Laser-based LiDAR Spoofing: Effects Validation, Capability Quantification, and Countermeasures

Zizhi Jin, Xiaoyu Ji, *Member, IEEE,* Yushi Cheng, Bo Yang, Chen Yan, Wenyuan Xu, *Fellow, IEEE*

***Abstract*—Autonomous vehicles and robots increasingly exploit LiDAR-based 3D object detection systems to detect obstacles in the environment. Correct detection and classification are important to ensure safe driving. Though existing work has demonstrated the feasibility of manipulating point clouds to spoof 3D object detectors, most of the attempts are conducted digitally. In this paper, we investigate the possibility of physically fooling LiDAR-based 3D object detection by injecting adversarial point clouds using lasers. First, we develop a laser transceiver that can inject up to 4200 points, and can measure the scanning cycle of victim LiDARs to schedule the spoofing laser signals. By designing a control signal method that converts the coordinates of point clouds to control signals and an adversarial point cloud optimization method with physical constraints of LiDARs and attack capabilities, we manage to inject spoofing point cloud with desired point cloud shapes into the victim LiDAR physically. We can launch four types of attacks, i.e., naive hiding, record-based creating, optimization-based hiding, and optimization-based creating. Extensive experiments demonstrate the effectiveness of our attacks against two commercial LiDAR and three detectors. We further analyze the impact of our attacks on four fusion-based detectors. The paper concludes with experiments on defense methods and discussion on potential defense strategies at both the sensor and autonomous vehicle system levels.**

***Index Terms*—LiDAR, 3D objection detection, autonomous driving, laser, adversarial attack.**

I. Introduction

THE proliferation of autonomous vehicle (AV) solutions has acted as a catalyst for the integration of LiDAR (Light Detection and Ranging) into advanced driving assistance systems (ADAS) [1], [3], [6] and cooperative vehicle infrastructure systems (CVIS) [24], [25], [23], [36] (Fig. 1). According to Yole Développement [41], the LiDAR market for automotive and industrial applications is expected to reach a value of USD 3.8 billion by 2025. By generating precise 3D point clouds of surrounding environments, LiDAR and its affiliated 3D object detection algorithms can detect and classify obstacles on the roads. This capability enables AVs to make safety-critical driving decisions.

Many prior works have demonstrated the vulnerabilities of LiDAR-based 3D object detection systems, primarily through the digital manipulation of 3D point clouds [10], [11], [35], [34]. Several works have endeavored to generate 3D adversarial point clouds physically by placing 3D-printed obstacles in specific locations [14] or by maneuvering drones around target objects [43]. Nevertheless, these methods predominantly focus on Denial of Service (DoS) attacks, and the adversarial object way is conspicuous to human eyes.

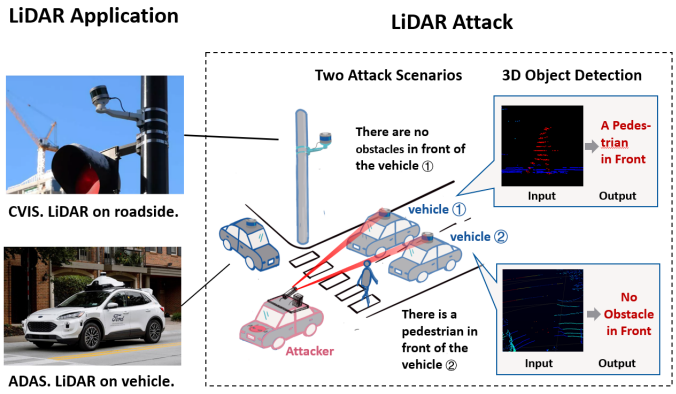


Fig. 1: **By injecting malicious laser signals into the LiDAR of the 3D object-detection system in autonomous vehicles, an attacker can fool decision-making.**

In this paper, we inquire *"Can we physically spoof 3D object detection by injecting adversarial point clouds using lasers?"* Specifically, we consider the following attack scenario: An adversary may shoot lasers into the onboard LiDAR of an autonomous vehicle that is waiting for traffic lights, as shown in Fig. 1, resulting in two kinds of errors, detecting a none-existent object or failing to detect an object ahead.

To the best of our knowledge, this is the first research endeavor to concentrate on physical attacks against LiDAR-based 3D object detection. While earlier research has demonstrated that 3D object detection systems are vulnerable to digital adversarial point clouds, no study has investigated the possibility of physically transmitting the generated adversarial point clouds to the LiDAR. Achieving such an aim is non-trivial, given that LiDAR continuously rotates in the horizontal plane and scans at the vertical plane at high speed. How to physically inject the point clouds into the LiDAR with the desired shape and location in the presence of environment noises and device jitters is still not studied. What's more, the capability of point cloud injection reported in previous work is 200 points at most [34], which is not strong enough to achieve the aforementioned attacks. The question of whether and how the point injection ability can be enhanced remains unclear.

To overcome the aforementioned challenges, we design a physical laser attack against LiDAR-based 3D object detection, PLA-LiDAR. To improve the capability to inject point clouds, we develop a laser transceiver that can inject up to 4200 points, which is 20 times more than that prior work has achieved and is the key factor to achieve physical attacks.

Corresponding author: Xiaoyu Ji
Z. Jin, X. Ji, Y. Cheng, B. Yang, C.Yan and W.Xu are with the College of Electrical Engineering, Zhejiang University, Hangzhou, CN.
E-mail: zizhi, xji, yushicheng, yb5, yanchen, wyxu@zju.edu.cn

To generate adversarial point clouds that can be physically injected into the victim LiDAR, we propose a new adversarial point cloud optimization method that considers the working principle of the LiDAR, the capability of the attack devices, and the distance error of the injected point during optimization. To precisely generate the desired shape of the injected point cloud, we propose a control signal design method that converts the shape of the point cloud into a control signal. To accurately control the distances of the injected points, we propose a new synchronization method to align the attack signal with the scanning sequence of the victim LiDAR.

Based on the aforementioned methods, PLA-LiDAR can induce the following attack effects affecting safety-critical decision making:

- **Hiding**: the victim AV fails to perceive an existing object.
- **Creating**: the victim AV perceives a non-existing object.

To validate our attacks, we conduct extensive physical evaluations with Velodyne VLP-16 [38] and Robosense RS-16 [30] on two academic 3D object detectors PointPillar [22] and SECOND [40] and one commercial 3D object detector Apollo [1].

In summary, our contributions include the points below:

- To the best of our knowledge, we are the first work on physical attacks against LiDAR-based 3D object detection via lasers.
- We design the PLA-LiDAR attack, which improves the capability to inject spoofing points by 20 times compared with prior work, and can inject adversarial point clouds into the LiDAR with the controllable shape and location to hide or create target objects.
- We validate the effectiveness of our attacks against two widely-used mechanical LiDARs with two academic 3D object detectors (PointPillars and SECOND) and one commercial detector (Apollo). We further analyze the impact of our attacks on four fusion-based detectors(F-Pointnet, EPNet, AVOD and CLOCs).
- We propose a defense method (PLA-Defense) that includes sensor fusion and spoofing points detection. The PLA-Defense can defend against all four types of attack effects of PLA-LiDAR.

II. Background

In this section, we introduce the basics of the LiDAR and LiDAR-based 3D object detection system.

A. Mechanical LiDAR

Common LiDARs on the market include (1) mechanical (spinning) LiDAR uses a rotating assembly to spin the sensor and transmits pulsed lasers during rotation to achieve 360-degree sensing, and (2) solid-state LiDAR that has no spinning mechanical components and scans using Micro-Electro-Mechanical System [31] or Optical Phased Array [29] technology. In these two types of LiDARs, the mechanical one takes up over 95% share of the global LiDAR market in 2020 [42], and is being adopted in many large-scale commercial autonomous driving projects (e.g., Waymo One [7], Baidu Robotaxi [2]). Due to its large market share and wide use, we study the mechanical LiDAR in this paper.

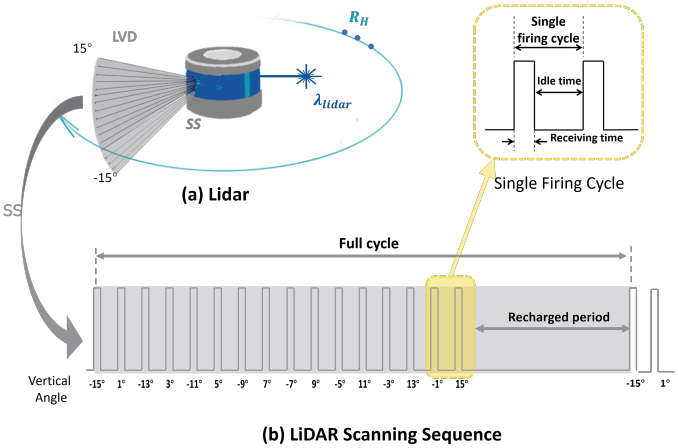


Fig. 2: **The (a) structure, and (b) scanning sequence of the VLP-16 LiDAR.**

As shown in Fig. 2 (a), a mechanical LiDAR uses an array of multiple infrared (IR) lasers paired with infrared detectors that fires and receives specific-wavelength laser pulses in a specified order and interval to measure distances to objects. The laser array is fixed in a specific vertical distribution, and rotates rapidly to scan the surrounding environment with a horizontal angular resolution related to the rotational speed. In general, a mechanical LiDAR has four key parameters, which are (1) Scanning Sequence SS, (2) Laser Vertical Distribution LVD, (3) Horizontal Angular Resolution R_H, and (4) Wavelength λ_{lidar}, as shown in below:

$$LiDAR = [SS, LVD, R_H, \lambda_{lidar}] \tag{1}$$

Scanning Sequence. SS refers to the time sequence that describes how LiDAR transmits and receives laser pulses. Every LiDAR model has its own SS. As shown in Fig. 2 (b), the length of a scanning sequence is *full cycle* (T_{fc}), during which all the lasers are fired and recharged once with a specific order. The minimum time between each firing is *single firing cycle* (T_{sfc}). After each firing, the LiDAR listens for an echo within a receiving time, and the specific pulse (the strongest one or the last one depending on the return mode of the LiDAR) received during the receiving time is considered a valid echo. After the receiving time is over, the LiDAR waits for an idle time before transmitting the next pulse.

Laser Vertical Distribution. The laser vertical distribution represents the vertical field of view and the vertical resolution of a LiDAR. It is a factory-set parameter and can be acquired from the user manual.

Horizontal Angular Resolution. The horizontal angular resolution represents the minimum angular difference of the lidar points in the horizontal direction. The faster the LiDAR rotates, the greater the R_H. The rotation speed of LiDAR is expressed in RPM (Rotation Per Minute) and can be configured by users.

LiDAR Wavelength. Current state-of-the-art LiDAR systems usually employ lasers with one of the following two wavelengths: 905 nm and 1550 nm [4]. Generally, the LiDAR

is most sensitive to the laser of the working wavelength, and will filter the light of other wavelengths.

B. LiDAR-based 3D Object Detection

Autonomous vehicles increasingly utilize deep learning techniques to process LiDAR point clouds. State-of-the-art 3D object detectors are usually based on deep learning techniques and have three categories: (1) bird's-eye view (BEV) based methods that take point cloud's BEV representation as model inputs and use 2D Convolutional Neural Networks (CNNs) in feature learning, (2) voxel-based methods that divide the 3D point cloud space into voxels and learn features through 3D CNNs, and (3) point-wise methods that directly operate on point clouds to learn features. Among these detectors, the BEV-based and voxel-based ones are commonly used. We study their representative models PointPillar [22], Apollo [1], SECOND [40] in this paper.

III. Threat Model

In this section, we present our attack goal and the attack capabilities possessed by the adversaries.

A. Attack Goal

In this paper, our attack goal is to inject malicious points into a mechanical LiDAR and spoof its 3D object detection into mistakes. Specifically, we consider two attack objectives: (1) *Hiding*: the victim AV fails to perceive an existing object, and (2) *Creating*: the victim AV perceives a non-existing object. We further consider 4 types of attacks as shown Fig. 3:

- **Naive Hiding Attacks (Nai-Hide)** that cause an existing object to be undetectable by creating a fake wall far away.
- **Record-based Creating Attacks (Rec-Create)** that induce a non-existing object by injecting recorded point clouds into the LiDAR.
- **Optimization-based Hiding Attacks (Opt-Hide)** that cause an existing object to be undetectable by injecting optimized adversarial points into the LiDAR.
- **Optimization-based Creating Attacks (Opt-Create)** that induce a non-existing object by injecting optimized adversarial points into the LiDAR.

Compared with naive hiding attacks and record-based creating attacks, optimization-based attacks require fewer injected points to achieve similar attack effects.

B. Adversary's Capabilities

To achieve the aforementioned attack goal, we assume the adversary has the following capabilities:

LiDAR Parameter Awareness. The adversary can acquire and analyze a LiDAR of the same model as the one used in the victim AV, from which or its user manual she can learn the LiDAR parameters including scanning sequence, laser vertical distribution, etc. In addition, the adversary can measure the rotation speed of the victim LiDAR using photoelectric sensors and oscilloscopes.

White-box Object Detector. For optimization-based hiding and creating attacks, the adversary has prior knowledge of the 3D object detection algorithm used in the victim AV, including but not limited to its architecture, parameters, outputs, etc. For naive hiding attacks and record-based creating attacks, the adversary does not require any prior information of the object detectors.

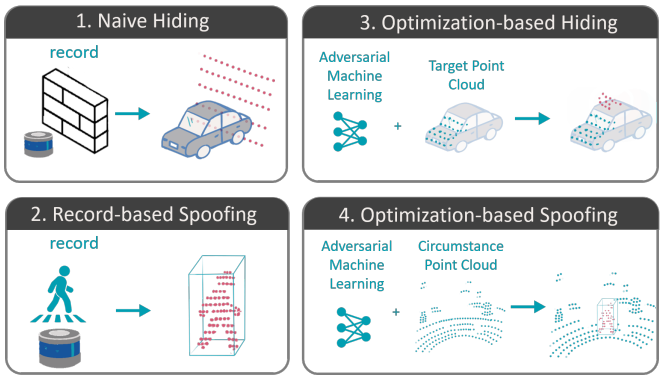


Fig. 3: **Four fine-grained attack types.**

Physical Attack Capability. The adversary can transmit lasers towards the LiDAR in the target AV by using an attack apparatus consisted of commodity devices such as photoelectric sensors, arbitrary waveform generators, and laser transmitters. To achieve it, she can drive a car in a similar speed to the target AV and measure the distance between the laser transmitter and the victim LiDAR by laser ranging techniques [9].

IV. Attack Design

To conduct physical adversarial attacks against 3D object detection using lasers, it is important to address the following challenges:

- **Challenge 1:** How to generate spoofing point clouds that can be injected into the LiDAR?
- **Challenge 2:** How to physically inject the spoofing point clouds into the LiDAR?

To address these challenges, we design PLA-LiDAR attack that incorporates four key modules, as shown in Fig. 4. The **LiDAR Parameter Measurement** module measures the victim LiDAR to acquire attack-related parameters including scanning sequence and horizontal angular resolution. The **Point Cloud Generation** module generates spoofing point clouds that theoretically can be injected into the LiDAR by recording or adversarial optimization. The **Control Signal Design** module converts a desired spoofing point cloud into a laser signal by designing a control signal that specifies the emitting time of each laser pulse. The **Synchronization** module synchronizes the scanning sequence of the victim LiDAR and the control signal, and transmits lasers with selected laser transmitters to launch physical attacks.

A. LiDAR Parameter Measurement

To physically inject laser into the victim LiDAR, we first acquire several key parameters by measurement.

As shown in Sec. II-A, a mechanical LiDAR has four key parameters, i.e., Scanning Sequence SS, Laser Vertical

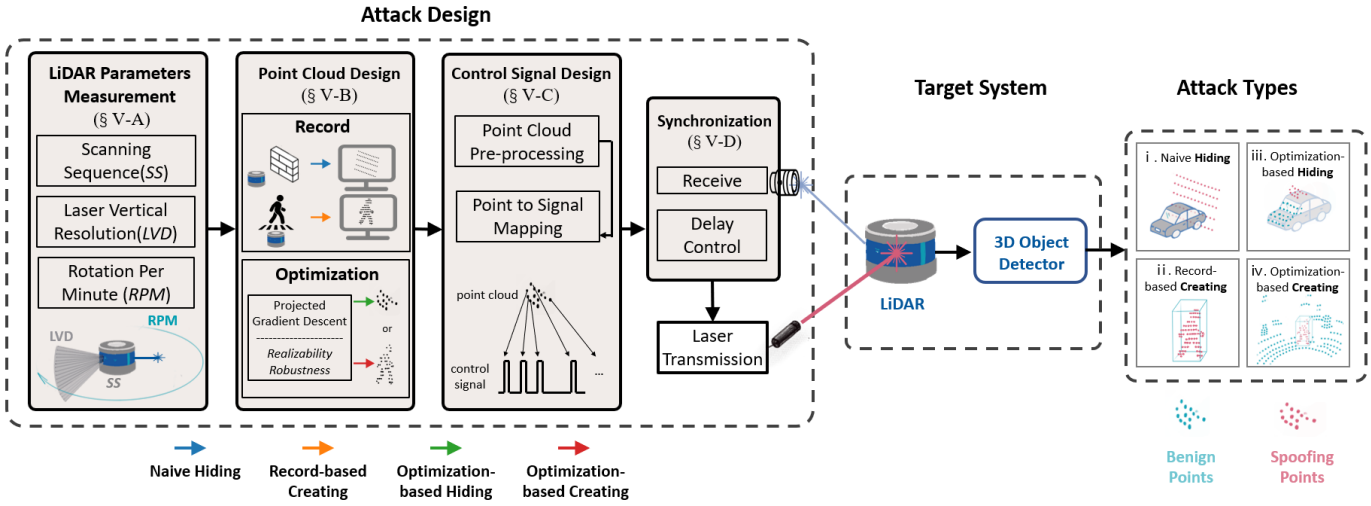
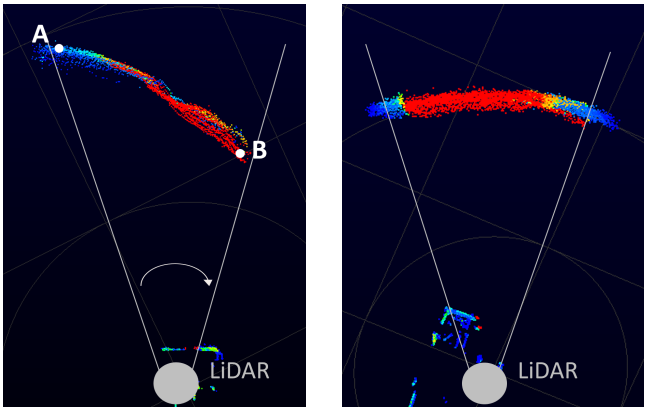


Fig. 4: **The attack workflow. By measuring the victim LiDAR, the adversary first generates injectable point clouds by recording or adversarial optimization, then converts the expected point clouds into control signals, and finally injects the lasers into the victim LiDAR by signal synchronization, which may deceive the 3D object detector and lead to hiding or creating attacks.**



(a) Before scanning sequence correction. SS = (55.296 μs, 2.304 μs).

(b) After scanning sequence correction. SS = (55.296216 μs, 2.304 μs).

Fig. 5: **The spoofing wall recovers from distortions after scanning sequence correction.**

Distribution LVD, Rotation Per Minute RPM, and Horizontal Angular Resolution R_H. Among those parameters, LVD and R_H are important in the point cloud design to generate spoofing points on the laser ray of the victim LiDAR such that they can be physically received by the LiDAR. SS is used in the control signal design and synchronization to ensure the emitted lasers are received by the victim LiDAR. λ_{lidar} provides guidance on the selection of attack lasers.

For the four parameters, LVD and λ_{lidar} can be obtained from the official documents of the victim LiDAR, while SS and R_H shall be acquired or corrected by measurement.

Scanning Sequence Correction. In generally, SS can also be obtained by official user manuals of LiDARs. However, injecting spoofing point clouds using the official SS will suffer obvious distortions since the real full cycle $T_{fc}^{'}$ has a slight offset compared with the official one T_{fc}.

To address it, we propose a scanning sequence correction method. First, we inject a spoofing "wall" into the victim LiDAR by using the attack device in Fig. 8 to fire a pulsed laser signal whose period is the same as the official SS.

Then, we observe the shape of the "wall", which will be a sphere ideally. If the "wall" is distorted as shown in Fig. 5(a)), we mark its starting point A and ending point B (preferably with the same vertical angle), and correct T_{fc} by Equ. 2

$$T_{fc}^{'} = \frac{D_A - D_B}{c} * \frac{1}{N_{fc}} + T_{fc} \tag{2}$$

where D_A and D_B are the distances of A and B to the LiDAR, c is light speed, N_{fc} is the number of full cycles caused by the LiDAR for going through from A to B.

Rotation Per Minute Measurement. The rotation per minute RPM of the LiDAR can be set by users. Thus, we measure it physically by using a photoelectric sensor to receive laser pulses from the victim LiDAR and using an oscilloscope to observe the interval between two pulses. An interval of 100 ms indicates a LiDAR rotation speed of 10 Hz, giving $RPM = 60 * \frac{1}{interval} = 600$.

Horizontal Angular Resolution Calculation. The horizontal angular resolution R_H relates to the rotation speed of the LiDAR, and can be obtained based on RPM and T_{fc} with the following Equ. 3.

$$R_H = 360 * \frac{T_{fc} * RPM}{60} \tag{3}$$

B. Point Cloud Design

To design spoofing point clouds that can be received by the victim LiDAR, we consider two types of point cloud generation methods in this paper: (1) record-based point cloud generation, and (2) optimization-based point cloud generation. The record-based method requires no prior information about the 3D object detectors but needs a substitute LiDAR of the same model as the victim LiDAR. The optimization-based method is more delicate and reduces the requirement of point numbers but requires white-box access to the object detectors. In practice, the adversary can choose the appropriate point cloud generation method according to the attack scenarios.

1) Record-based Point Cloud Generation: The record-based point cloud generation method is used for Nai-Hide and Rec-Create attacks. To achieve it, we first acquire a LiDAR of the same model as the victim LiDAR, which we call the substitute LiDAR. Then, based on the expected attack target (class) and the attack distance, we use the substitute LiDAR to record the point cloud of an object of the target class, e.g., a wall for Nai-Hide attacks or a pedestrian for Rec-Create attacks.

The benefit of this method is that the generated point cloud is collected from substitute LiDARs and thus is in line with the victim LiDAR's working principle. As a result, the replayed one can be received by the victim LiDAR naturally.

2) Optimization-based Point Cloud Generation: For optimization-based hiding and creating attacks, we generate spoofing point clouds by adversarial machine learning. Compared with the record-based method, the optimization-based one exploits the vulnerability of the object detection algorithms, and has the potential of hiding or inducing a point cloud of a target class at any distance with fewer points.

Problem Formulation. To achieve this goal, we first introduce a physical constraint that shall be considered during the generation. Digital adversarial point clouds may not be practical physically since they do not consider the working principle of the LiDAR, i.e., it emits and receives reflected signals discretely. Therefore, a spoofing point can only be injected during a firing cycle, and at most one point can be injected for an individual firing cycle. We formulate the above two observations as the physical constraints for optimization to ensure all the generated spoofing points can be physically injected into the victim LiDAR.

> ***Physical Constraint:*** *Every generated point only occurs on one of the LiDAR's laser rays and each laser ray has at most one point.*

To better comply with the physical contrast, we generate adversarial point clouds in the spherical coordinates and formulate this problem as a gradient-based optimization problem:

$$\begin{aligned} \min_{\mathrm{P}^{'}} \quad & \mathcal{L}(\mathrm{P}^{'}) \\ \text{s.t.} \quad & (r_i^{'}, \theta_i^{'}, \phi_i^{'}) \in \mathrm{Loc}^{exp}, i \in [1, n] \\ & |\theta_i^{'} - \theta_j^{'}| + |\phi_i^{'} - \phi_j^{'}| \neq 0, i, j \in [1, n] \\ & \theta_i^{'} \in \mathbb{W} \end{aligned} \tag{4}$$

where $\mathrm{P}^{'} = \{(R_i^{'}, \Theta_i^{'}, \Phi_j^{'}) | i \in [1, n]\}$ is the adversarial point cloud, $r_i^{'}$, $\theta_i^{'}$ and ϕ_i are the distance, vertical angle, and horizontal angle of the adversarial point respectively, $\mathrm{Loc}^{exp} = \{x_a, y_a, z_a, w_a, l_a, h_a, yaw_a\}$ represents the center point, length, width, and height of the target area, and $\mathbb{W}$ indicates the range of the vertical angle specified by the victim LiDAR.

Loss Function Design. We then design the loss functions for the Opt-Hide and Opt-Create attacks, respectively. For Opt-Hide attacks, our goal is to inject adversarial point clouds into the vicinity of a target object to make it undetectable. To achieve it, we suppress the bounding box proposals related to the victim objects. A proposal close to the target object can be considered relevant if (1) their intersection over union (IoU) is larger than a threshold ϵ_i, and (2) the class prediction confidence of the proposal is larger than a threshold ϵ_s. Considering the practicality of the physical attacks, we choose to inject adversarial points above the target object and suppress those relevant proposals to avoid the possible blocking from the target object. In this way, the loss function for Opt-Hide attacks is as follows:

$$\mathcal{L}_h = \sum_{b,s \in B} -\mathrm{IoU}(b^t, b)\log(1-s) \tag{5}$$

where $\mathrm{B} = \{(x_i, y_i, z_i, w_i, h_i, l_i, yaw_i) | i \in [1, n]\}$ is the set of all the bounding box proposals, b^t is the ground truth of the victim object, and b and s are the relevant bounding box proposals and their confidences, respectively. In our implementation, $\epsilon_i = 0.1$ and $\epsilon_s = 0.1$.

For Opt-Create attacks, our goal is to induce a target object into a specific location by injecting adversarial points into this area, e.g., 10 meters in front of the victim LiDAR. To achieve it, we improve the bounding box proposals related to the expected area. Different from Opt-Hide attacks, we select the Top 10 bounding box proposals that have the largest IoUs with the expected area as the relevant proposals. In this way, we design the loss function for Opt-Create as follows:

$$\mathcal{L}_c = \sum_{b,s \in \mathrm{B}} -\mathrm{IoU}(b^e, b)\log(s) \tag{6}$$

where $b^e = \{x_e, y_e, z_e, w_e, h_e, l_e, yaw_e\}$ is the target area.

Robustness Enhancement. In laser-based physical attack, there is often an error between the real injected point cloud and the desired point cloud due to the limitation of the device sampling rate and the physical noise. As detailed in Sec.14(d), we quantify this error as shape control error and position control error. To make the adversarial point cloud more robust, we incorporate the error into the optimization process when implement physical attack. Specifically, for each point, we give a random perturbation $\delta \sim U(-d, d)$, and for the adversarial point cloud we give a random perturbation $\Delta \sim U(-D, D)$, where $\sim$ denotes obedience, and U denotes uniform distribution.d and D can be determined by the shape control and distance control capabilities, respectively. The robustness-enhanced optimization problem can be formulated as:

$$\min_{\mathrm{P}^{'}}(\mathcal{L}(\mathrm{P}^{'}) + \mathcal{L}(E(\mathrm{P}^{'}, \delta, \Delta))) \tag{7}$$

where$E(\mathrm{P}^{'}, \delta, \Delta)$ denotes the process of adding random perturbations to the adversarial point cloud $P^{'}$

Optimization Process. With the loss functions, we then design the following optimization process for Opt-Hide or Opt-Create attacks:

- Step 1: Calculate the spherical coordinate range of the adversarial points according to the location where the adversary expects to induce or hide a target object;
- Step 2: Randomly add a given number of adversarial points in the aforementioned range;
- Step 3: Calculate the gradient of the loss function for Opt-Hide or Opt-Create attacks (robustness enhancement needs to be incorporated during physical attack);

- Step 4: Update R of the adversarial point cloud $P^{'}$.
- Step 5: Repeat Step 3 and Step 4 until the loss converges or the iterations end.

C. Control Signal Design

To inject the generated point cloud into the victim LiDAR, we design the attack signal to be a series of laser pulses. Each pulse represents a spoofing point and the occurring moment of each pulse's rising edge determines the space coordinate of the spoofing point. We use a laser diode to emit the attack signal. The emitting time of each laser pulse, which determines the location of each injected point, is determined by the TTL control signal of the laser diode driver board. As a result, given a spoofing point cloud with a specific shape, we shall design a control signal that specifies the emitting time of each laser pulse in the attack signal based on the location of its corresponding spoofing point.

To achieve it, we first perform point cloud pre-processing and then design corresponding attack signals by point-to-signal mapping. The algorithm of control signal design is shown in Algorithm 1.

1) Point Cloud Pre-processing: The space position of a LiDAR point is usually described in the spherical coordinate system with the vertical angle (θ), horizontal angle (ϕ), and distance (r). To transform a spoofing point to the corresponding laser pulse, we first transform the spherical coordinates of every point in spoofing point cloud to the time coordinate (*fullcycle_id*, *singlecycle_id*, *tof*), where *fullcycle_id* represents which full cycle the pulse is in, *singlecycle_id* represents which single firing cycle the pulse is in, *tof* represents the theoretical time-of-flight of the laser pulse in order to generate this point. The time coordinates can be used to calculate the occurring time of the rising edge of the TTL control signal.

To calculate *fullcycle_id*, we first set the *fullcycle_id* of the minimum-azimuth (whose horizontal angle is ϕ_0) point to zero. Then, the *fullcycle_id* of a point whose horizontal angle is ϕ can be calculated as follows:

$$\textit{fullcycle\_id} = \frac{\phi - \phi_0}{r_H} \tag{8}$$

where r_H is the horizontal angular resolution. To calculate *singlecycle_id*, we build a mapping between $vertical\_angle$ and $singlecycle\_id$ (denoted as $Angle2ID$) according to the scanning sequence. The *singlecycle_id* of a point can be obtained according to the vertical angle of the point by $Angle2ID$. The *tof* of a spoofing point can be calculated based on the principle of time of flight as follows:

$$\textit{tof} = 2 * \frac{r}{c} \tag{9}$$

where *r* is the radial distance of a point to the LiDAR and *c* is light speed.

2) Point to Signal Mapping: With the time coordinates obtained from the point cloud pre-processing, we then design the TTL control signal consisting of a series of pulses, where each pulse represents a spoofing point, and sample it into discrete signals to be readable for a signal generator.

To design the TTL control signal, we first calculate the precise timestamp for each point in the spoofing point cloud according to its time coordinates (***ToF***, ***singlecycle_ID***, ***fullcycle_ID***) as follows:

$$\begin{aligned} \textbf{\textit{Timestamp}} = &\textbf{\textit{fullcycle\_ID}} * T_{fc} + \\ &\textbf{\textit{singlecycle\_ID}} * T_{sfc} + \textbf{\textit{ToF}} \end{aligned} \tag{10}$$

Then, we generate an ideal control signal whose rising edges locate at the calculated ***Timestamp***. Specifically, we design the pulse start time of the ideal pulse to be the same as ***Timestamp***, the rising and falling edges to be a minimum value ε, and the pulse width to be 10 ns, which is similar to the pulse width of the laser signal of VLP-16 and RS-16.

With those settings, we design a control signal to determine when to emit the lasers. To make it readable by signal generators, we further sample it with the sampling rate of the used signal generator.

Algorithm 1: Control Signal Design

Input: Points Number:N;
Cartesian coordinates: $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$;
Light speed: c;
LiDAR rotation speed: RPM ;
Full cycle: T_{fc};
Vertical angle to laser id mapping: Angle2ID
Output: Ideal consecutive TTL control signal $\boldsymbol{Signal}_{ideal}$;
Discrete TTL control signal $\boldsymbol{Signal}_{discrete}$

/* Point Cloud Pre-Processing */
Distance: $\boldsymbol{R} = \sqrt{\boldsymbol{X}.^2 + \boldsymbol{Y}.^2 + \boldsymbol{Z}.^2}$;
Vertical Angle: $\boldsymbol{\Theta} = \arcsin(\boldsymbol{Z}./\boldsymbol{R})$;
Horizontal Angle: $\boldsymbol{\Phi} = \arctan(\boldsymbol{X}./\boldsymbol{Y})$;
Point Cloud: PC = ($\boldsymbol{R}$, $\boldsymbol{\Theta}$, $\boldsymbol{\Phi}$) = Coordinate_Conversion($\boldsymbol{X}$,$\boldsymbol{Y}$,$\boldsymbol{Z}$) ;
$\boldsymbol{ToF} = 2 * \frac{\boldsymbol{R}}{c}$;
$\boldsymbol{laser\_ID} = Angle2ID(\boldsymbol{\Theta})$;
Horizontal Resolution: $\delta_{hori} = 360 * T_{fc} * \frac{60}{RPM}$;
PC_{sort} = sort($PC|\boldsymbol{\Phi}, \boldsymbol{laser\_ID}$) ;
$\boldsymbol{fullcycle\_ID}(0) = 0$;
for *i=1:N-1* **do**
  $\Delta = \boldsymbol{\Phi}(i) - \boldsymbol{\Phi}(i-1)$;
  $\Delta N_{fullcycle} = fix(\Delta/\delta_{hori})$ **if** $\boldsymbol{laser\_ID}(i) < \boldsymbol{laser\_ID}(i-1)$ **then**
    $\boldsymbol{fullcycle\_ID}(i) = \boldsymbol{fullcycle\_ID}(i-1) + \Delta N_{fullcycle} + 1$
  else
    $\boldsymbol{fullcycle\_ID}(i) = \boldsymbol{fullcycle\_ID}(i-1) + \Delta N_{fullcycle}$
  end
end
/* Point to Signal Mapping */
Pulse width(time to live): $TTL = 10 * 10^{-9} s$;
$\boldsymbol{Time\_ideal}(0) = 0$;
$\boldsymbol{Amp\_ideal}(0) = 0$;
Minimal value $\varepsilon = 1 * 10^{-18}$;
for *i=0:N-1* **do**
  $\boldsymbol{Time\_ideal}(i*4+1 : i*4+4) = [-\varepsilon, 0, TTL, TTL+\varepsilon] + \boldsymbol{Timestamp}(i)$;
  $\boldsymbol{Amp\_ideal}(i*4+1 : i*4+4) = [0,1,1,0]$
end
$\boldsymbol{Time\_ideal}(N*4+1) = (\boldsymbol{fullcycle\_ID}(N-1)+1) * T_{fc}$;
$\boldsymbol{Amp\_ideal}(N*4+1) = 0$;
$\boldsymbol{Signal}_{ideal} \leftarrow$ Take ***Time_ideal*** as abscissa and ***Amp_ideal*** as ordinate ;
$\boldsymbol{Signal}_{discrete} \leftarrow$AD sample the $\boldsymbol{Signal}_{ideal}$ with the sampling rate SR ;

D. Synchronization

With the control signal, we can inject spoofing points into the LiDAR. However, to inject point clouds with specific shapes at particular locations, the attack signal must be aligned with the scanning sequence of the victim LiDAR.

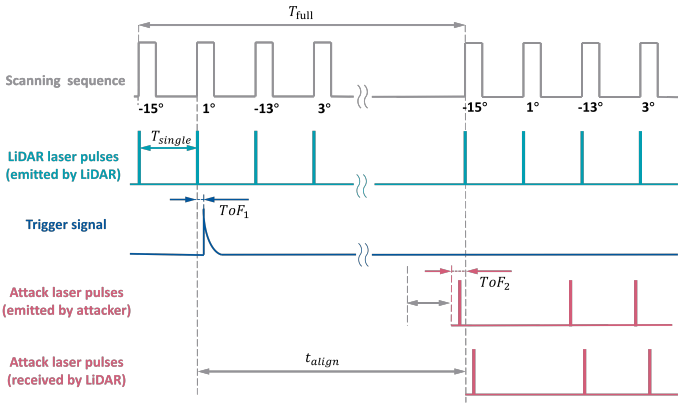


Fig. 6: **Synchronization of the VLP-16 scanning sequence and the attack laser pulses (received by VLP-16).**

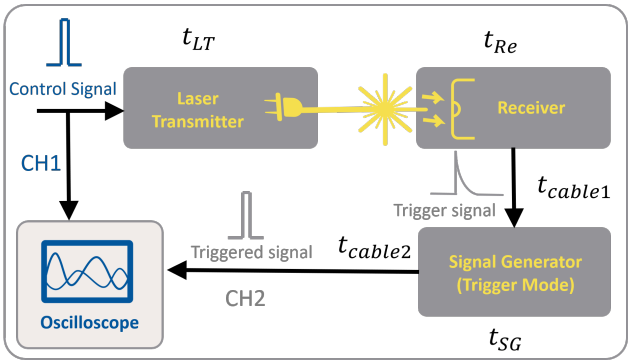


Fig. 7: **The measurement of inherent delay. A control signal is inputed into the workflow (laser transmitter ⇒ receiver ⇒ signal generator) and observed by the oscilloscope (CH1). And then, a triggered signal generateed by signal generator is oberseved through CH2 of the oscilloscope. The t_{device} is the delay between control signal and triggered signal, which can be measure by comparing the delay of CH1 and CH2.**

To address this issue, we propose our synchronization method. First, we obtain the scanning sequence and the sequence of LiDAR laser pulse of the victim LiDAR by user manuals [37], as shown in the first and second waveforms in Fig. 6. Here we take the VLP-16 LiDAR as an illustration to better present the alignment process. Then, we use the *receiver* to sense the specified working signal of the victim LiDAR. As soon as the receiver receives a laser pulse from the victim LiDAR, it generates a trigger signal as shown in the third waveform in Fig. 6, by which we can acquire the LiDAR's scanning status. We then transmit the trigger signal to the delay controller. After a precise delay (t_{delay}), the signal generator generates a control signal to control the laser transmitter to emit attack laser pulses. Finally, the attack laser pulses are received by the victim LiDAR after a time of flight.

Therefore, to align the attack signal received by the victim LiDAR and the scanning sequence, the key point is to set the delay precisely. To achieve it, we put the receiver in the path where specified-vertical-angle lasers will irradiate (1° in Fig. 6), and calculate t_{delay} as follows:

$$t_{delay} = t_{align} - ToF_1 - ToF_2 - t_{device} \tag{11}$$

where t_{align} represents the duration from the moment the LiDAR sends out a specific laser pulse to the moment it receives an aligned attack laser pulse. Therefore, $t_{align} = n * T_{fc} - T_{sfc}$ when receiving 1° Laser pulse as shown in Fig. 6. ToF_1 represents the flight time of a specific LiDAR laser pulse from the LiDAR to the receiver. ToF_2 represents the flight time of an attack laser pulse from the laser transmitter to the LiDAR. t_{device} represents the inherent delay of the device including signal response, laser charging, transmission of electrical signals in copper wires, etc. The method to measure t_{device} is shown in 7. A high-sample-rate oscilloscope is needed to measure the delay.

V. Attack Device Implementation and Point Injection Capability Evaluation

A. Attack Device Implementation

Based on the work of Shin et al. [33], we implement the attack setup consisting of four components: a receiver, a delay controller, a control signal generator, and a laser transmitter, as shown in Fig. 8. The receiver consists of a PIN photodiode [8] and an amplifier. Both the delay controller and control signal generator are integrated into an arbitrary waveform generator (AWG) [5]. The laser transmitter, which is significantly differs from previous work [33], [10] comprises of 3 components: a laser driver board, a laser diode, and a two-lens system. The specific models of those components can be substituted according to demand. During attacks, the receiver first receives laser pulses from the victim LiDAR and generates a trigger signal. Then, the AWG introduces a certain delay and generates a control signal. Finally, the laser transmitter fires attack lasers to the victim LiDAR and inject spoofing points.

B. Laser Systems Modeling

In this work, a laser system can be modeled with the following two types of parameters: fundamental laser parameters and final system parameters.

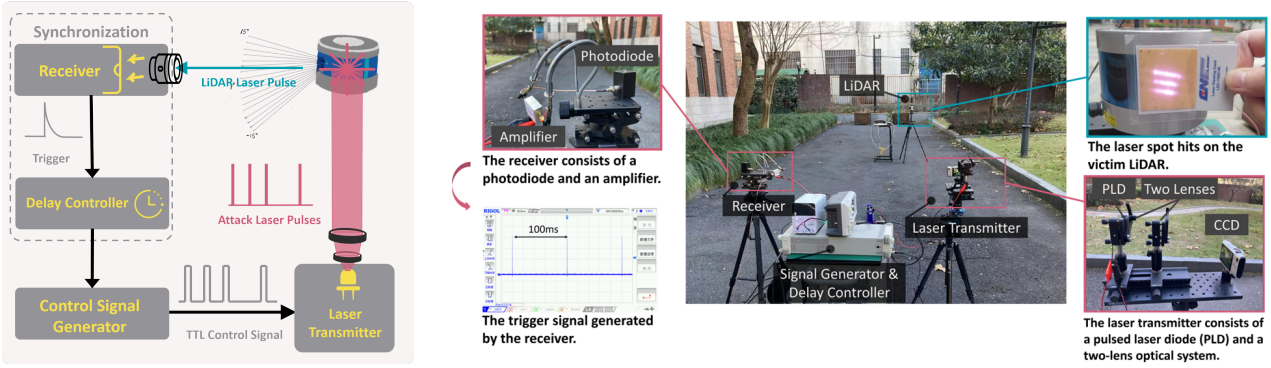
Fundamental laser parameters are the basic concepts of pulsed laser and are critical when choosing the laser transmitter. They are laser wavelength (Notation: λ_{laser}, Units: nm), peak power (Notation: P_{peak}, Units: W), and pulse repetition rate (Notation: f_{rep}, Units: Hz).

Final system parameters are target-related and describe the performance at the output of a laser system when hits on the target. They are spot size (Notation: S, Units: cm^2), attack distance (Notation: d, Units: m), and peak power intensity (Notation: I_{peak}, Units: W/cm^2).

C. Point Injection Capability Investigation

In this work, we comprehensively evaluate the attack capability across three dimensions: the number of points, position control capability, and shape control capability. Additionally, we analyze the factors that influence these capabilities.

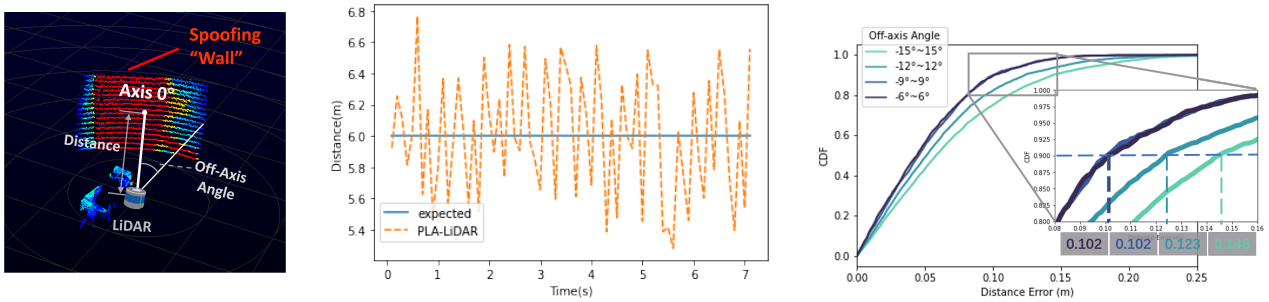
1) Points Number: The Number of injectable spoofing points has been the primary metric for measuring attack capability in previous works [33], [10], [34]. The previous state-of-the-art work [34] has demonstrated the ability to inject up to 200 points into the VLP-16. To investigate the feasibility of improving point injection capability, we conducted experiments specifically targeting the VLP-16. It is important to note



(a) Illustration of physical attack process

(b) Physical experiment setup consisted of a receiver, an arbitrary signal generator, a laser transmitter, and a victim LiDAR.

Fig. 8: **Illustration of the attack process and the experimental setup for physical attacks.**



(a) Spoofing "wall" injected by laser attacks

(b) Distance Control

(c) Shape Control: CDF of distance errors.

Fig. 9: **Point Injection Capability Investigation.**

that this metric is only meaningful when considering the same LiDAR model.

Experiment Measurement. For investigating the feasibility of point injection capability improvement, we conduct experiments against VLP-16 with λ_{laser} = [850 nm, 905 nm, 915 nm, and 940 nm], P_{peak} = [25W, 75W, 125W, 300W, 600W], f_{rep} = [0 to 800kHz]. Through experiments, We find that the laser can inject spoofing points of various shapes (shown in Appendix) into the LiDAR. Among those shapes, the "wall" as shown in Fig. 9(a) has the greatest number of controllable spoofing points (4800 points when the LiDAR's rotation speed is 300 RPM) and the largest attack area (30° horizontal angle * 30° vertical angle).

Analysis. The principle of injecting points is to make our attack signal appear as the echo of the LiDAR's operating signal. There are four key factors for successful point injection:

- **Waveform.** The waveform of attack signal should be identical to the LiDAR's signal in order to be recognized as a valid echo.
- **Wavelength.** The wavelength of the attack signal should be similar to or identical to the LiDAR's signal. Otherwise, it will be filtered out.
- **Peak Power Intensity.** The signal strength needs to be sufficiently strong. According to the return mode of the LiDAR, signals that are too weak will be considered environmental noise and will be filtered out.
- **Repetition Frequency.** In order to inject more points, we aim to inject attack signal at every *receiving time* of the LiDAR. Therefore, the repetition frequency of the signal needs to match the scanning sequence of the LiDAR.

Empirically, right waveform, appropriate wavelength, high peak power and precise repetition frequency will lead to the most injected points.

2) Position Control: The attacker typically aims to inject point clouds at specific position to achieve the intended attack objectiveness, whether it is hiding or creating objects. Position control capability refers to the attacker's ability to precisely control the overall position of the injected point clouds.

Experiment Measurement. We measure the position control capability by quantifying the difference between the centroid position of the injected point cloud and the desired target position. We designate a point as $p = (r, \theta, \phi)$,where r is the distance, θ is the vertical angle, and ϕ is the horizontal angle. The centroid of the point cloudwhich contains n points, is denoted as $p_c = ((r_c, \theta_c, \phi_c)$ and can be calculated as follows:

$$p_c = \frac{1}{n} * \sum_{i=0}^{n} p_i \tag{12}$$

In this experiment, we aimed to inject the point cloud with $p_c = (6m, 0, 0)$. We collected continuous data for 7 seconds for analysis. We found that we can achieve continuous and precise control over the vertical and horizontal angle of the point cloud. However, there is some error in controlling the distance as shown in Fig. 9(b), resulting in slight fluctuations

of the injected point cloud over time. The standard deviation of distance error is 0.38 meters.

Analysis. Based on the above experiment, we can conclude that the position control capability is primarily manifested in the control of distance. We suppose that the error in distance control is mainly caused by light noise and instrument jitters. During synchronization, we encounter the "curse of light" issue, where even a small timing error multiplied by the speed of light can result in a significant distance error. In our experiment, the distance control has an error of approximately 0.38 meters, which translates to a time error of only about 1.26 nanoseconds. We suppose that this error can be mitigated by applying noise reduction techniques.

3) Shape Control: During an attack, attackers typically aim to inject point clouds with specific shapes, such as pedestrians or carefully crafted adversarial point cloud. Therefore, the ability to control the shape of the injected point cloud is an important metric for assessing attack capabilities.

Experiment Measurement. We measure the shape control ability using the distance error of each point. In this experiment, we aim to design the attack signal in such a way that the distance of all points is 10 meters. As a result, we were able to inject a "wall" shape, as shown in the Fig. 9(a). First, we define the central axis of the spoofing "wall" as 0°, and divide the point cloud area by the off-axis angle (as shown in Fig. 9(a)). Then, we calculate the average distance of the points in each point cloud area, and use the average distance as the ground truth to calculate the distance error of each point. Finally, we can get the CDF (Cumulative Distribution Function) map as shown in Fig. 9(c), which is described the distance errors distribution of point clouds area containing different off-axis angles. We found that the distance error is various across different off-axis angles. In general, the closer to the 0° axis, the smaller the error. When it comes to the near central area, the CDF of error is almost the same, *e.g.* the distance error of 90% points in -9°∼9° area and -6°∼6° area is both within 0.102 meters.

Analysis. A smaller distance error indicates stronger shape control capability. Overall, we can inject points covering above 30° horizontal angle, and have relatively precise control over points within around a 20° horizontal angle. We assume the distance error of injected points relies on two reasons: (1) The optical path difference due to different incident angles and curved structure of the mechanical LiDAR. (2) The error induced by the limited sample rate of the signal generator (1GHz for DG5072 in this experiment). We suppose that this error can be mitigated by applying high sampling rate signal generator.

VI. Evaluation

In this section, we evaluate our attacks against LiDAR-based 3D object detection systems. We consider three sets of evaluations in this paper: (1) Simulation evaluation for Opt-Hide and Opt-Create, where the adversarial point clouds fed into the 3D object detectors are generated by optimization directly. (2) Physical evaluation for all 4 types of attacks, i.e., Nai-Hide, Rec-Create, Opt-Hide, and Opt-Create, on 2 mechanical LiDARs and 3 detectors, where the point clouds fed into the 3D object detectors are generated by the victim LiDAR physically. (3) Physical evaluation against fusion-based models. (4) Feasibility study of physical attacks when the attacker and victim are in motion.

TABLE I: **Top-1 success rates of simulated optimization-based attacks across various numbers of spoofing points.**

Attack	LiDAR	Detector	Category of Object			Avg.	Overall.
			Ped.	Cyc.	Car.		
Opt-Hide	VLP16	PoinPillar	100%	99%	88%	95.7%	68.00%
		SECOND	48%	35%	38%	40.3%	
	HDL64E	PoinPillar	100%	100%	100%	100.0%	79.67%
		SECOND	80%	59%	39%	59.3%	
Opt-Create	VLP16	PoinPillar	64%	64%	33%	53.7%	60.85%
		SECOND	95%	98%	11%	68.0%	
	HDL64E	PoinPillar	77%	81%	24%	60.7%	72.67%
		SECOND	93%	97%	64%	84.7%	

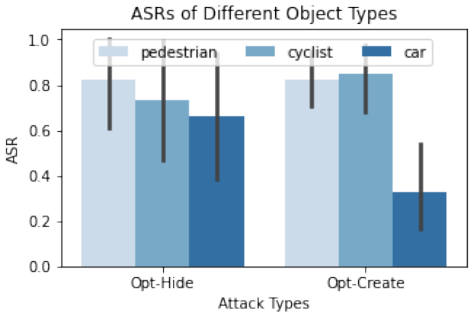


Fig. 10: **The attack success rates of Opt-Hide and Opt-Create attacks against various objects.**

A. Simulation Evaluation

1) Setup: In this section, we present the experimental setup of the simulation evaluation.

Victim LiDARs. We evaluate simulated attacks against a 16-line LiDAR VLP-16 and a 64-line LiDAR HDL-64E.

Object Detectors. We evaluate our attacks using two 3D object detectors PointPillar [22] and SECOND [40]. We use the implementation from MMDetection3D [13]. The average detection precision achieved on KITTI is 59.5% for PointPillar and 64.41% for SECOND.

Dataset. We use the KITTI [15] dataset in the simulation evaluation, which is widely used in the training and testing of 3D object detectors.

Classes of Interest. Given that most LiDAR-based 3D object detectors detect (up to) three classes of objects in the autonomous driving scenarios, i.e., (1) car, (2) pedestrian, and (3) cyclist, we consider them as classes of interest in this paper. Both the aforementioned object detectors PointPillar [22] and SECOND [40] support the detection of these three classes.

2) Evaluation Methodology: We use the attack success rate (ASR) as the metric, which is the ratio of the number of successful attacks against an object detector over the total number of conducted attacks. For Opt-Hide attacks, we randomly select 100 objects for each class of interest from the KITTI dataset and try to make them undetectable. For Opt-Create attacks, we randomly select 100 scenarios from

TABLE II: **The attack success rates of physical attacks against various LiDARs and object detectors.**

Detector	LiDAR Model	Attack Types			
		Nai-Hide	Rec-Create	Opt-Hide	Opt-Create
SECOND	**VLP-16**	100%	98%	47%	75%
	RS-16	100%	86%	46%	66%
PoinPillar	**VLP-16**	100%	64%	78%	24%
	RS-16	100%	51%	74%	19%
Apollo	**VLP-16**	100%	98%	81%	39%
	RS-16	100%	89%	76%	24%

the KITTI dataset, and try to inject a car, a pedestrian, or a cyclist into each scenario, respectively.

3) Attack Effectiveness: The attack results under various numbers of spoofing points are shown in Appendix. Since we can inject up to 4,200 spoofing points, we consider the highest attack success rate under various points as shown in Tab. I. The overall ASRs are 73.84% for Opt-Hide attacks and 66.76% for Opt-Create attacks, indicating that it is easier to hide an existing object than to create a non-existing one.

For different types of objects, our attacks perform better in hiding and creating pedestrians and cyclists compared with cars as illustrated in Fig. 10. Specifically, Opt-Hide attacks can hide all three types of objects with an ASR above 65%. Opt-Create attacks work well (above 82%) in generating pedestrians and cyclists, but have a relatively low ASR (33%) in generating cars. We suppose the variation in the ASR is due to the larger size of the car, necessitating a broader search space and more iterations to attain a relatively high success rate when attempting to create a car. To validate our hypothesis, we conducted additional experiments and investigation. (1) we tested different iterations (300, 500, 800, 1500, 2000, and 2500) for cars and found that the attack success rate increased with the number of iterations and reached saturation at 2000. By employing more iterations, we can improve the attack success rate of cars to 56%. (2) Creating a car requires a larger empty space in the original frame since it has a larger size compared with the pedestrian or cyclist. By investigating 100 randomly-selected frames from the dataset KITTI, we found that this requirement was not always met. Some of those frames might contain environmental point clouds in the target space (10 meters directly in front of the LiDAR) where we intended to inject spoofing point clouds. The frames that may not meet this requirement exceeded 10/100 for cars and were only around 5/100 for cyclists and pedestrians, leading to the lower attack success rates for cars.

B. Physical Evaluation

In the physical evaluation, we conduct Nai-Hide, Rec-Create, Opt-Hide, and Opt-Create attacks by physically injecting spoofing points into LiDARs.

1) Setup: In this section, we present the experimental setup of the physical evaluation.

Attack Signal Design As a pre-step to the physical world experiments, we need to obtain the desired point cloud in advance and design the attack signals accordingly. The point clouds for Nai-Hide and Rec-Create are obtained using substitute LiDAR recordings. For Nai-Hide, the original point cloud is a wall; for Rec-Create, the desired point cloud is a pedestrian. The desired point clouds for Opt-Hide and Opt-Create are generated by adversarial machine learning (detailed in Sec. IV-B2), where robustness enhancement is incorporated into the generation process and point clouds with high attack success rate in the digital world will be adopted for the physical attack.

Attack Device Setup. We use the attack device setup shown in Fig. 8, where we position the victim LiDAR in front of our attack equipment at varying distances and angles. We list the models of all equipment used in our experiments on the website. Based on the device setup, we conducted physical experiments on campus roads.

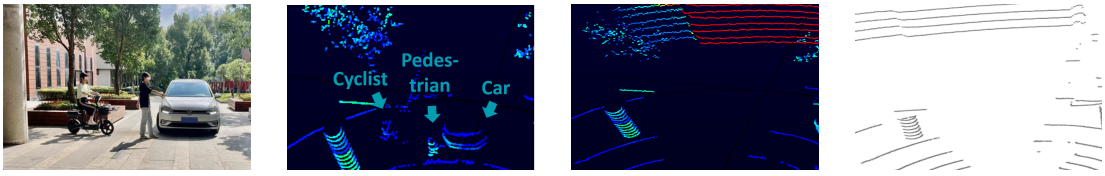
Victim LiDAR and Detectors. We conduct physical attacks on 2 mechanical LiDARs, i.e., VLP-16 and RS-16, which are the most widely used mechanical LiDARs in the world. The point clouds generated by the LiDARs under attacks are directly fed into two academic detectors SECOND [40] and PointPillars [22], and a commercial detector Apollo r6.5 [1].

2) Evaluation Methodology: For Nai-Hide and Opt-Hide attacks, we try to hide a pedestrian, a cyclist or a car. For Rec-Create and Opt-Create attacks, we try to create a non-existing pedestrian. For each attack, we randomly collect 100 frames for different LiDARs and different detectors, and use the ASR as the metric to report attack effectiveness.

3) Attack Effectiveness: In total, we collect 2400 frames during physical attacks. The overall performance of the physical attacks is shown in Tab. II. The videos of the physical attacks can also be found on the website.

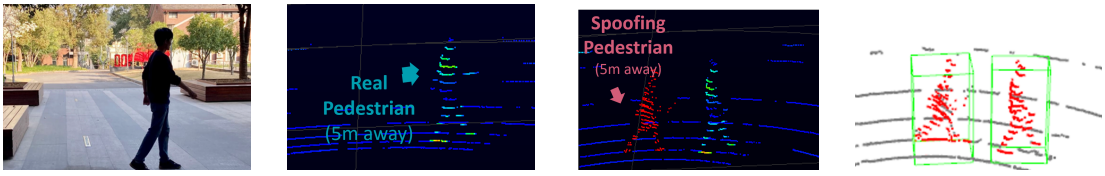
Impact of LiDAR model. For different LiDAR models, the attack performance shows slight differences. Specifically, the average ASRs on VLP-16 is 75.33%, while it is 69.25% on RS-16. The reason is that RS-16 has pulse randomizing technology which can eliminate the attack. Every about a hundred full cycles (55.555 μs), there will be a silent period of about 133 μs, bringing extra difficulty in precisely injecting the spoofing point clouds.

Impact of detection system. The attack performance varies across different detection systems. Nai-Hide attacks can achieve 100% ASRs against all three detection systems. Rec-Create attacks perform better on SECOND and Apollo. It is because SECOND and Apollo perform better in detecting a real pedestrian, while the spoofing point cloud we injected via Rec-Create attacks is very similar to the point cloud of a real person. Opt-Hide attacks perform better on PointPillars and Apollo, while Opt-Create attacks perform better on SECOND. We suppose the performance difference may come from the different feature extraction processes of these three models, i.e., SECOND divides the point cloud space into voxels while PointPillars and Apollo divide the point cloud space into vertical columns (pillars) and utilize PointNets to learn features. For Opt-Hide attacks, the location where we add the adversarial points is the space above the victim object, thus the adversarial points are more likely to harm the feature extraction of the point cloud below the pillar. For Opt-Create attacks, we optimize point clouds in a cuboid space, which



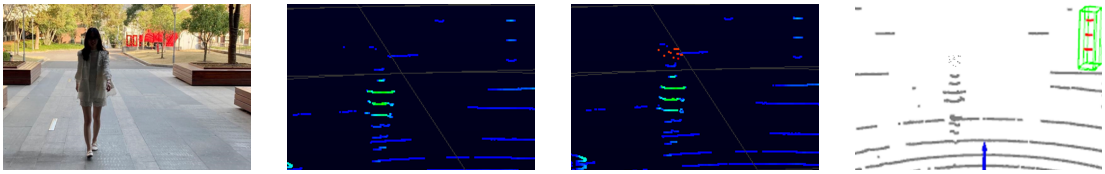
(a) Ground truth. (b) Benign point cloud. (c) Point cloud under attack. (d) Detection under attack.

Fig. 11: **Illustration of the naive hiding attack.**



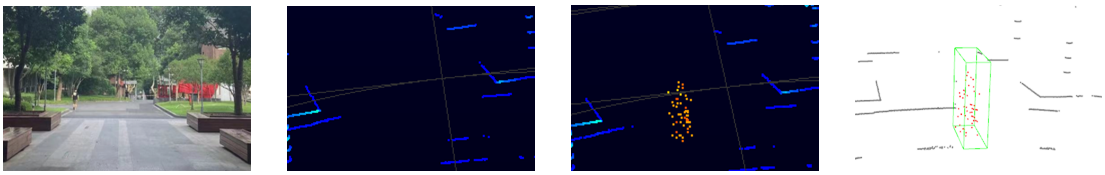
(a) Ground truth. (b) Benign point cloud. (c) Point cloud under attack. (d) Detection under attack.

Fig. 12: **Illustration of the record-based creating attack.**



(a) Ground truth. (b) Benign point cloud. (c) Point cloud under attack. (d) Detection under attack.

Fig. 13: **Illustration of the optimization-based hiding attack.**



(a) Ground truth. (b) Benign point cloud. (c) Point cloud under attack. (d) Detection under attack.

Fig. 14: **Illustration of the optimization-based creating attack.**

is more similar to the voxels rather than pillars, making SECOND more vulnerable.

4) Attack Robustness: We then investigate the attack robustness when the laser source is at various distances, heights, and angles with the VLP-16 LiDAR and the SECOND detector. For Nai-Hide and Opt-Hide attacks, we try to hide a real pedestrian 5 meters in front of the LiDAR. For Rec-Create and Opt-Create attacks, we try to create a pedestrian 5 meters in front of the LiDAR.

Impact of attack distance. We conduct experiments with attack distances of 1, 3, 5, 10, and 15 meters. We collect 20 frames for every distance and every attack type (400 frames in total) to report the attack success rates. The results in Fig. 15 show that the ASRs of Rec-Create, Opt-Hide and Opt-Create attacks decrease as the attack distance increases, while the Nai-Hide attacks can still achieve 100% ASR. We assume the reason is that the laser power intensity becomes lower as the distance increases, rendering the spoofing point cloud difficult to control. For the four types of attacks, Nai-Hide attacks have the lowest requirement on the control accuracy of the spoofing points, and thus are least affected by the attack distance.

Impact of LiDAR's installation height. We conduct experiments with various LiDAR installation heights of 0.2, 0.7, 1.2, 1.7, and 2.2 meters. We collect 20 frames for every LiDAR installation height and every attack type (400 frames in total) to report the attack success rates. The results in Fig. 15 show the ASRs of our attacks at different LiDAR's installation heights. Among the four types of attacks, the performances of Nai-Hide and Rec-Create attacks do not significantly change with the LiDAR installation height, while Rec-Create and Opt-Create attacks show the highest ASRs at a LiDAR installation height of 1.7m. It is probably because the training dataset KITTI [16] was collected by LiDARs installed at a height of 1.73 m. As a result, optimization-based attacks are more sensitive to the LiDAR's installation height.

Impact of attack angle. We investigate the attack's effectiveness when the laser source is at various horizontal and vertical angles. For hiding attacks, we adopt the Nai-Hide attack and try to hide a real pedestrian 5 meters away whose horizontal angle is about 0°. For creating attacks, we adopt the Rec-Create attack and try to create a fake pedestrian 5 meters away at a horizontal angle of around 0°. We collect 20 frames for every attack angle and every attack type (1080 frames in total) to report the attack success rates. The results shown in Fig. 17(a) and Fig. 17(b) demonstrate that both attacks are more affected by the horizontal angle than the vertical angle. Nai-Hide attacks mainly succeed within a horizontal angle within [-15°,15°]. We assume the reason is that the receiving angle of the LiDAR's receiver (mainly composed of a photodiode and a lens) is limited. Rec-Create attacks mainly succeed in a horizontal angle within [-10°, 10°], which is smaller than the hiding attack. We assume the reason is that

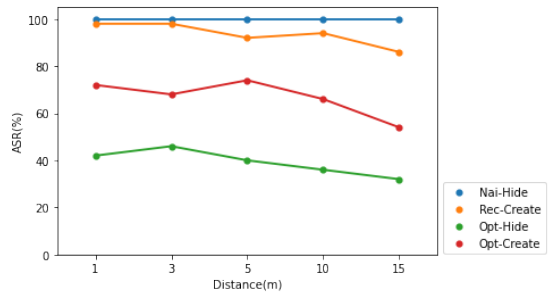


Fig. 15: **The attack success rates of physical attacks across various attack distances.**

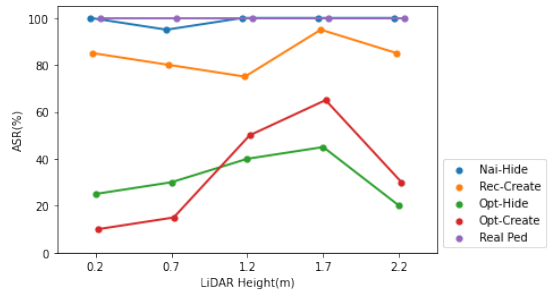
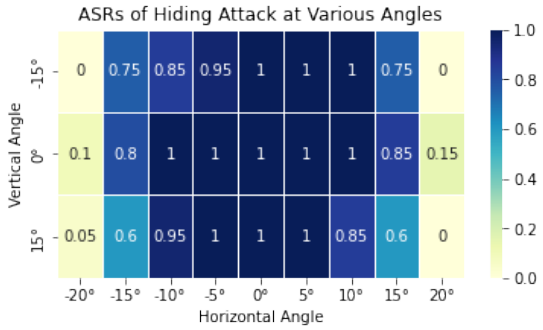
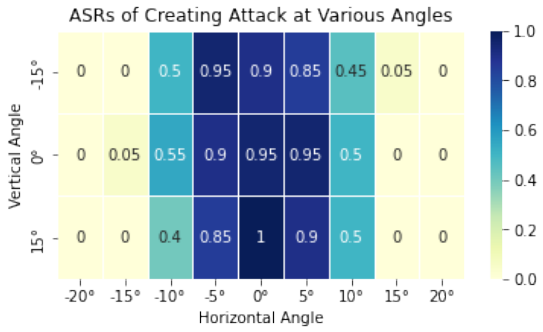


Fig. 16: **The attack success rates of physical attacks across various LiDAR's installation heights.**



(a) Results of Hiding attack at various angle.



(b) Results of creating attack at various angle.

Fig. 17: **The attack success rates of (a) hiding and (b) creating attacks across various attack angles.**

Rec-Create attacks require fine-grained control over the shape and distance of the point cloud while the points injected within the region of [-10°, 10°] show low errors as shown in Fig. 9.

C. Effectiveness on Fusion-based Model

The above experiments validate the performance of our attacks against stationary LiDARs. In this section, we test PLA-LiDAR against camera-LiDAR fusion models. The Camera and LiDAR are two complementary sensor types for 3D object detection. The camera captures color information that can be used to extract rich semantic features, while the LiDAR excels at localization and provides comprehensive information on 3D structures. Sensor fusion may provide additional robustness against naive black-box attacks on AV perception modules [17].

TABLE III: **The attack success rates of physical attacks against various Camera-LiDAR fusion-based detectors.**

Attack Types	Fusion Models			
	F-Pointnet	EPnet	AVOD	CLOCs
Hide	100%	100%	100%	100%
Create	0%	86%	83%	0%

1) Setup: Nai-Hide and Rec-Create attacks are evaluated against camera-LiDAR fusion systems.

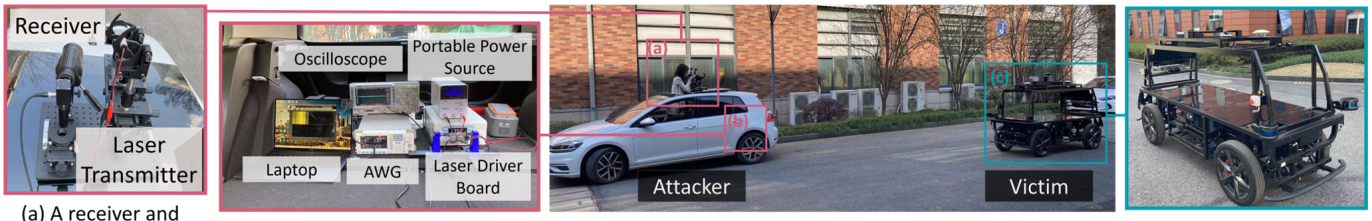
Hardware. The attack device setup is illustrated in Figure 8. The Camera-LiDAR fusion platform comprises a camera and a VLP-16 LiDAR on an Apollo-kit. Prior to executing the attack, we calibrated the camera and LiDAR to ensure that the image is aligned with the point cloud.

Model. We evaluate our attack on popular camera-LiDAR fusion models with four different architectures.

- **F-PointNet** [28] is a cascaded level fusion model that leverages a image detector to predict 2D proposals, each of which is then transformed to 3D space in order to extract corresponding frustum candidates, followed by a point cloud detector for segmentation and detection from the points in frustum.
- **EPnet** [18] is a feature-level fusion model that fuses image and LiDAR features in the backbone networks.
- **AVOD** [21] is a feature-level fusion model that conducts multi-modal feature fusion at the proposal generation and RoI refinement stage.
- **CLOCs** [27] is a data-driven result-level fusion model that fuses the outputs of 2D and 3D detections beofore Non-Maximum Suppression(NMS) via exploiting the consistency of geometries and semantics.

2) Evaluation Methodology: For hiding attacks, our strategy involves concealing a specific target located in front of the victim car. Conversely, for orchestrating attacks, we attempt to generate an illusory pedestrian. For every individual attack, we randomly assemble a set of 100 frames composed of both images and point clouds. These collected data are then directly supplied to various fusion models, accompanied by the calibration matrix. We employ the Attack Success Rate (ASR) as the metric to gauge the efficacy of our attacks. We adopt the standard Intersection-over-Union (IoU) thresholds for each model, namely 0.7 for vehicles, and 0.5 for pedestrians and cyclists.

3) Atttack Effectiveness: The overall performance of the the four fusion models is shown in TABLE. III. The hiding attack achieves a 100% ASRs for all four models. This high effectiveness is attributed to the formidable nature of hiding attacks, capable of almost entirely erasing an object's



(a) A receiver and a laser transmitter on the car roof.

(b) The attack equipment in the car trunk.

Setup of moving experiment .

(c) The victim car (Apollo KiT) with a VLP-16.

Fig. 18: **Experimental setup for physical attacks on moving vehicles.**

3D information. This action consequently prevents successful regression of the 3D bounding box.

The creation attack yields a relatively high success rate for two feature-level fusion models(EPnet and AVOD), while it is virtually ineffective against the other two(F-Pointnet and CLOCs). The failure of the creating on F-PointNet can primarily be attributed to its image detection-based filtering mechanism; the "pedestrian" we create within the point cloud lacks a corresponding pattern in the image and hence is filtered out. For CLOCs, the reason of low ASRs lies in the fact that late fusion mainly prunes rather than creating new detections. Therefore, even if our attack successfully deceives the LiDAR detector, the absence of a corresponding detection candidate in the 2D detector ultimately results in the false obstacle being overlooked by the fusion network. On the other hand, for AVOD and EPNet, the primary reason for the success of the attacks lies in the dominant role of point clouds within these feature-level fusion models, a finding corroborated by some literature. Consequently, altering the point cloud alone can have an impact on the final output results.

D. Feasibility Study on Moving Vehicle

The above experiments validate the performance of our attacks against stationary LiDARs. Then, we explore the feasibility of our attacks when the victim LiDAR is in motion.

Experimental Setup. We define an attack setting of moving vehicles as shown in Fig. 18, where both the attacker car and the victim car are moving with a similar speed of around 5km/h (for safety reasons), and the attacker is 5-15 meters away from the victim LiDAR. We integrate the attacking equipment into the attacker car by placing the receiver and laser transmitter on the car roof (each connected to a gimbal for manual aiming) and placing other attack devices such as the arbitrary waveform generator (AWG), laptop, laser driver board, and power source in the car trunk. The victim car is an Apollo D-kit equipped with a VLP-16 LiDAR, and the point cloud collected by the LiDAR is used for real-time 3D object detection.

Compared to the experimental setup for the stationary attack, we upgrade the attack hardware to mitigate the effects of jitters caused by the moving of the vehicles: (1) We use a large-diameter telescope ($\Phi = 50\ mm$) to expand the receiver's receiving area from 0.2 cm^2 to 78.5 cm^2. (2) We expand the spot diameter to 8 cm, and use a high-power laser diode (P_{peak} = 300 W) to ensure the peak power intensity is greater than 2 W/cm^2. With a larger receiving area and light spot, even a slight jitter in the process of vehicle driving will not affect the effectiveness of the attack.

Results. Both hiding and creating attacks can successfully spoof the LiDAR-based 3D object detection system on a moving vehicle. Specifically, hiding attacks can achieve an ASR of 94.1% (16/17 trials) and creating attacks can achieve an ASR of 78.9% ASR (15/19 trials). The videos of physical attacks on moving vehicles can be found on the website. We also tested our attacks when the victim is moving while the attacker is stationary in Appendix.

VII. Defense

In this section, we propose a defense method (named PLA-Defense) that includes sensor fusion (*Sen-Fusion*) and spoofing points detection (*Spoof-Det*). The combination of these two components effectively mitigates PLA-LiDAR attacks. The defense workflow is shown in Fig. 19. First, we employ pseudo points technology to achieve data-level sensor fusion, enhancing the model's robustness. Additionally, inspired by the attack capability investigation in Sec. V-C, we design a curvature-based spoofing points detection method, which is a posterior, plug-and-play detection approach.

A. Methodology

1) Sensor Fusion: Our evaluation in Sec. VI-C indicates that existing fusion methods are ineffective against Nai-Hide attacks. Therefore, it is necessary to design a new fusion paradigm to address this limitation. The Nai-Hide attack has the capability to directly erase point clouds, leading to the complete loss of 3D information in specific regions, which compromises the results of object detection. Inspired by pseudo points technology, we recognize that 3D information is implicitly embedded in 2D images. Consequently, we propose generating pseudo point clouds from images and fusing it with the real point cloud at the data level to restore the erased information. We employ PSMNet [12] for pseudo points generation, as it can generate pseudo points using only stereo images. This approach offers two advantages: 1) It addresses the issue of heterogeneity between image and point cloud data, which prevents direct fusion; 2) It allows the direct use of the point cloud processing pipeline for data handling, facilitating integration. It is important to note that methods like SFD [39], which use point clouds to assist in generating pseudo points from images, are not applicable to our defense, as the point cloud has already been compromised.

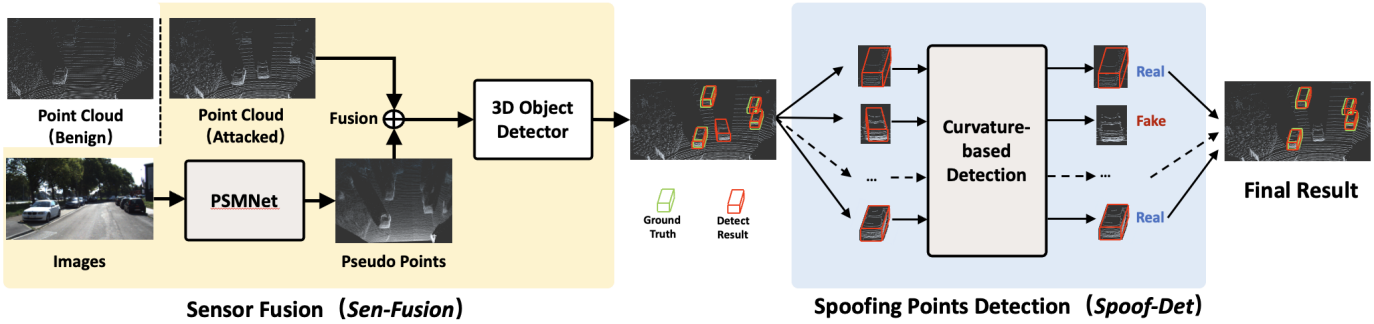


Fig. 19: **The Workflow of PLA-Defense.**

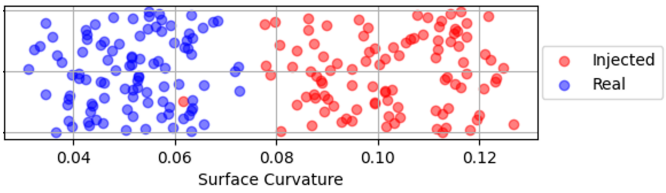


Fig. 20: **Surface Curvature of Real and Injected Objects.**

2) Spoofing Points Detection: Although above sensor fusion method can mitigate Nai-Hide attacks, its effectiveness against Rec-Create attack is limited, as the injected spoofing points may still be detected. According to the attack capability investigation in Sec. V-C, we know that while it is possible to inject point clouds with specific shapes (e.g., a pedestrian), there are inherent limitations in attack precision, leading to discrepancies between the shape of the injected object and that of a real object, as illustrated in the Fig. 12(c). These differences can be quantified and distinguished using the surface curvature [32] of the point cloud.We first utilize local neighborhood information captured by K-nearest neighbors to estimate the curvature at each point. Specifically, the method computes the covariance matrix of each point's neighborhood, extracts its eigenvalues, and derives the curvature as the ratio of the smallest eigenvalue to the sum of all eigenvalues. The average of these values provides the final surface curvature.

We calculated the surface curvature for 100 real objects, including cars and pedestrians, as well as 100 injected objects, with the results shown in the Fig. 20. It is evident that the surface curvature of real objects is significantly lower than that of injected objects. Therefore, we can employ a rule-based approach for defense: After the fusion model outputs its detection results, we extract the detected objects, calculate their surface curvature, and filter them using an empirical threshold (denoted as SC_{th}).

B. Evaluation

We conducted experiments to demonstrate the effectiveness of PLA-Defense to mitigate PLA-LiDAR attacks. We implemented sensor fusion and integrated attack detection based on the SECOND model. The overall results are shown in Table IV. The results indicate that the Sen-Fusion module provides strong defense against Nai-Hide and Opt-Hide attacks and also offers some defense against Opt-Create attacks, owing to the influence of pseudo points on the adversarial attack's effectiveness. In terms of Spoof-Det, the hyperparameters were set to K = 30, $SC_{th} = 0.076$. The Spoof-Det module exhibits strong defense against Rec-Create and Opt-Create attacks. The PLA-Defense is designed to combine the strengths of both modules, effectively defending against all four types of PLA-LiDAR attacks.

TABLE IV: **The Evaluation Results of PLA-Defense.** The ASR of *PLA-LiDAR* attacks on the detection model(SECOND) with or without the *Sen-Fusion* and *Spoof-Det* defense methods.

Attack Types	Attack Success Rate (ASR)			
	w/o Defense	Sen-Fusion only	Spoof-Det only	PLA-Defense
Nai-Hide	100%	7%	100%	**7%**
Rec-Create	98%	91%	1%	**1%**
Opt-Hide	38%	2%	38%	**2%**
Opt-Create	72%	32%	0%	**0%**

VIII. Discussion of Potential Mitigation

Our attacks exploit the vulnerabilities of mechanical LiDARs and mislead the 3D object detection algorithms to ultimately affect the decisions. In this section, we provide several potential defense mechanisms by increasing the difficulty of launching our attacks.

Rotation Speed Customization. Our attacks rely on the RPM of the victim LiDAR to design the control signals. If the RPM used for control signal design is different from the one used in the victim LiDAR, the injected point cloud will be deformed and thus possibly invalid (shown in appendix). Therefore, the users can manually set the RPM of the LiDAR from time to time. Although the adversary can still measure it using photodiodes and oscilloscopes, this method can increase the attack overhead in terms of both cost and time.

LiDAR Pulse Coding and Randomizing. Another exploited vulnerability is that most LiDARs receive laser pulses without verification. We envision it can be mitigated by applying a laser pulse coding technique, which will increase the spoofing difficulty. For instance, Kim et al. [19], [20] have proposed a LiDAR verification scheme, which encodes the pixel location information in the laser pulses using the direct-sequence optical code division multiple access (DS-OCDMA) method. In addition, randomizing the emitting pulses and rejecting pulses different from the emitted ones is another potential defense method, and similar approaches have been studied for military radars [26]. However, pulse coding and

pulse randomization may decrease the robustness and increase the cost of LiDARs.

Multi-sensor Fusion and Security Redundancy. Another complementary defense approach is to exploit multi-sensor fusion for decision-making. Autonomous vehicles can employ multiple types of sensors, e.g., cameras, radars, ultrasonic sensors combined with LiDARs to perceive the environment. Such information fusion and redundancy may help further improve the security of autonomous vehicles.

IX. Conclusion

In this paper, we investigate the possibility of physically fooling LiDAR-based 3D object detection by injecting adversarial point clouds into it using lasers. By carefully measuring the victim LiDARs, delicately designing laser signals, and emitting them in a precise delay, we achieve to inject spoofing point cloud with desired shapes into the victim LiDAR, and hide or create object against 3D detectors in the physical world. Evaluations with two widely-used mechanical LiDARs and three 3D object detectors demonstrate the effectiveness of our attacks. Further directions include exploring the vulnerabilities of other types of LiDARs.

References

[1] Apollo. https://github.com/ApolloAuto/apollo.

[2] Apollo robotaxi. https://www.apollo.auto/robotaxi/index.html.

[3] Arcfox baic hbt. https://www.techgenyz.com/2021/04/07/huaweis-lidar-solution-arcfox-baic-hbt-car/.

[4] A guide to lidar wavelengths for autonomous vehicles and driver assistance. https://velodynelidar.com/blog/guide-to-lidar-wavelengths/.

[5] Rigol dg5072 arbitrary waveform generator. https://www.batronix.com/shop/waveform-generator/Rigol-DG5072.html.

[6] Waymo driver. https://waymo.com/waymo-driver/.

[7] Waymo one. https://waymo.com/waymo-one/.

[8] S5971 si pin photodiode. https://www.newark.com/hamamatsu/s5971/diode-photo-900nm-to-18-3/dp/62M0262, 2021.

[9] Markus-Christian Amann, Thierry M Bosch, Marc Lescure, Risto A Myllylae, and Marc Rioux. Laser ranging: a critical review of unusual techniques for distance measurement. *Optical engineering*, 40:10–19, 2001.

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

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

[12] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5410–5418, 2018.

[13] MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d, 2020.

[14] Jin Fang, Ruigang Yang, Qi Alfred Chen, Mingyan Liu, Bo Li, et al. Invisible for both camera and lidar: Security of multi-sensor fusion based perception in autonomous driving under physical-world attacks. *arXiv preprint arXiv:2106.09249*, 2021.

[15] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[16] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

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

[18] Tengteng Huang, Zhe Liu, Xiwu Chen, and Xiang Bai. Epnet: Enhancing point features with image semantics for 3d object detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, pages 35–52. Springer, 2020.

[19] Gunzung Kim and Yongwan Park. Lidar pulse coding for high resolution range imaging at improved refresh rate. *Optics express*, 24(21):23810–23828, 2016.

[20] Gunzung Kim and Yongwan Park. Independent biaxial scanning light detection and ranging system based on coded laser pulses without idle listening time. *Sensors*, 18(9):2943, 2018.

[21] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018.

[22] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019.

[23] LeiShen. V2x roadside perception system. http://www.lslidar.com/en/solution/41, 2021.

[24] Inc Neuvition. Cvis and v2x with lidar. https://www.neuvition.com/media/cvis-and-v2x-with-lidar.html, December 2020.

[25] Ouster. Ouster for intelligent transportation systems. https://ouster.com/resources/smart-infrastructure-resources/its-lidar-solution-overview/, 2021.

[26] Phillip E Pace. *Detecting and classifying low probability of intercept radar*. Artech House, 2009.

[27] Su Pang, Daniel Morris, and Hayder Radha. Clocs: Camera-lidar object candidates fusion for 3d object detection. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10386–10393. IEEE.

[28] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018.

[29] Quanergy. Quanergy s3.

[30] Inc. Robosense LiDAR. *RS-16*, 2022.

[31] Inc. Robosense LiDAR. *RS-M1*, 2022.

[32] Philip Schneider and David H Eberly. *Geometric tools for computer graphics*. Elsevier, 2002.

[33] Hocheol Shin, Dohyun Kim, Yujin Kwon, and Yongdae Kim. Illusion and dazzle: Adversarial optical channel exploits against lidars for automotive applications. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 445–467. Springer, 2017.

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

[35] James Tu, Mengye Ren, Sivabalan Manivasagam, Ming Liang, Bin Yang, Richard Du, Frank Cheng, and Raquel Urtasun. Physically realizable adversarial examples for lidar object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13716–13725, 2020.

[36] Velodyne. Smart city. https://velodynelidar.com/industries/smart-city/, 2021.

[37] Inc. Velodyne LiDAR. *VLP-16 Data Sheet*, 2018.

[38] Inc. Velodyne LiDAR. *VLP-16 User Manual*, 2019.

[39] Xiaopei Wu, Liang Peng, Honghui Yang, Liang Xie, Chenxi Huang, Chengqi Deng, Haifeng Liu, and Deng Cai. Sparse fuse dense: Towards high quality 3d detection with depth completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5418–5427, 2022.

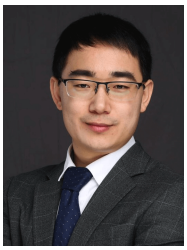
[40] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.

[41] Yole Déppement (Yole). Lidar for automotive and industrial applications 2020, August 2020.

[42] Yole Déppement (Yole). Lidar for automotive and industrial applications 2021, September 2021.

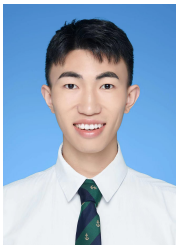
[43] Yi Zhu, Chenglin Miao, Tianhang Zheng, Foad Hajiaghajani, Lu Su, and Chunming Qiao. Can we use arbitrary objects to attack lidar perception in autonomous driving? In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 1945–1960, 2021.

Zizhi Jin received his B.S. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 2020. He is currently working toward the PhD degree in the College of Electrical Engineering at Zhejiang University. His research interests include cyber-physical system (CPS) security, with a particular focus on autonomous driving security.



Xiaoyu Ji received his B.S. degree in Electronic Information & Technology and Instrumentation Science from Zhejiang University, Hangzhou, China, in 2010. He received his Ph.D. degree in Department of Computer Science from Hong Kong University of Science and Technology in 2015. From 2015 to 2016, he was a researcher at Huawei Future Networking Theory Lab in Hong Kong. He is now an associate professor with the Department of Electrical Engineering of Zhejiang University. His research interests include IoT security, including sensor, network, and AI security. He won the best paper award of ACM CCS 2017, ACM AsiaCCS 2018. He is a member of IEEE.

Yushi Cheng received her B.S. degree in Electrical Engineering from Zhejiang University in 2016, and her Ph.D. degree in Control Science and Engineering from Zhejiang University in 2021. She now is a postdoctoral researcher with the Department of Automation of Tsinghua University. Her research interests include IoT security, AI security, and mobile & ubiquitous computing. She received a WST best paper runner-up award in 2017, and an ASIACCS best paper award in 2018.



Bo Yang received the B.S. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 2021. He is currently working toward the Master's degree in the College of Electrical Engineering at Zhejiang University. His research interests include sensor security and IoT security.

Chen Yan received the B.S. degree in Electrical Engineering in 2015 and the PhD degree in Control Theory and Engineering in 2021 from Zhejiang University, Hangzhou, China. He is currently an assistant professor at the College of Electrical Engineering, Zhejiang University. His research interests include sensing security, CPS security, and IoT security. He received the Best Paper Award of ACM CCS in 2017 and the Doctoral Dissertation Award of ACM China in 2021. He was acknowledged by Tesla Motors in the Security Researcher Hall of Fame in 2016.

Wenyuan Xu is currently a professor in the College of Electrical Engineering at Zhejiang University. She received her B.S. degree in Electrical Engineering from Zhejiang University in 1998, an M.S. degree in Computer Science and Engineering from Zhejiang University in 2001, and the Ph.D. degree in Electrical and Computer Engineering from Rutgers University in 2007. Her research interests include wireless networking, network security, and IoT security. Dr. Xu received the NSF Career Award in 2009, a CCS best paper award in 2017, and an ASIACCS best paper award in 2018. She was granted tenure (an associated professor) in the Department of Computer Science and Engineering at the University of South Carolina in the U.S. She has served on the technical program committees for several IEEE/ACM conferences on wireless networking and security, and she is an associated editor of TOSN.