined as a high anomaly score using features of the shadow region – it is likely an FP. **SVF** [11] is a model-based defense and guards LiDAR-only perception against naive spoofing by augmenting LiDAR data with a point-wise

confidence score from the front-view (FV) under the intuition that naive FPs do not maintain FV consistency. **LIFE** [18] is a hybrid model-based detection-centric perception defense that compares LiDAR and camera data detections and raw sensor data. To cross-check sensor detections, the object matching method compares camera and LiDAR detections in the front view. To compare raw sensor data, the corresponding point method checks consistency of camera feature points with raw LiDAR data in a depth image, and the sensor reliability evaluation uses machine-learned prediction algorithms to compare predicted and captured sensor data. LIFE was tested against naive spoofing attacks using LiDAR and stereo imagery [18].

Sensor Fusion. The use of multi-sensor fusion to enhance perception resiliency has been suggested [10, 11, 18, 19, 36]. Yet, no systematic evaluation of sensor fusion under spoofing attacks has been performed (e.g., **LIFE** [18] was evaluated using naive spoofing without analysing spoofing performance, [17] used optimized physical adversarial objects as threat model). Thus, in this work we thoroughly evaluate the fusion models' performance under spoofing attacks.

3 Attack Objectives and Threat Model

We use the following terminology in describing the attacker goals, capabilities, and strategy. By the ***victim*** vehicle, we refer to the AV running perception algorithms. The attack's goal is to cause adverse outcomes for the victim. The attacker may wish to orchestrate attacks in some relation to an object in the scene (e.g., another vehicle) other than the victim. This object is referred to as a ***target*** vehicle. Any other vehicle or object in the scene is denoted as ***other***.

3.1 Attack Goals

We consider false positive (FP) and false negative (FN) attack outcomes consistent with the literature [10–12, 17], as well as *translation attack outcomes* where a detected object's bounding box is translated (i.e., moved) by some distance.

The goal of achieving a ***false positive outcome*** is to force the victim to perform dangerous maneuvers (e.g., emergency braking or lane change) to avoid the false object. For example, LiDAR spoofing attacks can result in safety-critical incidents, as shown with Baidu's Apollo [6, 11].

The goal of achieving a ***false negative outcome*** is to remove an existing object from the perception output such that path planning and control are compromised. Such attacks can have the devastating consequence of the victim crashing into an unsuspecting object hidden to perception (e.g., as in [17]).

Translation outcome. We find FP and FN outcomes are insufficient to fully capture the effects of perception attacks. Some cascaded semantic fusion architectures (e.g., FPN) enforce one-to-one matching between 2D and 3D detections; thus, an FP necessarily implies an FN. We call such instances *translation outcomes* as the attacker has created physical distance between the negated ground truth (FN) and the spoofed detection (FP). Translation outcomes may cause emergency braking if objects are moved to front-near positions or collision when moved farther from the victim or to a different lane.

Table 2: Gaussian moments for sampling spoof point positions relative to the desired FP location. Coordinate frame is local-level Cartesian, axes are frustum-relative (forward is toward the victim).

Direction	Forward	Left	Up
Mean (m)	1.0	0	1.0
Std. Dev. (m)	0.1	0.5	0.2

3.2 Threat Model

3.2.1 Environment

We consider scenarios where the victim AV may have multiple sensors; i.e., we consider both LiDAR-only and camera-LiDAR perception models, widely used in AVs [1, 5, 6, 23].

3.2.2 Attacker Capability

We assume the attacker has no access to the AV's internal processing, has no way to attack the camera, and can only inject signal along the same physical channels as normal LiDAR. The attacker uses a LiDAR spoofing attack similar to [10, 11, 14–16], which established how to control the 3D positioning of LiDAR points using a relay and delay system. Further, we follow the threat model from [11] which demonstrated injecting up to 200 spoof points. While [10, 11] assume high-precision spoofing where LiDAR points are placed in well-crafted patterns (e.g., outline of a car), we also relax this assumption in some cases by allowing the attacker to place points by randomly sampling a distribution; the parameters of this distribution are summarized in Table 2. This simplifies the attack design compared to the model from [10, 11] and may be more representative of a noisy attack laser.

3.2.3 Attack Strategy

In this work, we consider the following attack strategies.

Naive attacks. In general, naive attacks compromise a single sensor without regard for consistency between multiple sensors or the environment. Naive LiDAR spoofing attacks against AVs were first proposed in [10] and followed up with [11]. Naive spoofing attacks are examined in Section 4.

Frustum attacks. We introduce the novel *frustum attack* which retains consistency across multiple sensors even only attacking a single sensor. It is motivated by the fact that 2D detections of a target vehicle from the victim camera's front-view cannot resolve range, and thus the 3D uncertainty of a 2D (camera) detection defines a *frustum* from the camera image plane in the direction of the target vehicle (see Fig. 2). Attacking within the frustum of a target vehicle retains consistency with semantic and feature information between camera and LiDAR data. Frustum attacks are examined in Section 5.

3.2.4 Attacker Knowledge

System. In all cases, the attacker requires no knowledge of the underpinnings of perception, including the machine learning model and perception architecture. Further, to instantiate

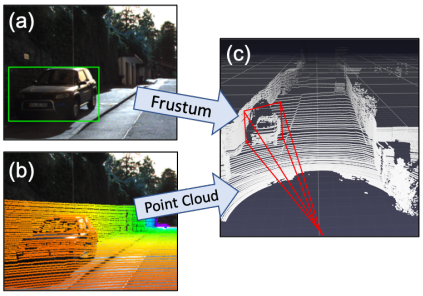


Figure 2: The frustum attack leverages that the camera is only a 2D projection of 3D space. Any feature or detection in a single 2D image could be resolved to any distance along the range axis. Thus, spoofing within the frustum retains consistency between camera and LiDAR data association. The frustum is defined by an object's 2D bounding box when extended into 3D (diagram shown is unattacked).

the attacks, the attacker need not have access to existing sensor data, other than what is required in the relay system [14, 15].

Environmental. For the frustum attacks, we assume the attacker knows the approximate position of the target object so as to obtain a frustum region for spoof point placement. In addition to FP outcomes, this also enables FN or translation attack outcomes targeting a particular (valid) object.

4 Naive LiDAR Spoofing

We first consider general black-box (i.e., naive) attacks on LiDAR-only perception. To fully evaluate the state of the art, we reproduce the LiDAR spoofing attack from [11] using patterns of occluded vehicles extracted from KITTI and sweep number of attack points in steps of 10, from 10 to 200. This is a ***naive*** method as it does not attempt to maintain consistency among sensors and is ***black-box*** as it does not require knowledge of the employed perception model or sensor data.

4.1 Spoofing Against LiDAR-Only Perception

We test a perception algorithm from each of the three categories of LiDAR-only perception architectures, consistent with [11] and outlined in Table 1. Specifically, we use voxel-based PointPillars [27], point-based PointRCNN [28], and BEV-based PIXOR [33] for 3D object detection. We reproduce the attack success rate (ASR) from [11]. Details on the reproduced results are in Appendix A.1, showing high ASR of the naive spoof attacks at front-near positions.

4.1.1 State-of-the-Art Defenses

We reproduced CARLO, SVF, and ShadowCatcher, as no source code was available; reproduced results are presented in Appendix A.2. Our results for CARLO and SVF against naive attacks are consistent with [11] – i.e., the ASR is greatly reduced in front-near positions against naive attacks. However, with realistic assumptions, we obtained lower defense performance for ShadowCatcher than reported in [20]. The reasons, outlined in Appendix A.2, include that the original

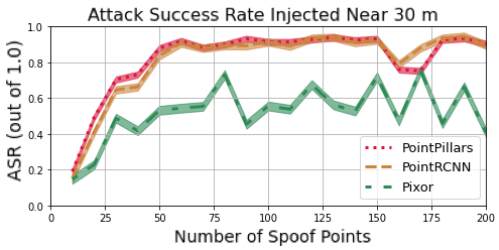


Figure 3: *Naive spoofing attacks against LiDAR-only with CARLO defense outside front-near:* CARLO is not suitable at guarding perception against naive spoofing for false positives outside of front-near; the ASR of CARLO-guarded models is nearly as high as without CARLO (see Fig. 16)

work tuned parameters on the test set as well as used the ground-truth information instead of the output of a perception algorithm; ground-truth information is not available for a real system and significantly reduces noise.

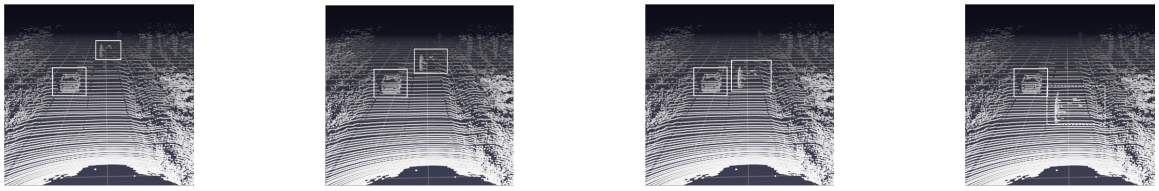
Very recently, [18] introduced LIFE defense that designed point-based and frame-based camera-LiDAR consistency checks as a preprocessing step to guard against both camera and LiDAR attacks. As reported in [18], LIFE is well-suited to detect naive spoofing attacks, as naive spoofing does not retain consistency between the camera and LiDAR data.

4.1.2 Some Existing Defenses Have Vulnerabilities

Under further scrutiny of existing defenses, we find several naive attack configurations not tested in [11] that suggest the CARLO defense introduces additional vulnerabilities.

CARLO Vulnerability to False Positives. While CARLO demonstrates high success guarding against naive spoofing in front-near [11], *naive spoofing is stealthy to CARLO when placed outside of front near*. Intuitively, as the range to spoofed objects *increases*, the angle subtended by the frustum towards the detection *decreases*. This leads to a decrease in the number of LiDAR points contained in the frustum as the emitted LiDAR points spread in a spherical pattern (constant angular density). Thus, an increase in range leads to spoofed instances that appear more similar to normal instances under the CARLO hypothesis. We show this by spoofing points at a range of 30 m from the victim; our results in Fig. 3 show that CARLO is incapable of guarding against these naive spoofing attacks, as the ASR is on-par with the defense-less system (compare Figs. 3 and 17). Analysis for additional spoofed point distances is provided in Appendix B.

Spoofing attacks applied at greater ranges *can* have severe adverse outcomes when exercised longitudinally (i.e., over multiple time steps). For example, since LiDAR-only perception cannot be guarded by CARLO outside of front-near, as shown in Fig. 4 an adversary can create false positives over multiple time steps to give the appearance of a vehicle moving directly toward the victim with high velocity. This may trigger braking and collision avoidance maneuvers even before the false vehicle reaches close range. Detailed investigation of longitudinal attacks is provided in Section 6.



(a) Stealthy FP at 40 m (b) Stealthy FP at 30 m (c) Stealthy FP at 20 m (d) FP invalidated at 10 m

Figure 4: Even with the CARLO defense, a spoofing scenario starting at long-range can evade the defense for many frames until it reaches front-near position. During this time, the AV will build a (adversarial) track on the spoofed object, which can cause adverse control outcomes (e.g., collision avoidance maneuver).

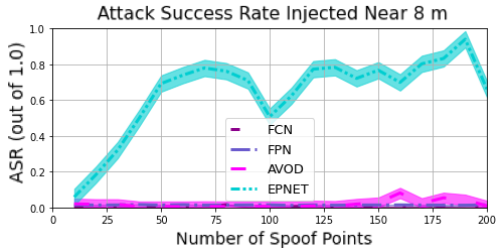


Figure 5: *Naive spoofing attacks against camera-LiDAR fusion:* Many fusion algorithms (including FPN, FCN, AVOD) have high inherent resiliency to naive spoofing attacks ($< 5\%$ ASR) without specialized defenses. EPNET's vulnerability is due to its high baseline false positive rate - nearly 40% of all EPNET's detections are FPs, even without attack.

CARLO Vulnerability to False Negatives. We also find that CARLO, even at front-near, is vulnerable to FN invalidation attacks. Since CARLO relies on physics violations – i.e., many points appearing to "pass through" a detected object, which should not occur for normal objects – the attacker can instead use the spoofer to create these physics violations on *normal instances* to obtain FNs (i.e., invalidation of true objects). Specifically, an adversary can spoof points behind valid objects (in no particular spoofing pattern, unlike the attacks required by [10, 11]) which will trigger CARLO into believing the detected object is invalid. Thus, a true object will be rendered as a false negative, potentially causing a head-on collision of the victim vehicle.

The FN outcomes against CARLO depend on the range to the target to invalidate. This follows from the decrease in the frustum angle with range and the constant angular density of LiDAR points. Using a 200-point maximum capability and requiring only random injections, attacks achieve upwards of 40% success at invalidating objects 50 m away (Appendix B).

4.2 Sensor Fusion

No prior work has systematically evaluated whether sensor fusion is more resilient to spoofing. Thus, we evaluate the naive spoofing attacks against multiple camera-LiDAR fusion algorithms across multiple architectures summarized in Table 1. We find the majority of tested sensor fusion are inherently resilient to the naive spoofing attacks (Fig. 5). Overall, this level of intrinsic defense renders naive spoofing attacks ineffective even without the addition of specialized defenses. Specifically, widely used camera-LiDAR fusion algorithms FPN [29], FCN [30], and AVOD [31] have high resiliency with ASR generally lower than 5%. We find EPNET [32] is still vulnerable; we believe this is due to EPNET's high baseline FP rate. On the (unattacked) KITTI validation set, EPNET has 220% the number of FPs compared to AVOD; nearly 40% of all EPNET's detections on (unattacked) KITTI are FPs.

Summary: Impact of Black-box Attacks. LiDAR-only perception alone is vulnerable to naive black-box spoofing attacks in front-near positions, as previously reported [10, 11]. However, there are several promising specialized defenses, although CARLO is insufficient in preventing black box spoofing attacks outside front-near positions and is vulnerable to invalidation attacks. Finally, we showed that sensor fusion is intrinsically more robust to naive attacks. Yet, in what follows, we demonstrate that the perception models and defenses perform poorly under a new class of attacks: the *frustum attacks*.

5 Frustum Attack on LiDAR

In this section, after establishing the feasibility of the frustum attack, we evaluate the impact of frustum spoofing on modern perception methods. We show both LiDAR-only and camera-LiDAR fusion perception are widely vulnerable to the context-aware frustum attacks: *all 8 tested algorithms falling across 6 different architectures from Table 1 are vulnerable* and none of the state-of-the-art defenses against LiDAR spoofing are capable of defending against the frustum attack.

5.1 Frustum Attack Motivation & Definition

While naive spoofing is damaging against LiDAR-only perception, it does not maintain consistency with physical invariants or between camera and LiDAR data; as shown in Section 4, this inconsistency can be leveraged to filter out the naive spoofing attacks. Consequently, the *frustum attack* is conceived as a black-box method of retaining consistency between the camera and LiDAR data *and* consistency with physical invariants using easily obtained contextualizing information from the environment. Specifically, an adversary can leverage that the camera is only a 2D projection of the 3D space; any detection or feature in the camera can be resolved to any point along the line extending from the camera out to infinite range (in practice, ~ 100 m for AV applications) because a single camera cannot resolve range information.

The frustum attack thus places spoof points to leverage the projective nature of the camera. Points are placed behind or in front of existing objects so that they have front-view consistency. This can be realized by spoofing within a pyramid (i.e., the *frustum*) where the tip of the pyramid is at the victim sensor and the base is the projection of the 2D, front-view bounding box of a true object out along the range axis.

Thus, due to the projection and by spoofing in-view of existing (target) objects, perception algorithms may associate (unattacked) features/detections in the camera and the spoofed LiDAR points even if the spoof LiDAR points are at a *different* range than the target object. By spoofing in the frustum of valid objects, frustum attack FPs maintain many natural qualities of normal objects (see Fig. 8), helping them to be stealthy against existing defenses relying on physical invariants or camera-LiDAR consistency checks.

We denote this 'in-view' spoofing as the ***frustum attack*** since a 2D bounding box around an object in the camera's image defines a frustum when the uncertainty of the 2D box is extended into 3D along the range axis, as illustrated in Fig. 2; also, in the bird's eye view (BEV) in Fig. 8. Importantly, the adversary needs to only approximately know the frustum.

5.2 Attack Feasibility and Practicality

Feasibility of naive spoofing attacks has been shown in [10, 11]. Here, we provide experimental justification for the frustum attack. We first describe five situations that naturally arise in nearly all day-to-day driving conditions that enable frustum-based spoofing. We then demonstrate one scenario of the frustum attack experimentally. Three of the situations are attainable with current engineering/LiDAR technology. Work is underway to advance optics and tracking which may enable additional spoofing scenarios (e.g., see [16]).

5.2.1 Frustum-Attack Spoofing Scenarios

We describe five common scenarios where the frustum attack can be exercised; the scenarios are illustrated in Fig. 22 in Appendix C. In all cases, the spoofer has full control over the range of placement of the spoof points along the frustum by increasing or decreasing the delay timing.

S1: Spoofer on target. A spoofer is placed on a target car and aimed at victim (the target AV owner/passengers may or may not be aware of this). The target car does not have to be endangered for this to have impact because the attacker creates FPs that cause the victim to perform evasive maneuvers. The target car is by definition in line with its own frustum. Any spoofed points along the line-of-sight (LOS) between the target and the victim will remain in the frustum.

S2: Spoofer on other vehicle in line. Spoofer is placed on a non-target car on the line defined by the victim AV and target vehicle. This scenario arises often in natural driving, as a lane, which is usually locally straight, helps cars stay in line with each other, and thus in each others frustums.

S3: Spoofer on other vehicle not in line. A fully general spoofing attack could take place out-of-line. However, executing this attack outside the frustum is not currently feasible and requires more precise aiming of the laser than has been demonstrated. Engineering advances will enable this scenario, and work is already underway in this area [16].

S4: Spoofer on environment in line. A spoofer is placed in the environment in line with a lane. Examples include placing the spoofer on a bridge transverse to the road or on low-lying traffic signs, tree limbs, etc.

S5: Spoofer on environment not in line. This resembles **S3** with similar feasibility constraints but with a spoofer placed on a static object in the environment (e.g., road-side sign).

5.2.2 Feasibility Demonstration

We adopt the physical hardware from [10, 11] and use a VLP-16 PUCK for the LiDAR sensor and for the spoofing system, an OSRAM SFH 213 FA photodiode, an OSRAM SPL PL90 attack laser, and an additional lens for beam focusing. The VLP-16 is a rotating LiDAR scanner providing full 360° azimuth coverage and is compatible with many modern industry AVs and perception, including LGSVL [21] and Baidu Apollo [6], which have LiDAR plugins for the VLP-16.

To test frustum attack feasibility, the spoofing device is placed behind the target vehicle (Fig. 6). The spoofer has just enough visibility above the target car for the attack laser to have LOS to the LiDAR sensor. This is easily realized in everyday driving so long as the attacking vehicle is slightly larger than the victim or the spoofing device is elevated (e.g., placed on the roof of AVs, like existing LiDAR).

We find the spoofer can command the delay timing to inject spoof clusters at varying distances relative to the target car. In Fig. 7, spoof point clusters are moved successively farther from the target car (or closer, if run in reverse) in a dynamic environment with longitudinal consistency. We also repeat the experiment with a moving target car but stationary victim and spoofer; for conciseness, those scenarios (including videos) are only available online, along with the project code, at [37].

Discussion. The above experiments cover two situations: (1) victim, target, and spoofer are in-line and (relatively) static which encompasses **S1** and **S2** for vehicles traveling in unison (e.g., vehicle platooning), and (2) target moving relative to spoofer which encompasses the same prior situations (this time with relative motion) as well as **S4** due to the relative velocity between the vehicles. The outcomes of these spoofing experiments validate that common, everyday spoofing scenarios are feasible even with existing spoofing hardware, although executing the frustum attack with motion of the spoofer and target has not been fully demonstrated.

In fact, a frustum attack only requires LOS between the spoofing device and the victim AV with at least one object in the scene. The victim and target vehicles always define a frustum, so it is up to the attacker to position the spoof points within the frustum; this is trivially satisfied when the spoofer is in the same lane as the vehicles (e.g., on another car) or may simply require a lane change or velocity adjustment. With improvements in laser aiming, the number of natural frustum attack scenarios will only increase [16].

Figure 6: A spoofer launches a malicious *frustum attack* against a victim AV using a target car. Spoof points are placed at any distance within the frustum behind the target car to obtain false positive, false negative, and/or translation outcomes.

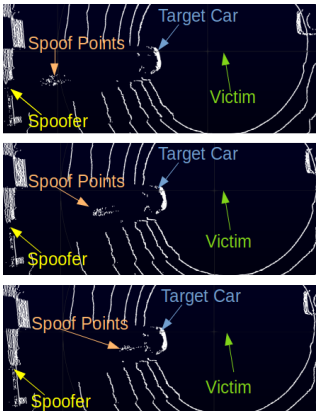


Figure 7: A physical experiment demonstrates that an attacker can stably spoof longitudinally consistent points in the frustum of a target vehicle and has control over the placement distance. We run a continuous scenario where spoof points are moved successively towards and away from the target, and show three select frames here. Running FPN as perception would cause a *translation outcome* where the target car would be detected at the spoof point location. Full video online [37].

Finally, if started near the target vehicle, moving spoof point clusters, as in videos online [37], can create longitudinally-consistent frustum attacks that drift the track further away (or closer to the victim, if run in reverse), as described in Section 6.1. Case studies of the related tracking and end-to-end control outcomes are presented in Sections 6.1 and 6.2.

5.3 Frustum Attack Performance Analysis

5.3.1 Experimental Methods

We run a large-scale experiment on over 75 million scenarios to assess the vulnerability of perception to the frustum attack for FP, FN, and translation outcomes.

For each of the first 7 perception algorithms in Table 1, we select each valid vehicle in each frame of the KITTI validation set. Each vehicle becomes the "target vehicle". For our analysis, we discard any valid vehicles not detected by the unattacked perception algorithm, as this would artificially inflate the FN attack success metric. For each valid vehicle, we simulate frustum attack spoofing using different combinations of the number of spoof points n_i and the relative distance of placement d_i, i.e., (n_i, d_i), within $n_i \in [2, 200]$ points and $d_i \in [r_0 - 10, r_0 + 30]$ m; here, r_0 is the original range to the target. The experiment captures existence of spoofing-induced: (a) FP at the spoof location, and (b) FN of the target object.

This experiment yielded on average 11 million attack traces for each perception algorithm with a total of over 75 million attack traces for the frustum attack. We also assess the four aforementioned (three experimentally, one in discussion) defenses for each perception algorithm. Due to the combinatorial nature of such evaluation (algorithm $\times$ points $\times$ distance $\times$ defense), we sample a set of attack traces over a coarse grid of parameters for each tested perception algorithm.

Example outcome. An example successful frustum attack against FPN fusion is in Fig. 8, where 20 spoofed points are placed in a random pattern with a mean location 7 m behind and within the frustum of a target valid object. We find that, as long as spoof points are within the frustum, it is less important how precise those points are placed. In fact, we find in general that spoofing using a normal distribution of points with moments specified in Table 2 can achieve performance on-par with extracting occluded traces from KITTI as done in [11] (see Appendix D for detailed comparison).

In this case, the target object is composed of 238 points, an order of magnitude more than the spoofed points, and is at 25 m range from the victim. As shown in Fig. 8, even only attacking LiDAR, the frustum attack is successful in obtaining an FP at the spoof point cluster.

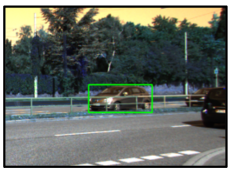
5.3.2 Results I: Frustum Attacks Compromise All Perception Algorithms

We now show that the frustum attack is capable of not only compromising LiDAR-only perception but also compromising camera-LiDAR fusion. Also, the frustum attack succeeds across multiple architectures of both LiDAR-only and camera-LiDAR perception. Here, we describe the main observed results. Additional results are presented in Appendix E.

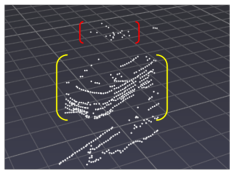
Attackability: Attack Existence

A frustum is "attackable" if there is at least one combination (n_i, d_i) within the established attacker capability that is successful in generating an FP near the spoof points (or FN of the targeted object, depending on attacker's goal). Given a fixed set of input sensor data, the vulnerability of a perception algorithm depends on how many target objects are attackable.

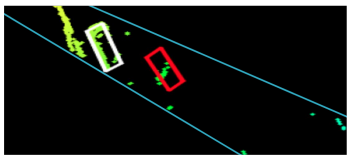
Fig. 9 illustrates the vulnerability of each perception algorithm by presenting the fraction of target objects that are attackable. Presented are both FP (Fig. 9a) and FN (Fig. 9b) outcomes to comprehensively illustrate the vulnerability. Considering FP outcomes, at middle ranges to target objects (i.e.,



(a) Clean RGB detection

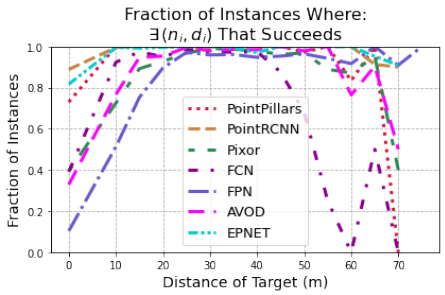


(b) Frustum with attack

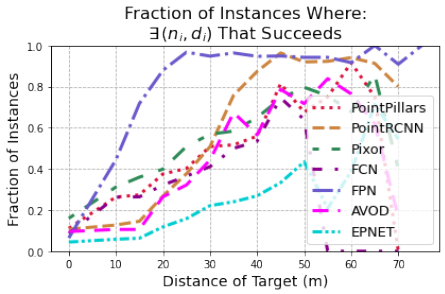


(c) BEV shows FPN detects inserted points, not original points.

Figure 8: (a) 2D detection yields (b) a 3D frustum of points. Injecting just 20 points in random pattern (bracketed red) deceives 3D object detection, even against a valid object (bracketed yellow) of 238 points; (c) BEV projection of the 3D detection show success of the frustum attack with translation outcomes, as the FP detection (red box) is far from the FN ground truth (white box).



(a) For FP: All algorithms are highly attackable for FP outcomes, particularly when the target objects are at $15-60$ m range – here, the attackability is near 100% across the board (except FCN's dip).



(b) For FN: Perception demonstrates a surprising vulnerability to FN outcomes under LiDAR spoofing. The targeted object can be negated (i.e., not detected) for all perception algorithms, even under a small, spoofing frustum attack model.

Figure 9: Percentage of instances in the KITTI dataset (over number of points and distance of placement) where there exists a successful (a) FP, and (b) FN frustum attack; all perception algorithms show widespread vulnerability to both (a) FP and (b) FN outcomes under the frustum attack.

$15-40$ m) *nearly 100% of instances using any of the perception algorithms are attackable*, showing the widespread vulnerability to frustum attacks. In fact, except for FCN which has a dip in attackability from $40-60$ m, this near-100% attackability extends for all other perception algorithms from $15-60$ m, which is a devastating outcome for AV perception.

Similarly, we show a surprisingly high degree of FN vulnerability (Fig. 9b) even after discarding targeted vehicles not detected without attack. At a 35 m range, with a suitable selection of spoof distance, half of all vehicles can be negated with a frustum spoofing attack for all algorithms except EPNET.

Due to space constraints, in the rest of the work, we focus on analysis of FP outcomes which we find are more successful, repeatable, and adaptable to different spoofing distances compared to FN outcomes.

Attack Success Across Number of Spoofed Points

Attack success depends on the number of spoofed points, up to a certain point of convergence. Here, we look at two key indicators of spoofing success: 1) the rate of attack success across the range to target objects for discrete numbers of spoof points, and 2) the minimum attacker requirements for successful attacks in general.

Attack success by number of points. Fig. 10 presents how attack success depends on the discrete numbers of spoofed points. Surprisingly, even spoofing just 2 points may be enough to obtain FP outcomes at the site of spoofed point placement given an optimal selection of spoof point distance. The attack success quickly converges to a rate similar to the one in Fig. 9 at just 60 spoof points.

Minimum attack requirement. In general, more spoof points yields higher ASR. However, since an attacker only needs a few successful attacks to cause devastating outcomes and may not have the ability to spoof large numbers (e.g., hundreds) of spoof points, it is important to understand the average smallest number of points needed for a successful attack. To compute this estimate of the 0th order statistic of spoofing, for each perception algorithm, if the target vehicle were attackable, we logged the range to that target object, r_0, and stored the smallest number of points, $n_{i,\min}$, where an attack succeeded, marginalizing over distance, d_i. We then computed the mean of this collection of minima against range to the target object (see Fig. 11) and find that only tens of points are needed on average. Note that these results can be interpreted as a measure of robustness of the perception algorithm to small numbers of spoof points; e.g., FPN is significantly more robust at intermediate ranges, FCN, PIXOR, and AVOD are more robust than PointPillars, PointRCNN, and EPNET.

Attack Success Across Range to Target Vehicle

The location of the target car is an important element in the success of a frustum attack. For different ranges to target vehicles, Fig. 12 breaks down the success against AVOD as a function of both the placement of the spoof points relative to the target vehicle and the number of spoof points and does not marginalize over parameters. This highlights that spoofing

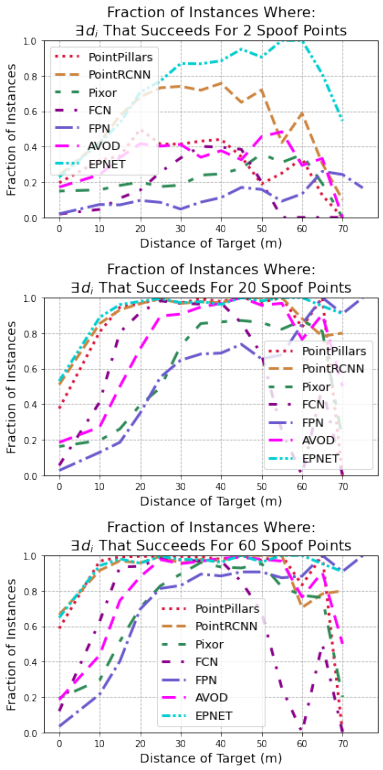


Figure 10: Percentage of instances in the KITTI dataset (over the placement distance) where there exists a successful FP attack for different numbers of spoof point: the frustum attacks are successful across a wide range of spoof point numbers.

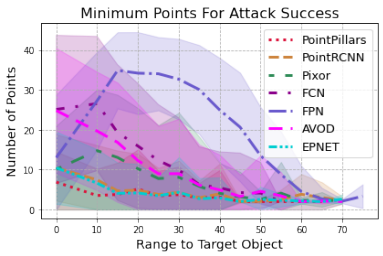


Figure 11: For each perception algorithm, the mean of the smallest number of spoof points for a successful FP attack, marginalized over the relative spoof placement, is low. Generally, the mean smallest set is less than 20 for nearly all algorithms for all ranges to target vehicles.

attacks are generally more successful as the range to target vehicles increases. A similar pattern is observed across all tested perception algorithms, as shown in Fig. 25 in Appendix E.

5.3.3 Results II: Frustum Attacks Compromise Defenses

We show that in addition to being effective against both LiDAR-only and camera-LiDAR fusion, the frustum attack is stealthy to the aforementioned defenses.

We collected a sample of attack traces using each pairwise combination of spoof points in $n_i \in \{10, 60, 100, 200\}$ and attack distance $d_i \in r_0 + \{5, 9, 12, 16\}$ m and run each combination for 200 frames of data for each algorithm, totaling nearly 25,000 attack traces per each defense. We observed that stealthiness to the defenses as a function of the parameters is nearly constant; thus, aggregated results across parameters are summarized in Tables 3, 4, 5. We report the fraction of frustum attacks that are still successful after applying the defense as the "% Stealthy", and, where relevant, the fraction of erroneously invalidated valid objects as the "Induced FN Rate".

Table 3: Nearly all frustum attacks against both fusion (left) and LiDAR-only (right) are stealthy to CARLO defense

Algorithm	% Stealthy	Algorithm	% Stealthy
FCN	100%	PointPillars	100%
FPN	99.76%	PointRCNN	99.9%
AVOD	100%	PIXOR	92.3%
EPNET	99.9%		

Table 4: Frustum attack is stealthy to SVF defense

Algorithm	% Stealthy
SVF-PointPillars	90.3%

Table 5: ShadowCatcher fails to detect a significant number of frustum attacks and has too high induced FN rate.

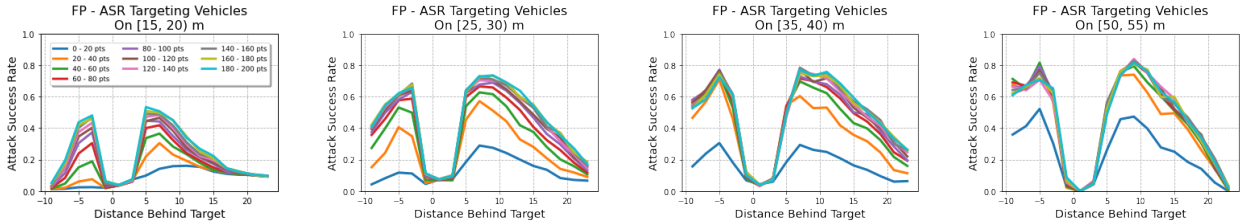
Algorithm	% Stealthy	% Induced FN Rate
FCN	80.7%	70.3%
FPN	57.8%	96.9%
AVOD	84.9%	72.9%
EPNET	90.5%	64.4%
Algorithm	**% Stealthy**	**% Induced FN Rate**
PointPillars	91.0%	68.0%
PointRCNN	89.3%	67.1%
PIXOR	81.5%	42.9%

CARLO: The frustum attack against all perception algorithms is *nearly completely stealthy* to the CARLO defense since attacks are placed in the frustum and few LiDAR points travel through the spoofed object; see results in Table 3.

SVF: The frustum attack is stealthy to the SVF defense since frustum spoofs are consistent with information from the front-view projection; see results in Table 4.

Shadow-Catcher: ShadowCatcher does not perform well at detecting the frustum attack or identifying normal objects as valid, as confirmed by results in Table 5. Our results show an unacceptably high induced FN rate (i.e., it invalidates true objects at too high of a rate).

LIFE: LIFE is designed to identify faults, miscalibrations, and attacks against AVs equipped with panoptic stereo cameras and a central, wide-angle or scanning LiDAR sensor using an Object Matching Method (OMM), a corresponding point method (CPM), and a sensor reliability evaluation (SREM) [18]. However, each of these components are ill-posed for detecting the frustum attack, as noted even by



(a) FP ASR on vehicles [15, 20)m (b) FP ASR on vehicles [25, 30)m (c) FP ASR on vehicles [35, 40)m (d) FP ASR on vehicles [50, 55)m

Figure 12: ASR when AVOD is used as a function of the spoof points' distance (relative to the target vehicle). Attacks are more successful at increased range of the target ((a) vs. (b), (c), (d)). Horizontal axis represents relative placement of spoof points; each line represents a different number of spoof points from $0-200$. Number of points determines attack success up to a steady state where additional points provide marginal benefit. High FPs are seen spoofing both in front (-x axis) and behind (+x) the target.

the authors in [18]; specifically, Section 8.4.1 of [18] states that a common failure mode is when "*most injected fake echoes/points are behind or very near existing aboveground objects...the induced fake objects cannot be detected.*" Specifically, OMM fails to detect the frustum attack because it uses a projection of LiDAR onto the 2D image plane to check consistency between 2D image and 2D LiDAR – the frustum attack is designed to retain consistency for this very purpose. Second, CPM fails to detect the frustum attack because it generates a small set of 3D features from the camera, *then* checks for a corresponding LiDAR point. Thus, CPM *cannot detect the frustum attack* as it maintains consistency with the camera data and is placed in sparse regions where no checking will occur. Finally, SREM projects LiDAR to the image plane and compares the 2D camera and 2D (front-view) projected LiDAR where the frustum attack is designed to be consistent.

5.3.4 Security Implications

The presented results establish that the *frustum attack* is successful in compromising both LiDAR-only perception as well as camera-LiDAR fusion, whereas existing state-of-the-art defenses against LiDAR spoofing are ineffective against the frustum attack. Consequently, *existing perception algorithms are not secure against LiDAR spoofing* when additional contextual information is available for identifying frustums.

6 Longitudinal Case Studies

Isolated instances of spurious attacks on perception will not survive against real AVs with multiple sensors capturing data over time. With map-aided tracking, AVs can flag FPs that do not comply with semantic map or dynamics information. Tracking also builds resiliency to isolated FNs by allowing for coast time in between measurements [38].

The frustum attack, with robustness to number of points and distance of injections, as well as success against multiple algorithms and random spoof patterns, is suitable for temporally consistent spoofing to achieve impact at the tracking level (i.e., over time). The physical spoofing experiments from Section 5.2 and linked videos [37] show longitudinal frustum attacks where a spoofer can gradually adjust the position of spoof points to simulate motion of a spoofed object. We provide additional visualizations to understand longitudinal frustum attacks in Appendix F, Fig. 24.

To confirm impact of the frustum attack on real systems using temporal fusion, we perform two evaluations. First, we explicitly analyze spoofing's impact on the multi-frame tracking algorithm and present two case studies showing that such attacks jeopardize AV safety. Second, we apply the frustum attack to an end-to-end, industry-level AV software stack, Baidu Apollo [6] using the LGSVL simulator [21] and show resulting adverse planning and control outcomes.

6.1 Frustum Attack Impact on Tracking

6.1.1 Tracking Algorithm

We implement a Kalman filter tracker with position, velocity, and acceleration states according to [38]. All major industry players, including Baidu Apollo [6], Autoware [23], and OpenPilot [39] use variations on the Kalman Filter for tracking and fusion. We use one tracker per frustum using FPN as perception, as FPN encodes a one-object-per-frustum requirement. The track (i.e., trajectory) is predicted forward using a nearly-constant acceleration model and process noise according to [40], which is consistent with industry-level AVs [6]. 3D detections from camera-LiDAR perception are fed at 5 Hz to the tracking module which tracks box centers over time. We use an industry-standard χ^2 gating between predicted tracks and timestamped measurements as tracking integrity; this ensures temporal consistency between measurements and prevents unlikely associations from updating tracks. We use the 99% threshold of the χ^2 gate, specified as $0.99 = \Pr(g_k > \tau)$, where $g_k = z^T Q^{-1} z$; here, τ is the threshold found using the χ^2 inverse CDF, z is the innovation between propagated state and measurement, and Q is the innovation covariance from the Kalman update [38]. In other words, we neither forced perception to detect our spoofed points nor did we force tracks to accept the resulting detection. We fixed attacker capability at 65 points, which is substantially less than the maximum demonstrated capability.

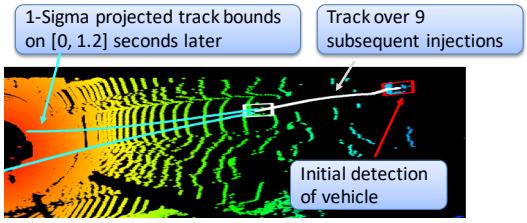


Figure 13: Attacking over multiple frames at attacker-specified distances creates adversarial tracks. Sequences of 65 point spoofs create false detections that are accepted by tracker integrity. Initial spoof detection (red) travels along (white) track with time-to-impact with the victim vehicle predicted 1 s later, with high-certainty (cyan) near impact.

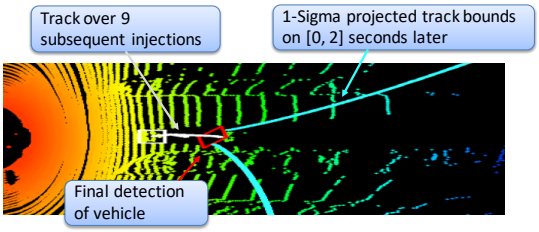


Figure 14: Attacking 5 frames using only 60 points can transform a high-confidence track of a valid vehicle into a low-confidence track with adversarial velocity away from victim. Predicted path shows target vehicle is moving away when, in reality, the target and victim vehicles have no relative velocity.

6.1.2 Scenario I: Vehicles at Intersection

We first consider an attacker creating an adversarial track on a crash course for collision with the victim. We select a scene where the target is at 35 *m* range. With traffic lanes 4 *m* wide and vehicles 5 *m* long, this scenario represents a large intersection where the cars are initially static.

Due to perception's high frame rate, the attacker need only to succeed in attacking over a short time window for a false track to be created. The attacker injects 10 sequences of point clusters behind the target, corresponding to 2 *s* of real-time, and alters the distance between successive spoofs so that the vehicle appears to accelerate towards the victim.

Fig. 13 illustrates the BEV of the false track created from this spoofing attack. Eight of the attacker's ten injections were falsely detected by perception and accepted by the χ^2 gate to update the created (adversarial) track.

For path planning, it is essential that AVs understand both the current states of nearby vehicles as well as their future trajectories in order to plan a safe path through the environment. After two seconds of attack, a path planner predicts the existing track forward, shown in Fig. 13, and the vehicle in front of the victim is on a collision course with a time-to-impact of just over 1 *s*. This can trigger dangerous, aggressive and unnecessary collision avoidance maneuvers.

6.1.3 Scenario II: Highway Adaptive Cruise Control

Here, we consider highway flow of traffic (e.g., 25 m/s) where adaptive cruise control uses perception to monitor objects and to keep up with traffic flow. We consider a likely case in which the victim AV has already achieved high-precision track on a true vehicle in front. An existing high-confidence track is more challenging for an attacker to manipulate (e.g., see [41]). In this case, any dramatic deviation in the location of that object may trigger an alarm or rejection by the χ^2 integrity monitor, particularly since perception operates at high rate.

Over just five spoofs which corresponds to 1 s of real-time, an attacker can manipulate an existing track by gradually increasing the distance of spoof points away from the target (Fig. 14). While initially the track has no relative velocity (i.e., vehicles traveling in unison), path planning updates track prediction for the vehicle in front after the frustum attack, and it appears to travel away at an accelerating rate. This will cause an increase in the adaptive cruise control speed of the victim vehicle, due to the *apparent* increased velocity of the traffic flow, when in reality, the ground truth vehicle is still traveling with no relative velocity; thus, the victim vehicle will dangerously approach the car in front.

6.2 Frustum Attack Impact At Driving-Decision Level: Baidu Apollo Study

We perform another case study using the high-fidelity physics-based simulator, LGSVL [21], and the Baidu Apollo AV stack [6]. We use LGSVL with a 32-beam LiDAR model and a Full-HD 1080p camera model to capture realistic LiDAR and camera sensor data for the frustum attack. While the physics engine of LGSVL is built on Unity and robust, the LGSVL API is under continuous development. As a result, it is unclear how to modify low-level sensor data in real-time. Thus, we took a multi-stage approach to evaluating the end-to-end consequences of the frustum attack on Apollo's control. First, we capture LiDAR and camera data during normal operation. Second, we execute the frustum attack on the captured data and run Apollo offline to get detected objects and control decisions. Finally, we replay the control data through the LGSVL bridge to observe and visualize the outcomes. We were able to use this approach as the control commands of the vehicles were matching in the first and second runs, up to the point when the victim vehicle initiates emergency braking.

The scene is set consistent with the physical experiment (Section 5.2) and following Fig. 22b (i.e., **S2**): a target car is between the victim and the spoofing adversary. Fig. 15 shows snapshots of two LiDAR captures with detected objects when running Apollo perception (left) and the control outcome observed when replayed through LGSVL (right). Initially, Apollo detects the target car, as expected. The spoofing adversary is not detected due to strong occlusion, also as expected. Part-way through the sequence, the adversary launches the frustum attack, and Apollo detects the spoofed points as a nearby object which triggers emergency braking, unnecessarily stopping (and thus endangering the victim vehicle). Full video of the playback sequence is available at [37].

Remarks. Attacks on perception must propagate into adversarial tracks to impact AVs. We have shown that frustum attacks can be exercised longitudinally to have high-impact at the tracking, decision, and control levels. An attacker can use mere seconds of real-time to create false scenarios of predicted collision or accelerate the flow of traffic. The frustum attack allows for both starting attacks at longer range and attacking in front-near. We show that this can be of great benefit for the attacker because it can create a diverse set of attacker-specified, high-confidence maneuvers.

7 Discussion and Future Work

7.1 Limitations

Datasets. We use KITTI and LGSVL to evaluate the considered LiDAR spoofing attacks. Although we generated over 75 million attack scenarios, KITTI is a small dataset and may not be fully representative of day-to-day AV driving. We use the LGSVL simulator to study Baidu Apollo, and future work will leverage high-fidelity simulators and additional open-source datasets to perform studies on frustum attack generalization.

Apollo Evasive Maneuvers. Besides our testing of the frustum attack in front of the target vehicle on Apollo (Sec. 6.2), we intended to test the frustum attack in the shadow region (i.e., behind target) since the shadow is more vulnerable to attack, as identified in Sec. 5.3.2. However, this was not attainable with Apollo's capabilities. We observed that Apollo has minimal ability to execute any evasive maneuvers, even when we aimed a target car heading straight for Apollo using ground-truth perception data. This is consistent with findings from evaluations of limitations of Apollo capabilities [42].

Optical Engineering and Dynamic Spoofing. The frustum attack is logistically possible in the scenarios described in Sec. 5.2. While these commonly occur in everyday driving, only three are technically feasible with today's technology, and only two have thus been demonstrated - those also used static spoofer and static victim. The current experiments have not shown attack feasibility when there are relative distance and angle changes between the spoofer and the victim. In all five targeted attack scenarios in Fig. 22, the victim should be moving to cause serious attack consequences. This would require the spoofing device to dynamically track and aim at the victim, and this engineering feat has not yet been fully demonstrated, with some recent progress in this direction [17].

7.2 Future Work

Shadow Vulnerability. The shadow region, a subset of the frustum behind the target object, is an important element for the frustum attack success. Future work will explore in detail the low-level behavior of the perception DNNs to illuminate why the shadow is so vulnerable, including possibilities of overfitting or intrinsic vulnerability of free space.

Defenses. We evaluated state-of-the-art defenses against LiDAR spoofing and found none are suitable to protect against the frustum attack. Future works will propose defenses capable of defending against the frustum attack.

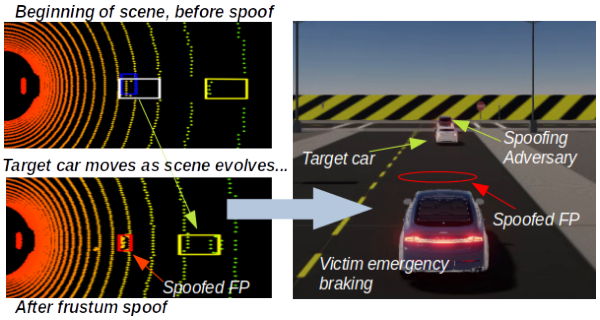


Figure 15: Frustum attack achieved on AV running Baidu Apollo software using perception data from LGSVL simulator with setup matching the physical experiment from Section 5.2. Target vehicle (white) detected throughout by image-detection pipeline. Target vehicle initially detected by LiDAR perception (blue) and followed by Apollo. When spoofing happens, Apollo detects spoofed points as new object (red); thus, unnecessarily engages emergency brakes and stops mid-lane.

Generalization. The high degree of frustum attack success and the large number of evaluations performed suggests that a single choice of attack parameters can generalize across perception algorithms. However, this belief has only been tested implicitly using consistent attack parameters in the large-scale study. Future works will explicitly consider the success of transferring specific attack traces between algorithms.

8 Conclusion

In this work, we exposed the vulnerability of LiDAR-only perception and camera-LiDAR fusion to the frustum attack: small-scale (i.e., tens of points) LiDAR spoofing in-view of existing, valid objects. We evaluated the frustum attack on three distinct LiDAR-only architectures and five models within three different architectures of camera-LiDAR fusion, including fusion at the semantic, feature, and tracking levels. Within each class, we used single-sensor and sensor fusion algorithms from top-performers on popular datasets ([27, 28, 30–33]), established benchmarks ([27–29, 31]), and algorithms representative of leading end-to-end, full-stack industry pipelines ([6, 27, 28]). We demonstrated a singular attack model capable of compromising each class perception in AVs. The attack model is black-box; furthermore, it does not require any knowledge of the perception algorithm. Such broad success with a black-box attack model illuminates a systematic vulnerability across both LiDAR-only and camera-LiDAR perception algorithms.

Acknowledgements

This work is sponsored in part by the ONR under agreement N00014-20-1-2745, AFOSR award number FA9550-19-1-0169, and NSF CNS-1652544 and CNS-2112562 awards.

References

[1] A. Hawkins, "Waymo's autonomous cars have driven 8 million miles on public roads." https://www.theverge.com/2018/7/20/17595968/waymo-self-driving-cars-8-million-miles-testing, 2018.

[2] "The Evolution of Automated Safety Technologies." https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety, 2021.

[3] J. Hecht, "Lidar for self-driving cars," *Optics and Photonics News*, vol. 29, no. 1, pp. 26–33, 2018.

[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] "NVIDIA DRIVE." https://developer.nvidia.com/drive.

[6] "Baidu Apollo." apollo.auto.

[7] B. Schoettle and M. Sivak, "A preliminary analysis of real-world crashes involving self-driving vehicles," *Univ. of Michigan Transportation Research Institute*, 2015.

[8] P. Kohli and A. Chadha, "Enabling pedestrian safety using computer vision techniques: A case study of the 2018 uber inc. self-driving car crash," in *Future of Information and Communication Conf.*, pp. 261–279, 2019.

[9] 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 Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1625–1634, 2018.

[10] 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 *Proc. of the 2019 ACM SIGSAC Conf. on Computer and Communications Security*, pp. 2267–2281, 2019.

[11] J. Sun, Y. Cao, Q. A. Chen, and Z. Morley Mao, "Towards robust LiDAR-based perception in autonomous driving: General black-box adversarial sensor attack and countermeasures," in *Proceedings of the 29th USENIX Security Symposium*, pp. 877–894, 2020.

[12] J. Tu, H. Li, X. Yan, M. Ren, Y. Chen, M. Liang, E. Bitar, E. Yumer, and R. Urtasun, "Exploring Adversarial Robustness of Multi-Sensor Perception Systems in Self Driving," *arXiv preprint arXiv:2101.06784*, 2021.

[13] M. Abdelfattah, K. Yuan, Z. J. Wang, and R. Ward, "Adversarial Attacks on Camera-LiDAR Models for 3D Car Detection," *arXiv preprint arXiv:2103.09448*, 2021.

[14] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote Attacks on Automated Vehicles Sensors: Experiments on Camera and LiDAR," *Blackhat.com*, vol. 11, pp. 1–13, 2015.

[15] H. Shin, D. Kim, Y. Kwon, and Y. Kim, "Illusion and dazzle: Adversarial optical channel exploits against lidars for automotive applications," in *Lecture Notes in Computer Science*, vol. 10529 LNCS, pp. 445–467, 2017.

[16] Y. Cao, J. Ma, K. Fu, R. Sara, and M. Mao, "Automated Tracking System For LiDAR Spoofing Attacks On Moving Targets," 2021.

[17] Y. Cao, N. Wang, C. Xiao, D. Yang, J. Fang, R. Yang, Q. A. Chen, M. Liu, and B. Li, "Invisible for both Camera and LiDAR: Security of Multi-Sensor Fusion based Perception in Autonomous Driving Under Physical-World Attacks," in *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 176–194, 2021.

[18] J. Liu and J. Park, "" Seeing is not Always Believing": Detecting Perception Error Attacks Against Autonomous Vehicles," *IEEE Transactions on Dependable and Secure Computing*, 2021.

[19] M. Pajic, J. Weimer, N. Bezzo, P. Tabuada, O. Sokolsky, I. Lee, and G. J. Pappas, "Robustness of attack-resilient state estimators," in *2014 ACM/IEEE Int. Conf. on Cyber-Physical Systems (ICCPS)*, pp. 163–174, 2014.

[20] Z. Hau, S. Demetriou, L. Muñoz-González, and E. C. Lupu, "Shadow-Catcher: Looking Into Shadows to Detect Ghost Objects in Autonomous Vehicle 3D Sensing," *arXiv preprint arXiv:2008.12008*, 2020.

[21] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, and S. Mehta, "Lgsvl simulator: A high fidelity simulator for autonomous driving," in *2020 IEEE 23rd Int. Conf. on Intelligent Transportation Systems (ITSC)*, pp. 1–6, 2020.

[22] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[23] S. Kato, S. Tokunaga, Y. Maruyama, S. Maeda, M. Hirabayashi, Y. Kitsukawa, A. Monrroy, T. Ando, Y. Fujii, and T. Azumi, "Autoware on board: Enabling autonomous vehicles with embedded systems," in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, pp. 287–296, IEEE, 2018.

[24] M. Kutila, P. Pyykönen, W. Ritter, O. Sawade, and B. Schäufele, "Automotive LIDAR sensor development scenarios for harsh weather conditions," in *19th IEEE ITSC*, pp. 265–270, 2016.

[25] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep Continuous Fusion for Multi-sensor 3D Object Detection," in *Lecture Notes in Computer Science*, vol. 11220 LNCS, pp. 663–678, 2018.

[26] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," in *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4490–4499, 2018.

[27] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 12689–12697, 2019.

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

[29] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D Object Detection from RGB-D Data," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 918–927, 2018.

[30] Z. Wang and K. Jia, "Frustum ConvNet: Sliding Frustums to Aggregate Local Point-Wise Features for Amodal," *IEEE International Conference on Intelligent Robots and Systems*, pp. 1742–1749, 2019.

[31] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3D Proposal Generation and Object Detection from View Aggregation," in *IEEE Int. Conference on Intelligent Robots and Systems*, pp. 5750–5757, 2018.

[32] T. Huang, Z. Liu, X. Chen, and X. Bai, "Epnet: Enhancing point features with image semantics for 3d object detection," in *European Conference on Computer Vision*, pp. 35–52, Springer, 2020.

[33] B. Yang, W. Luo, and R. Urtasun, "PIXOR: Real-time 3D Object Detection from Point Clouds," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 7652–7660, 2018.

[34] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *ASIA CCS 2017 - Proceedings of the 2017 ACM Asia Conference on Computer and Communications Security*, pp. 506–519, 2017.

[35] A. Boloor, K. Garimella, X. He, C. Gill, Y. Vorobeychik, and X. Zhang, "Attacking vision-based perception in end-to-end autonomous driving models," *Journal of Systems Architecture*, vol. 110, p. 101766, 2019.

[36] R. Ivanov, M. Pajic, and I. Lee, "Attack-resilient sensor fusion," in *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1–6, IEEE, 2014.

[37] "The Frustum Attack." https://cpsl.pratt.duke.edu/research/frustum-attack.

[38] S. S. Blackman, "Multiple-target tracking with radar applications," *Dedham*, 1986.

[39] "Comma AI." https://comma.ai/.

[40] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. Part I. Dynamic models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, 2003.

[41] Y. Jia, Y. Lu, J. Shen, Q. A. Chen, Z. Zhong, and T. Wei, "Fooling Detection Alone is Not Enough: First Adversarial Attack against Multiple Object Tracking," in *Int. Conference on Learning Representations (ICLR)*, 2020.

[42] A. Piazzoni, J. Cherian, M. Azhar, J. Y. Yap, J. L. W. Shung, and R. Vijay, "ViSTA: a Framework for Virtual Scenario-based Testing of Autonomous Vehicles," *arXiv preprint arXiv:2109.02529*, 2021.

A Existing Attacks and Defenses

A.1 Naive Spoofing on LiDAR-only Perception

We implement the naive spoofing of [10, 11] with the attack model described in Section 5.2. We follow [11] to evaluate the attack success; i.e., we selected 5 attack traces using $\{10, 20, ...200\}$ points per trace over multiple trials for 100 attack evaluations, and placed the spoofed points in front-near positions, $5-8$ *m* from the victim vehicle. We extend the evaluation to outside of the front-near when evaluating defenses. Attack Success Rate (ASR) for FP outcomes is defined as the fraction of times a fictitious object is detected over the number of targeted attempts (e.g., number of spoof point clusters), as there could be more than one FP per frame. Similarly, we define the FN ASR as the fraction of times an object is missed over the number of attempts.

Our results, summarized in Fig. 16, confirm the success of the naive black-box spoofing attacks. The ASR is consistently high when 60 or more spoof points are used for all LiDAR-only algorithms showing that each of the 3 architectures are deeply vulnerable to spoof injections in front-near positions.

A.2 Existing Defenses against LiDAR Spoofing

We reproduce the state-of-the-art defenses against LiDAR spoofing; we showed that both CARLO and SVF dramatically reduce the ASR in front-near positions, while ShadowCatcher has challenges defending naive attacks (see Figs. 17 and 18). Note that SVF requires model-level changes and expensive retraining which are not possible with all the tested perception algorithms. Thus, we rearchitect and retrain PointPillars with SVF, following the approach from [11].

While the ShadowCatcher defense is impressively simple, we do not expect to obtain the high accuracy reported in [20]. The reason is that the original work made several assumptions that are unrealistic, including tuning parameters on the test set, using ground truth bounding boxes instead of outputs of a perception algorithm, which significantly alters the shadow-region estimation noise, and only testing on 200 scenes with only three selections of 200 spoofed points. Still, we reproduced the original ShadowCatcher defense and obtained not as

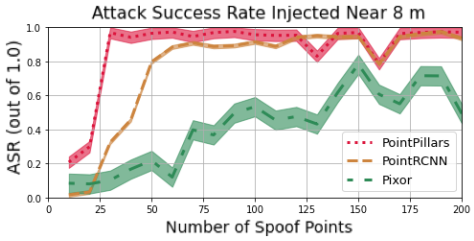


Figure 16: *Naive spoofing attacks against LiDAR-only perception:* Reproduced naive black-box spoofing attacks from [11] applied to LiDAR-only perception, one method from each of the three LiDAR-only architecture categories from Table 1.

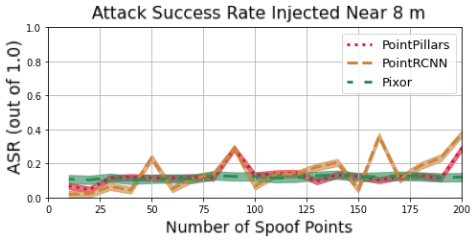


Figure 17: *Naive spoofing attacks on LiDAR-only perception with CARLO*: CARLO guards LiDAR-only perception against naive black-box spoofing attacks **only** in front-near positions.

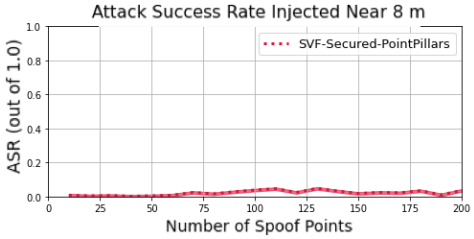


Figure 18: *Naive spoofing attacks on LiDAR-only perception with SVF:* SVF guards LiDAR-only perception against naive black-box spoofing in front-near positions. SVF requires rearchitecting the perception model which is not feasible for every algorithm. We test SVF-modified PointPillars.

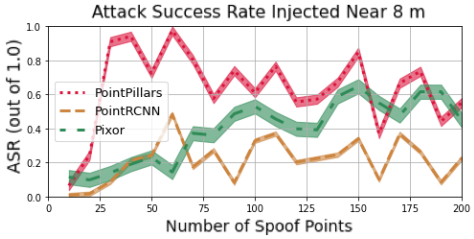


Figure 19: *Naive spoofing attacks on LiDAR-only perception with ShadowCatcher:* ShadowCatcher has limited ability to guard against LiDAR spoofing attacks presented in [11] due to difficulty handling noisy shadow estimation.

strong detection results without these assumptions (Fig. 19).

B CARLO Vulnerabilities

We described in Section 4.1.1 that CARLO introduces vulnerability to FP attacks outside front-near and FN invalidation

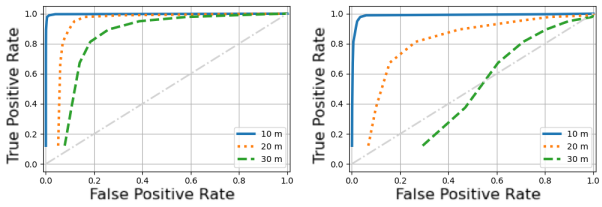


Figure 20: ROC for FP attack on CARLO using PointPillars as perception with (left) 60 and (right) 200 spoofed points.

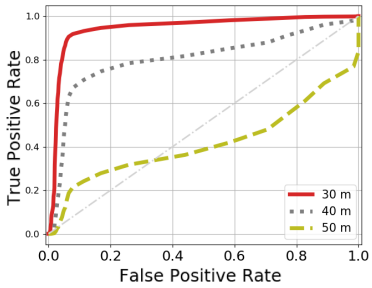


Figure 21: The CARLO defense is vulnerable to FN attacks. Classification of valid objects significantly degrades when randomly spoofing 200 points behind valid objects, as well as when range to target object increases.

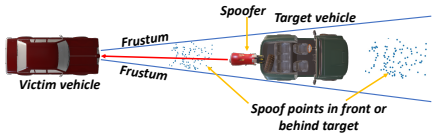
attacks. Here, we provide additional quantitative analysis.

To evaluate CARLO outside front-near, we collect receiver operating characteristics (ROC) on CARLO's ability to distinguish between valid and spoof objects placed at different ranges from the victim AV; the results are presented in Fig. 20. As in [11], we use PointPillars for this test; yet, CARLO is model-agnostic and these results generalize across other algorithms. We observe that, as range of the spoofed objects increases, CARLO's classification performance deteriorates (i.e., the defense breaks) and the ROC curve moves towards the center; e.g., in the case of 200 spoof points at 30 m (green), CARLO is similar to the random-guessing classifier.

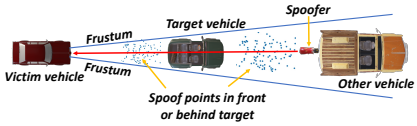
Similarly, to test the FN invalidation attack, we collect ROC curves where each curve represents the range to the targeted valid object in Fig. 21. For each object, we record the range to that object, add 200 spoof points in a random pattern behind it, run CARLO on the detected result, and check if it invalidated the true object. We find that CARLO's performance against the invalidation attack deteriorates when true objects are at an increased range from the victim, likely due to the decreased density of LiDAR points when objects are farther away.

C Spoofing Scenarios for Frustum Attacks

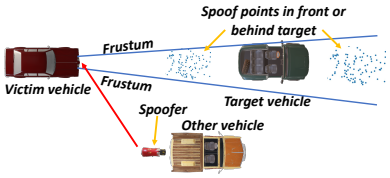
We illustrate common situations where vehicle configurations enable frustum attack spoofing (Fig. 22); the specific scenarios are described in Sec. 5.2. We execute the physical experiment in Section 5.2 corresponding to the scenario in Fig. 22b and similar to the scenario in Fig. 22a. We also perform the longitudinal case study in Sec. 6.2 in accordance with the scenario from Fig. 22b. Anticipated advances in optical technology and tracking will soon enable spoofing points



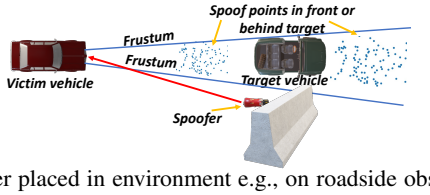
(a) Spoofer placed on target vehicle and points spoofed in LOS.



(b) Spoofer placed on larger vehicle in front of target, pointing in LOS.



(c) Spoofer placed on other vehicle in nearby lane, pointing out of LOS.



(d) Spoofer placed in environment e.g., on roadside obstacle, in LOS (e.g., on bridge) or out of LOS (e.g., on roadside)

Figure 22: The frustum attack in everyday driving scenarios. Adversaries can place a spoofer on target car, on other cars, or on roadside obstacles, placing points anywhere along line-of-sight (LOS) (e.g., in front or behind the target car).

outside of line-of-sight as demonstrated in e.g., [16].

D Impact of Spoof Point Placement

Prior works required spoofed points to be placed in patterns of occluded vehicles [11]. Here, we show that spoofing in a normally-distributed pattern for the frustum attack can have success nearly matching using occluded traces.

In Fig. 23, we compare the attackability of FPN perception when using the two spoof point generation methods (similar results are also obtained for the other aforementioned perception algorithms). We first provide a comparison of the fraction of instances where an attack succeeds using both methods, as function of the distance to the target (Fig. 23-left) As can be observed, it is difficult to distinguish between the performance of the two methods (random points vs. car-pattern).

We further provide an analysis of the attackability as a function of the number of points in the target objects' bounding box (Fig. 23-right) We define attackability as the ability to find an attack that succeeds within the attacker-specified capabilities. Both methods perform similarly, with a small benefit of car patterned injections at medium range. This improve the feasibility of LiDAR-based spoofing, as a normally distributed pattern does not require unrealistically careful placement of spoof points, and is robust to small displacements.

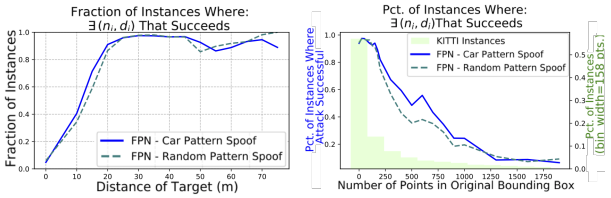
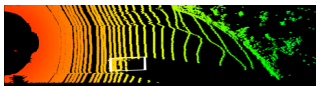
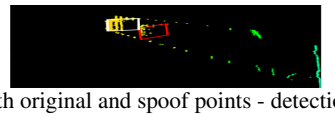


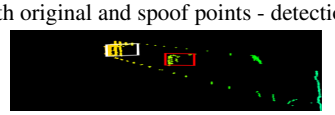
Figure 23: Spoofing in a Gaussian random pattern (Table 2) achieves performance on-par with using an occluded car pattern; we test spoofing patterns on FPN and show dependence of attackability on (left) range to target and (right) number of points in target object bounding box with histogram showing frequency (%) of occurrence of such objects in KITTI.



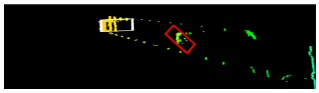
(a) Full point cloud with ground truth object box at 16 m (white)



(b) Frustum with original and spoof points - detection (red) at 20 m



(c) Frustum with original and spoof points - detection (red) at 24 m



(d) Frustum with original and spoof points - detection (red) at 27 m

Figure 24: Target object (front, white box) at 16 m with 492 points in bounding box. Just 65 points alter the target vehicle's location and achieve detections (translations) using FPN.

E Frustum Attack By Range to Target

Fig. 12, in Section 5.3.2, summarizes the frustum attack performance against the AVOD perception algorithm for different parameter combinations. Here, we provide the results for all other aforementioned algorithms (Fig. 25). The majority of algorithms are vulnerable to frustum attacks both in front and behind the targeted object. In general, attack success increases as the range to the target object increases (left to right in a row), with low attack success for all algorithms when attacks occur near the target object; this is expected as the original object and false positive will be "merged" (i.e. only a single detection) once they are on top of each other.

F Longitudinal Frustum Attack Visualizations

Spoofing points in successively changing distances causes the FP injection to appear to travel longitudinally. Fig. 24 shows a BEV visualization of such a longitudinal attack where the perception detects motion of the spoofed points (red).

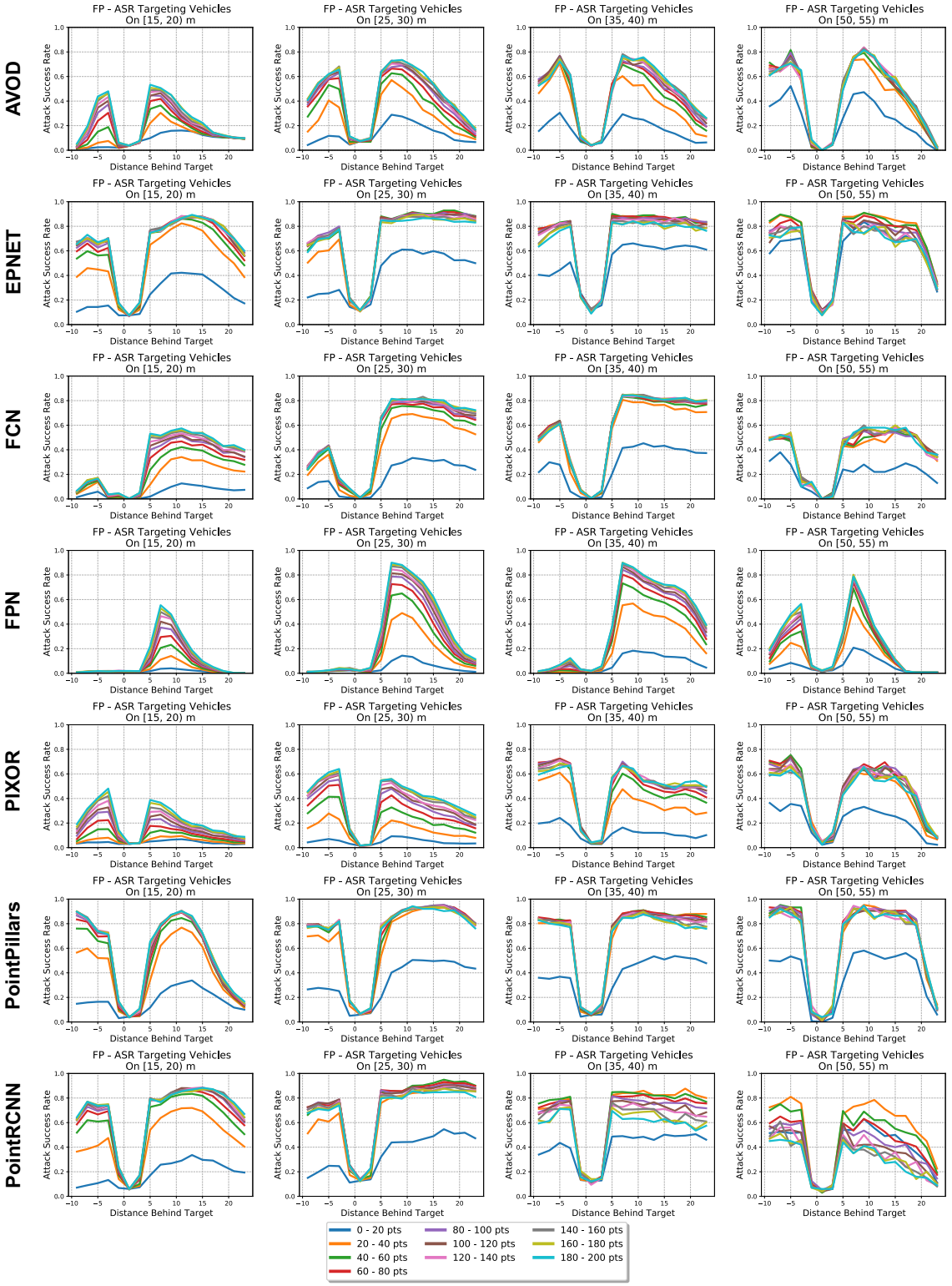


Figure 25: Frustum attack success rate against all perception algorithm for different parameters (i.e., number of spoofed points, target vehicle range, distance of spoofing behind the target) combinations, tested on all objects in KITTI validation set. Each tested algorithm (row) is widely vulnerable to the frustum attack. ASR depends on the range to the target vehicle (column). Note a dead-zone near the target vehicle (i.e. relative distance=0) where attacks do not succeed and increased ASR as target range increases. Most algorithms are vulnerable to spoofing both in front (< 0 on x-axis) and behind (> 0 on x-axis) target objects.