PLA-LiDAR: Physical Laser Attacks against LiDAR-based 3D Object Detection in Autonomous Vehicle

Zizhi Jin[1], Xiaoyu Ji[1†], Yushi Cheng[2], Bo Yang[1], Chen Yan[1], Wenyuan Xu[1†]
[1]Ubiquitous System Security Lab (USSLAB), Zhejiang University
[2]Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University
{zizhi, xji, yushicheng, yb5, yanchen, wyxu}@zju.edu.cn

***Abstract*—Autonomous vehicles and robots increasingly exploit LiDAR-based 3D object detection systems to detect obstacles in 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, which is 20 times more than prior work, 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 also discuss defense strategies at the sensor and AV system levels.**

I. Introduction

The growth of autonomous vehicle (AV) solutions is the catalyst for the adoption of LiDAR (Light Detection And Ranging) being incorporated into advanced driving assistance systems (ADAS) [1], [3], [6] and cooperative vehicle infrastructure systems (CVIS) [29], [30], [25], [41] (Fig. 1). According to Yole Développement [49], the LiDAR market for automotive and industrial applications is expected to reach US $3.8 billion in 2025. By providing precise 3D point clouds of surrounding environments, the LiDAR and the subsequent 3D object detection algorithms detect and classify obstacles on the roads to help AVs to make safety-critical driving decisions. As a result, correct detection and classification in the midst of a dedicated adversary is important to ensure safe driving.

Many prior works have demonstrated the vulnerabilities of LiDAR-based 3D object detection systems, but mostly by manipulating 3D point clouds digitally [11], [12], [40], [38]. Several works have attempted to generate 3D adversarial point clouds physically by placing 3D-printed obstacles in specific locations [16] or flying drones around the target objects [51]. However, those methods mainly focus on Denial of Service (DoS) attacks, and the adversarial object way is conspicuous to human eyes.

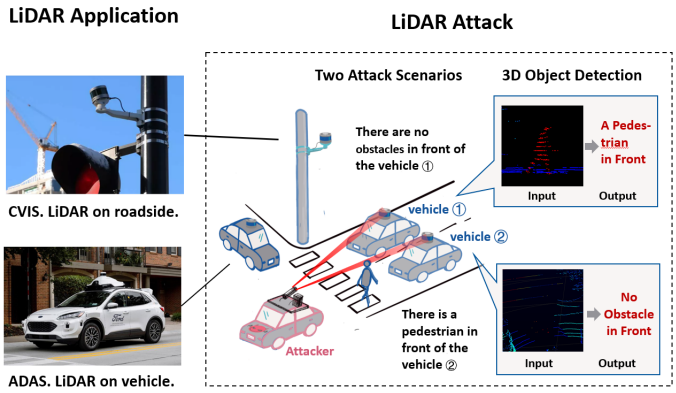


Fig. 1: **By injecting malicious laser signals into the LiDAR of the 3D object-detection system in autonomous vehicles, an attacker can fool decision-making.**

In this paper, we ask *"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, causing two types of errors, detecting a none-existent object or failing to detect an object ahead.

To the best of our knowledge, this is the first work focusing on physical attacks against LiDAR. Although prior work has shown that 3D object detection systems are vulnerable to digital adversarial point clouds, none has investigated whether the generated adversarial point clouds can be physically received by the LiDAR. Such a goal is non-trivial because a LiDAR is continuously rotating in the horizontal plane and scanning at the vertical plane at a high speed. How to physically inject the point clouds into the LiDAR with the right shape and location in the presence of environment noises and device jitters are still not studied. Third, the capability of point cloud injection reported in previous work is 200 at most [38], which is not strong enough to achieve the aforementioned attacks. Whether

Demo https://sites.google.com/view/physical-lidar-attack
†Corresponding author

and how can we improve the point injection ability is still unknown.

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. 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 [43] and Robosense RS-16 [35] on two academic 3D object detectors PointPillar [24] and SECOND [48] 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 right 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).

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 [36] or Optical Phased Array [34] technology. In these two types of LiDARs, the mechanical one takes up over 95% share of the global LiDAR market in 2020 [50], 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.

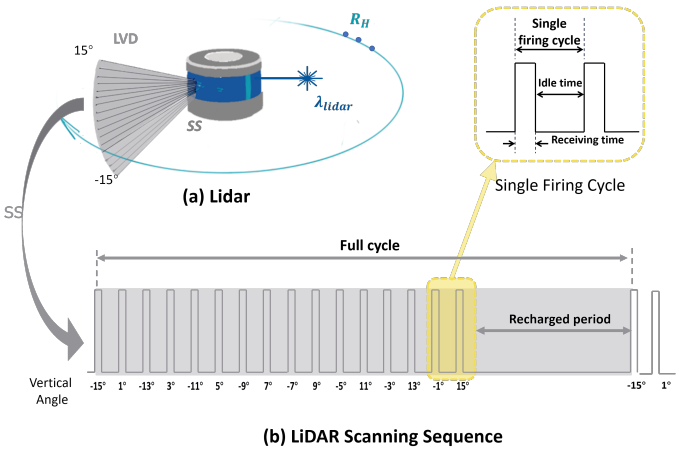


Fig. 2: **The (a) structure, and (b) scanning sequence of the VLP-16 LiDAR.**

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:

$$LiDAR = [SS, LVD, R_H, \lambda_{lidar}] \quad (1)$$

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 [24], Apollo [1], SECOND [48] in this paper.

III. Threat Model

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

A. Threat Model

1) 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.

2) 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.

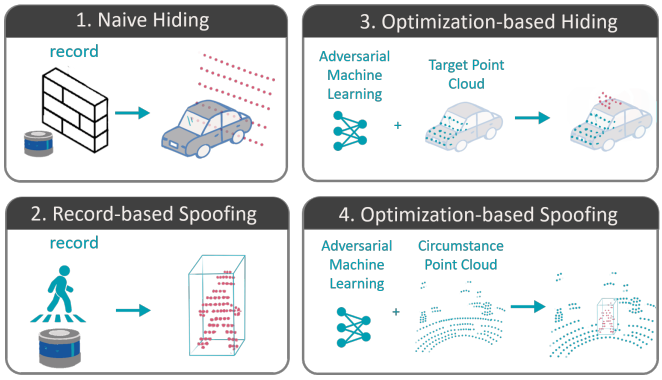


Fig. 3: **Four fine-grained attack types.**

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.

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 [10].

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.

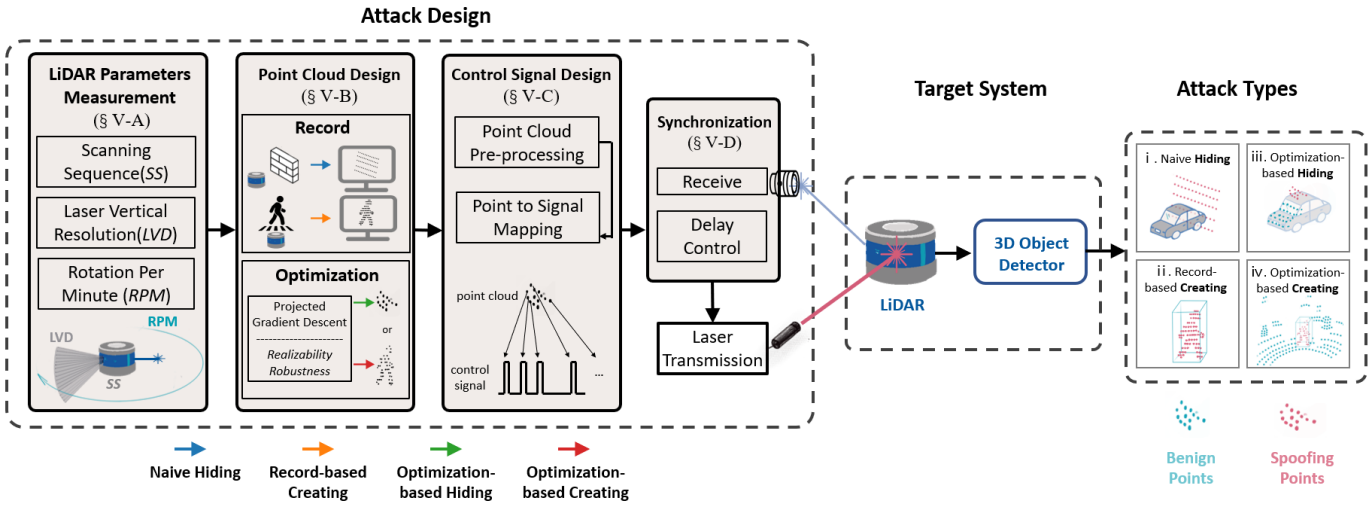
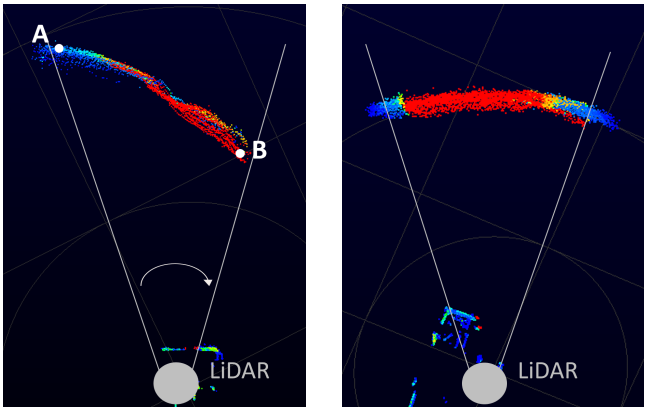


Fig. 4: **The attack workflow. By measuring the victim LiDAR, the adversary first generates injectable point clouds by recording or adversarial optimization, then converts the expected point clouds into control signals, and finally injects the lasers into the victim LiDAR by signal synchronization, which 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.**

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 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. 7 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:

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

where $\mathrm{P}^{'} = \{(R_i^{'}, \alpha_i^{'}, \omega_i^{'}) | i \in [1,n]\}$ is the adversarial point cloud, $R_i^{'}$, $\alpha_i^{'}$ and $\omega_i^{'}$ are the distance, horizontal angle, and vertical 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.

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 to the spherical coordinates of the adversarial points;
- Step 4: Update R of the adversarial point cloud $P^{'}$ and add random noise to increase the robustness of adversarial points.
- 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

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 $\boldsymbol{Time\_ideal}$ as abscissa and $\boldsymbol{Amp\_ideal}$ as ordinate ;
$\boldsymbol{Signal}_{discrete} \leftarrow$AD sample the $\boldsymbol{Signal}_{ideal}$ with the sampling rate SR ;

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 control 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:

$$\textit{fullcycle\_id} = \frac{\phi - \phi_0}{r_H} \tag{7}$$

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:

$$\textit{tof} = 2 * \frac{r}{c} \tag{8}$$

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:

$$\begin{aligned}\boldsymbol{Timestamp} =& \boldsymbol{fullcycle\_ID} * T_{fc} + \\ & \boldsymbol{singlecycle\_ID} * T_{sfc} + \boldsymbol{ToF}\end{aligned} \tag{9}$$

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.

D. Synchronization

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

Prior work [37] has proposed a synchronization method to induce the spoofing points closer than the attacker, but it may not suffice for our attacks since it does not consider the alignment of the attack signal and scanning sequence.

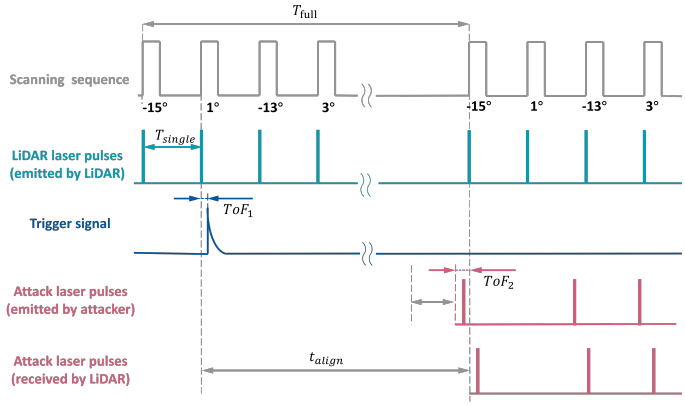


Fig. 6: **Synchronization of the VLP-16 scanning sequence and the attack laser pulses (received by VLP-16).**

To address it, 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 [42], 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:

$$t_{delay} = t_{align} - ToF_1 - ToF_2 - t_{device} \tag{10}$$

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} can be found in Appendix. F.

V. Attack Device Implementation and Point Injection Capability Evaluation

A. Attack Device Implementation

Based on the work of Shin et al. [37], 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. 7. The receiver consists of a PIN photodiode [8] and an amplifier. The delay controller and control signal generator are integrated in an arbitrary waveform generator (AWG) [5]. The laser transmitter, which is quite different with previous work [37], [11] consists of 3 components: a laser driver board, a laser diode, and a two-lens system. The specific models of those components can be replaced according to demand. During attacks, the receiver first receives laser pulses from the victim LiDAR and generates a trigger signal. Then, the AWG delivers a certain delay and generates a control signal. Finally, the laser transmitter fires attack lasers to the victim LiDAR.

B. Point Injection Capability Investigation

Although state-of-the-art work demonstrated that 200 points can be injected into VLP-16 at most [38], we believe that the number of spoofing points that can be injected should be far more than 200 due to LiDAR can often form hundreds of thousands of points per revolution. For investigating the feasibility of point injection capability improvement, we conduct experiments against VLP-16 with different laser transmitters. The point cloud injection capability mainly depends on the performance of the laser transmitter. Therefore, we first model the laser system, then test the point cloud injection capability of the laser system with different parameters experimentally, and finally we analyze the error of the injected point cloud.

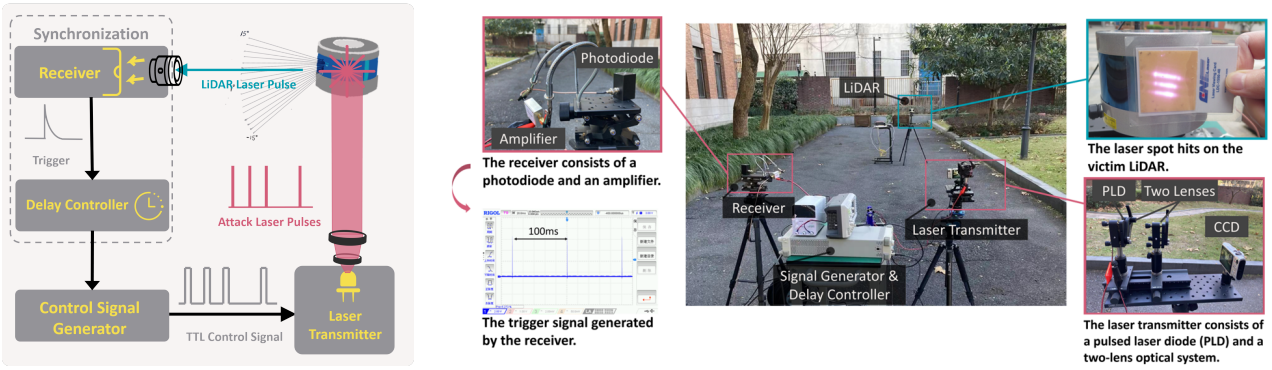
1) Laser Systems Modeling: In this work, a laser system can be modeled with the following two types of parameters: fundamental laser parameters and final system parameters.

Fundamental 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 hit 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).

2) Experiment: 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]. The pulsed laser diodes and laser driver boards that we used can be seen in Fig. 22 and Fig. 23 in Appendix C.

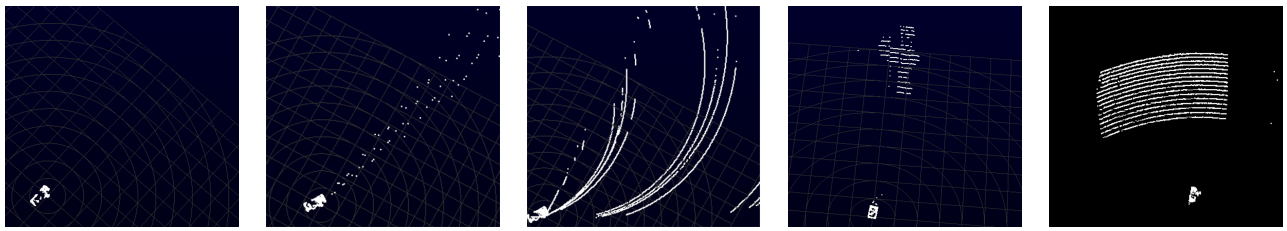
We find that the laser can inject spoofing points of various shapes (shown in Fig. 8) into the LiDAR, and the possible point cloud shapes include "scatter", "wave", "wall", and "broken wall". The relationship between these shapes and laser parameters is shown in Appendix B. Among those shapes, the "wall" has the greatest number of controllable spoofing points (4200 points when the LiDAR's rotation speed is 300 RPM) and the largest attack area (30° horizontal angle * 30° vertical angle). The "wall" can be generated when the λ_{laser} = 905 nm and f_{rep} = 434.02608 kHz (when $\triangle t$ = 2.304009 μs). Empirically, I_{peak} should be greater than 2



(a) Illustration of physical attack process

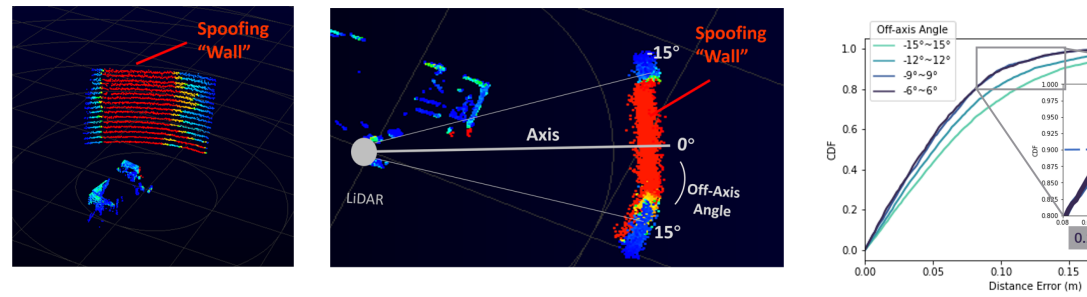
(b) Physical experiment setup consisted of a receiver, an arbitrary signal generator, a laser transmitter, and a victim LiDAR.

Fig. 7: **Illustration of the attack process and the experimental setup for physical attacks.**



(a) Oiginal (b) Scatter (c) Wave (d) Broken Wall (e) Wall

Fig. 8: **The point clouds injected into LiDAR under various laser systems parameters.**



(a) Spoofing "wall" injected by laser attacks

(b) Illustration of point cloud areas containing different off-axis angles

(c) CDF of distance errors.

Fig. 9: **Distance error distribution for point cloud areas containing different off-axis angles.**

W/cm^2 in order to generate a complete "wall". Therefore, a larger P_{peak} allows the final system to have a larger spot size and a longer distance. For example, using a laser diode with P_{peak} = 600W, we can generate a complete "wall" with the spot size of $100cm^2$ at a distance of $20m$ away. Surprisingly, we can still inject up to 4000 points with the same laser diode (SPL-PL90, P_{peak}=25W) used in previous work [37], [11], [38]. In the following experiments, we usually use a 120W laser diode (GYLDTO905120C) [9] due to the heat and safety risks associated with higher power lasers.

3) Distance Error Analysis of Spoofing Point Cloud: Although we can inject thousands of points within a 30° horizontal angle, not all points can be injected exactly where desired. When we inject a spoofing "wall", theoretically every spoofing point is designed to have the same distance from LiDAR and the "wall" is expected to be a sphere with LiDAR as the center. But by moving the "wall" to the near front (as shown in Fig. 9(b)), we found that the actual generated "wall" is distorted: the edge of the "wall" will be farther from the LiDAR than the middle of the "wall". By adjusting peak power and spot size, the distortion still exists. We assume the distortion relies on two reasons: (1) The optical path difference due to different incident angles. (2) The error induced by the limited sample rate of the signal generator (1GHz for DG5072). We use the distance error of each point to measure the accuracy of the point cloud. A smaller distance error means we can have more precise control over the point cloud. 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(b)). 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 for a "wall", 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. Overall, we can inject points covering above 30° horizontal angle, and have relatively precise control over points within around a 20° horizontal angle.

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) Feasibility study of physical attacks when the attacker and victim are in motion.

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 HD-L64E.

Object Detectors. We evaluate our attacks using two 3D object detectors PointPillar [24] and SECOND [48]. We use the implementation from MMDetection3D [14]. The average detection precision achieved on KITTI is 59.5% for PointPillar and 64.41% for SECOND.

Dataset. We use the KITTI [17] 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 [24] and SECOND [48] 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 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 Fig. 29 in Appendix J. 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 create a non-existing one.

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%	

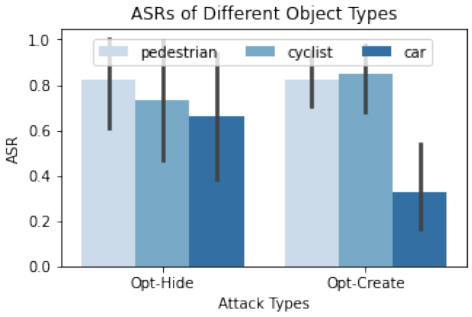


Fig. 10: **The attack success rates of Opt-Hide and Opt-Create attacks against various objects.**

For different types of objects, our attacks perform better in hiding and creating pedestrians and cyclists compared with cars, as shown 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. Through the further experiments detailed in Appendix. G, we find the reason of the ASR variation is that due to the larger size of the car, a larger search space and more iterations are required to achieve a relatively high success rate when creating a car.

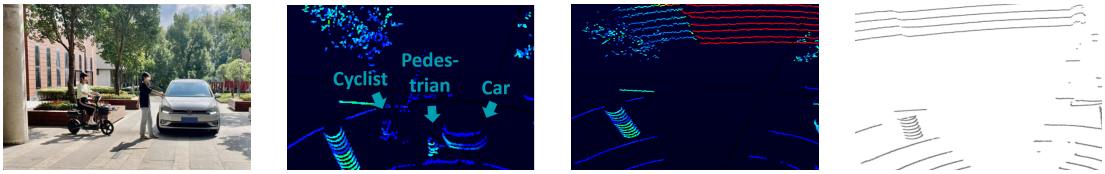
B. Physical Evaluation

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.

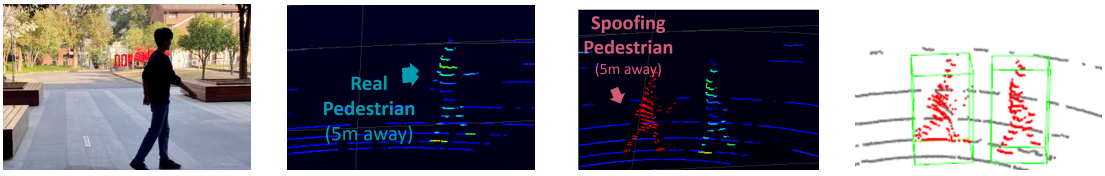
Attack Device Setup. We use the attack device setup shown in Fig. 7, where we position the victim LiDAR in front of our attack equipment at varying distances and angles. We list the models of all equipment used in our experiments on the website. Based on the device setup, we conducted physical experiments 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 parameters of two victim LiDARs are shown in Tab. IV. The point clouds generated by the LiDARs under attacks are directly fed into two academic detectors SECOND [48] and PointPillars [24], and a commercial detector Apollo r6.5 [1].



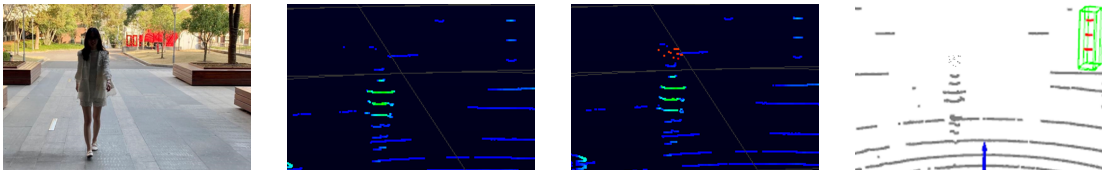
(a) Ground truth. (b) Benign point cloud. (c) Point cloud under attack. (d) Detection under attack.

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



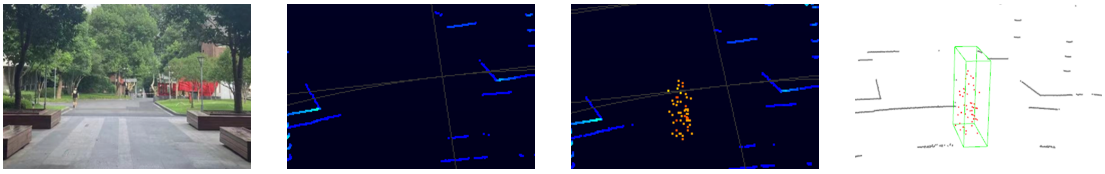
(a) Ground truth. (b) Benign point cloud. (c) Point cloud under attack. (d) Detection under attack.

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.**

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 70.75%, while it is 60.50% 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. For different detection systems, the attack performance also varies as shown in Fig. 15. 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 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

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%	38%	72%
	RS-16	100%	86%	33%	61%
PoinPillar	**VLP-16**	100%	64%	79%	15%
	RS-16	100%	51%	68%	12%
Apollo	**VLP-16**	100%	98%	77%	37%
	RS-16	100%	89%	73%	21%

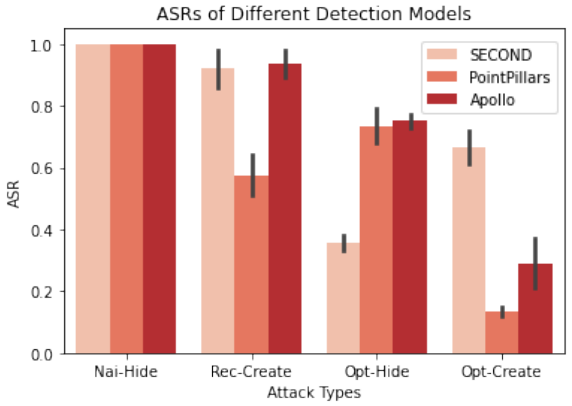


Fig. 15: **The attack success rates of physical attacks across various detectors.**

frames for every distance and every attack type (400 frames in total) to report the attack success rates. The results in Fig. 16 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. 16 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 [18] 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

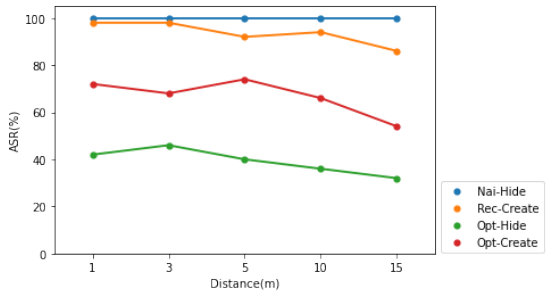


Fig. 16: **The attack success rates of physical attacks across various attack distances.**

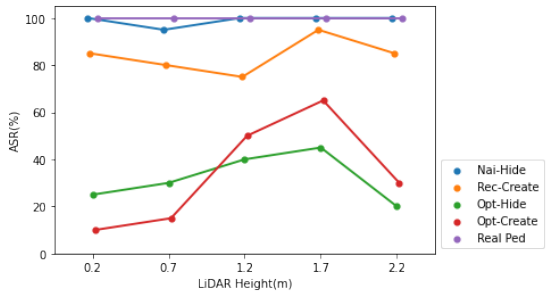
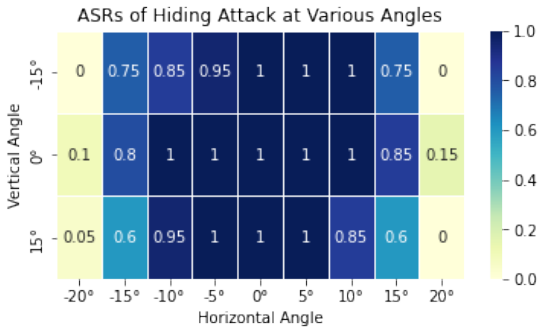
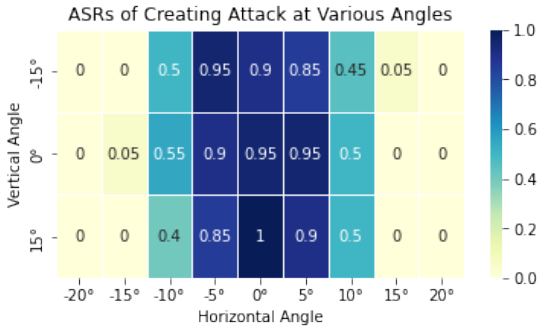


Fig. 17: **The attack success rates of physical attacks across various LiDAR's installation heights.**



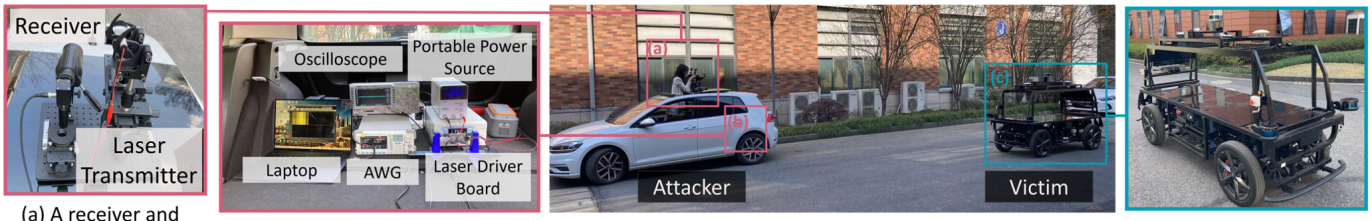
(a) Results of Hiding attack at various angle.



(b) Results of creating attack at various angle.

Fig. 18: **The attack success rates of (a) hiding and (b) creating attacks across various attack angles.**

in Fig. 18(a) and Fig. 18(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



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

(b) The attack equipment in the car trunk.

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

Fig. 19: **Experimental setup for physical attacks on moving vehicles.**

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 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. 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. 19, 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. I.

VII. Discussion

A. Comparison with Related Work

We compare our work with related work [37], [11], [38] in terms of the physical point injection capability, attack type, and attack performance, as shown in Tab. III. In summary, with improved point cloud injection capability and a new attack design for physical attacks, we can inject more points into the LiDAR and achieve more types of attacks with higher success rates.

Point cloud injection capability. With the more powerful lasers, we can inject up to 4200 spoofing points while prior work can inject up to 200 spoofing points against the same type of LiDAR (VLP-16).

Performance of simulated attacks. For simulated attacks, we can achieve an attack success rate of 94% with 180 points, while Cao et al. [11] can achieve a success rate of 75% with 60 points, and Sun et al. [37] can achieve a success rate of around 85% with more than 80 points.

Performance of physical attacks. Benefit from the hardware improvement and new attack flow design, including scanning sequence correction, physical-realizable point cloud design, new control signal design method, and precise synchronization. We are able to control the shape and location of spoofing points. We are the first work that spoofs 3D object detection by injecting spoofing points into a LiDAR physically while prior work has not conducted or reported the results of physical attacks in their papers.

B. 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 (see Fig. 24). 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

TABLE III: **Comparison with related work.**

Method	Target LiDAR	Attack Capability				Attack Effects		Simulation Attack		Physical-world Attack	
		Number of Spoofing Points	Distance Control	Patterns Control	Shape Control	Hide/DoS	Create	Object Type	Performance (success rate)	Spoofing points injection	Performance (success rate)
Shin et al. [37]	VLP-16	∼10 pts	●	–	–	●(by saturation)	–	–	–	●	–
Cao et al. [11]	VLP-16	∼100 pts	●	–	–	–	●	untargeted	75% (create, 60 pts)	●	–
Sun et al. [38]	VLP-16	∼200 pts	●	–	–	–	●	car	85% (create, 80 pts)	●	–
Ours	VLP-16, RS-16	∼4200 pts (300RPM)	●	●	●	●	●	targeted (3 objects)	94% (create, 180 pts) 100% (hide)	●	100% (Nai-Hide), 81% (Rec-create) 61% (Opt-Hide) 36% (Opt-Create)

● Applicable – Unknow/None

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. [22], [23] 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 [31]. 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.

C. Limitation

Our attacks still have the following limitations at present. First, we have successfully attacked mechanical LiDARs but have not succeeded on MEMS solid-state LiDARs with pulse coding technology yet. We assume that our attacks can also be applied to MEMS solid-state LiDARs by cracking the laser pulse code. We remain it as the future work. Second, though we have addressed the jitter issue for physical attacks to some extent by increasing the light spot and receiving area, it is still difficult to aim manually and continuously when the attacker and victim vehicles are relatively moving. We assume using detection and tracking techniques may be a potential solution.

VIII. Related Work

A. LiDAR Sensor Security in Autonomous Driving

Extensive studies have explored the sensor security problem in autonomous driving systems, and have identified a wide variety of vulnerabilities in cameras [21], [47], [28], [16], [27], [33], LiDARs [33], [37], [11], ultrasonic sensors [47], [15], [32], Radars [47], [26], etc.

Existing attacks against LiDARs can be classified into (1) object-based ones, and (2) laser-based ones according to the implementation methods. The object-based attacks generate adversarial points with 3D meshes [12], [40]. For instance, Fang et al. [16] proposed to place a well-designed 3D-printed adversarial object on the road and make it “invisible" to the object detection systems. Zhu et al. [51] proposed to attack LiDAR-based detection systems by placing commercial drones around adversarial locations of a car and hiding the victim car against LiDAR-based detection systems. However, these methods mainly focus on Denial of Service attacks and the adversarial object can be conspicuous to human eyes. The laser-based attacks aim to inject adversarial points into the LiDAR by infrared lasers. Petit et al. [33] and Shin et al. [37] have proved that LiDARs are vulnerable to laser attacks, and the attacker can inject spoofing points into LiDARs. Cao et al. [11] firstly adopted the optimization method to generate adversarial points, and found that merely 60 points were sufficient to create a car against Apollo 2.5 by simulation. Sun et al. [38] constructed the first black-box spoofing attack based on occlusion information. However, all laser-based adversarial attacks [11], [38], [20] are evaluated in simulation by far.

B. 3D Adversarial Machine Learning

Recently, much attention has been devoted to adversarial attacks that utilize the vulnerabilities of machine learning algorithms. Researchers have proposed various ways to construct adversarial examples (images) that can cause misclassification in 2D image classification and object detection [39], [19], [13]. With the rapid development of 3D perception, the adversarial machine learning against LiDAR-based 3D object detection begins to draw attention as well [44], [45], [46], [40]. Several prior works have designed and evaluated 3D adversarial attacks in the physical world [51], [16] by printing adversarial 3D objects. By contrast, we are the first to conduct physical adversarial attacks against LiDAR-based 3D detection model using lasers.

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 vulnerabilities ofLiDARs of other types.

X. Acknowledgments

We thank the anonymous shepherd and reviewers for their valuable comments. We would like to thank Weiying Kong, Linghao Zhang, Mengtong Yuan, Ziyuan Song and Yuxuan Zhujiang for providing valuable feedback on our work. This work is supported by National Natural Science Foundation of China (NSFC) Grant 61925109, 62071428, 62222114 and China Postdoctoral Science Foundation Grant BX2021158.

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] Gyldto905120c 120w laser diode. http://www.laser-opto.com/app/web/index/cs?id=1079_22b5018d-9f72-11ec-9e81-00163e0dae94_0_0, 2022.

[10] 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.

[11] 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.

[12] 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.

[13] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Proceedings of the 38th IEEE Symposium on Security and Privacy (S&P'17)*, pages 39–57. IEEE, 2017.

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

[15] Raj Gautam Dutta, Xiaolong Guo, Teng Zhang, Kevin Kwiat, Charles Kamhoua, Laurent Njilla, and Yier Jin. Estimation of safe sensor measurements of autonomous system under attack. In *Proceedings of the 54th Annual Design Automation Conference 2017*, pages 1–6, 2017.

[16] 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.

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

[18] 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.

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

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

[21] Xiaoyu Ji, Yushi Cheng, Yuepeng Zhang, Kai Wang, Chen Yan, Wenyuan Xu, and Kevin Fu. Poltergeist: Acoustic adversarial machine learning against cameras and computervision. *algorithms*, 47(26):11.

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

[23] 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.

[24] 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.

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

[26] Noriyuki Miura, Tatsuya Machida, Kohei Matsuda, Makoto Nagata, Shoei Nashimoto, and Daisuke Suzuki. A low-cost replica-based distance-spoofing attack on mmwave fmcw radar. In *Proceedings of the 3rd ACM Workshop on Attacks and Solutions in Hardware Security Workshop*, pages 95–100, 2019.

[27] Nir Morgulis, Alexander Kreines, Shachar Mendelowitz, and Yuval Weisglass. Fooling a real car with adversarial traffic signs. *arXiv preprint arXiv:1907.00374*, 2019.

[28] Ben Nassi, Dudi Nassi, Raz Ben-Netanel, Yisroel Mirsky, Oleg Drokin, and Yuval Elovici. Phantom of the adas: Phantom attacks on driver-assistance systems. *IACR Cryptol. ePrint Arch.*, 2020:85, 2020.

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

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

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

[32] Jonathan Petit and Steven E Shladover. Potential cyberattacks on automated vehicles. *IEEE Transactions on Intelligent transportation systems*, 16(2):546–556, 2014.

[33] Jonathan Petit, Bas Stottelaar, Michael Feiri, and Frank Kargl. Remote attacks on automated vehicles sensors: Experiments on camera and lidar. *Black Hat Europe*, 11(2015):995, 2015.

[34] Quanergy. Quanergy s3.

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

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

[37] 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.

[38] 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.

[39] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[40] 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.

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

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

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

[44] Yuxin Wen, Jiehong Lin, Ke Chen, and Kui Jia. Geometry-aware generation of adversarial and cooperative point clouds. 2019.

[45] Chong Xiang, Charles R Qi, and Bo Li. Generating 3d adversarial point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9136–9144, 2019.

[46] Chaowei Xiao, Dawei Yang, Bo Li, Jia Deng, and Mingyan Liu. Meshadv: Adversarial meshes for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6898–6907, 2019.

[47] Chen Yan, Wenyuan Xu, and Jianhao Liu. Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle. *Def Con*, 24(8):109, 2016.

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

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

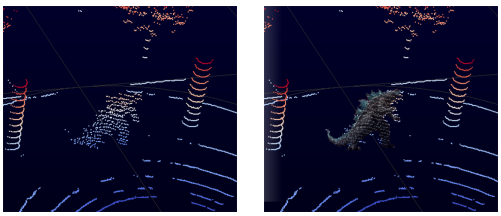
[50] Yole Déppement (Yole). Lidar for automotive and industrial applications 2021, September 2021.

[51] 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.

APPENDIX

A. Godzilla Spoofing

Fig.20 is a record-based spoofing point cloud of godzilla. Technically, any shape of point cloud we want can be injected, as long as its horizontal angle range is less than about 30°.



(a) Godzilla spoofing points (b) Godzilla picture

Fig. 20: **Record-based spoofing of Godzilla**

B. Feasibility Experiment

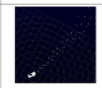
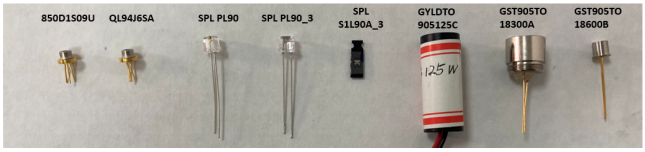
Shape of Spoofing Points		Possible Laser Parameters $[\lambda_{\text{laser}}, P_{peak}, f_{rep}]$
shape	Picture	
scatter		[905nm, 5mW, 0Hz], [915nm, 5mW, 0Hz], [940nm, 5mW, 0Hz]
Wave		[905nm, 5W~600W, 400kHz~434kHz]
Broken Wall		[940nm, 25W, 434.02608kHz], [905nm, 100mW, 434.02608kHz]
Wall		[905nm, 5W~600W, 434.02608kHz]

Fig. 21: **The relationship between the shapes of spoofing points and laser parameters.**

C. The Hardware Component Under Test

The laser diodes we tested are shown in Fig. 22 and the driver boards are shown in Fig. 23.



λ	850nm	940nm	905nm	905nm	905nm	905nm	905nm	905nm
Typ. P_{peak}	13W	50mW	25W	75W	120W	125W	300W	600W
Min Δt	1µs	0	1µs	1µs	0.5µs	1µs	1µs	1µs

Fig. 22: **The laser diode under test.**

D. The Influence of LiDAR Rotation Speed

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.

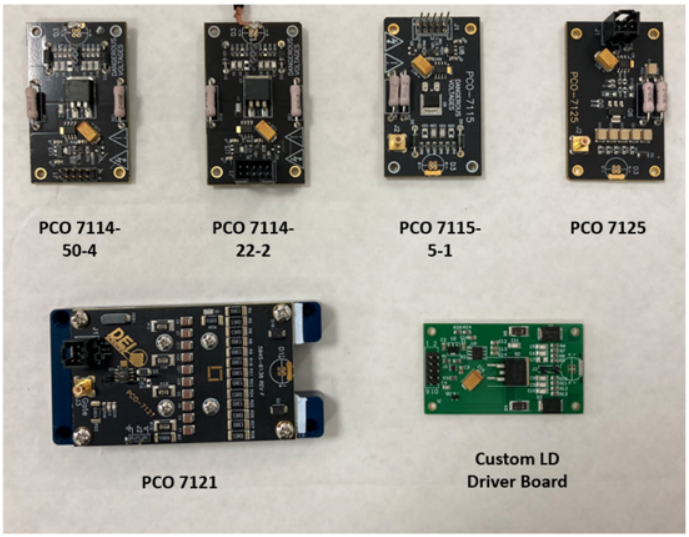


Fig. 23: **The laser diode driver under test.**

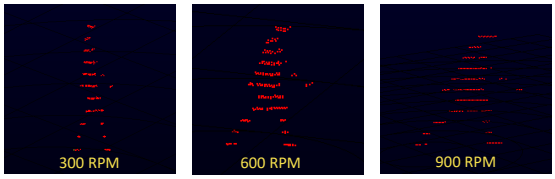


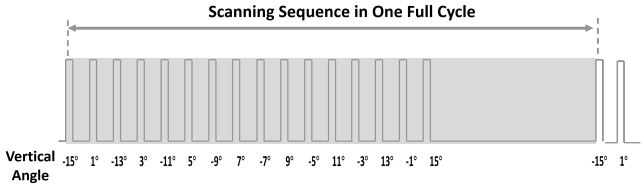
Fig. 24: **The pedestrian point clouds received by the VLP16 LiDAR at different rotation speeds under the same control signal designed for a PRM of 600.**

E. Angle to ID Mapping

As shown in Fig. 25, according to the scanning sequence of a LiDAR, a vertical angle to *laser_id* mapping ($Angle2ID$) can be established.

F. Measurement of Device Inherent delay

t_{device} needs to be measured in advance with an accuracy of at least 1 ns. The t_{device} is consisted of the delays from receiver (t_{Re}), signal generator (t_{SG}), laser transmitter (t_{LT}) and some cable delays. Since the input of the receiver and the output of the laser transmitter is optical signal instead



(a) The scanning sequence of VLP-16 in one full cycle

***Angle2ID* Mapping**

Vertical Angle	-15	-13	-11	-9	-7	-5	-3	-1	1	3	5	7	9	11	13	15
laser_id	0	2	4	6	8	10	12	14	1	3	5	7	9	11	13	15

(b) Vertical angle to *laser_id* mapping ($Angle2ID$) of VLP-16

Fig. 25: **According to (a) scanning sequence, (b) a vertical angle to *laser_id* mapping ($Angle2ID$) can be established.**

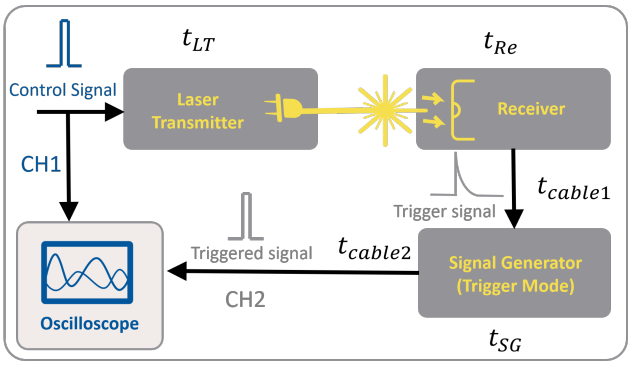


Fig. 26: **The measurement of inherent delay. A control signal is inputed into the workflow (laser transmitter ⇒ receiver ⇒ signal generator) and observed by the oscilloscope (CH1). And then, a triggered signal generateed by signal generator is oberseved through CH2 of the oscilloscope. The t_{device} is the delay between control signal and triggered signal, which can be measure by comparing the delay of CH1 and CH2.**

of electric signal that can be observed by oscilloscope. The measurement flow is shown in Fig. 26. A high-sample-rate oscilloscope is needed to measure the delay.

G. Further Experiments for the Performance Variation of Opt-Create

In this experiment, the success rate of creating a car was lower than that of a pedestrian and a cyclist. To reveal the reason, we conducted more experiments after receiving the review comments. Through experiments, we found two possible reasons for the low success rate of creating a car:

(1) We set a larger search space for a car due to its larger size compared with the pedestrian and cyclist, which required more iterations to achieve a better success rate. In our previous experiments, we set the search space for each target according to its average bounding box size, therefore, the search space of the car (2.6m*4.6m*1.7m) was much larger than that of the pedestrian (0.65m*0.8m,*1.7m) or the cyclist (0.65m*1.6m*1.7m). However, we used the same number of iterations (300 times) for each target, which resulted in a lower success rate of creating a car. In the new experiments, 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 approached saturation when it reached 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 the randomly-selected frames from the real-world dataset KITTI, we found this requirement was not always met. We found some of those frames might have environmental point clouds in the target space (10 meters directly in front of LiDAR) where we intend to inject spoofing point clouds. The frames may not meet the requirement are over 10/100 for cars and around 5/100 for cyclists and pedestrians, resulting in the lower attack success rates of cars.

H. The Influence of the Light Conditions

We tested the influence of light conditions by collecting VLP-16 frames in various light conditions: a) sunny day (200 frames), b) cloudy day (100 frames), c) night (200 frames). The results showed in Fig. 27 demonstrate no significant performance difference across different light conditions. It is because that both the attack laser and the working signal of the victim LiDAR are of 905 nm, and the LiDAR has its own filter mechanism for sunlight. Therefore, the light condition hardly impacts our attacks.

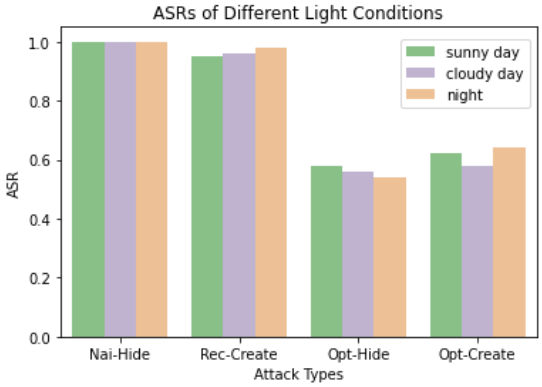


Fig. 27: **The influence of light conditions.**

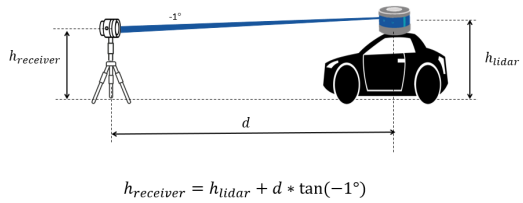
I. Supplementary Feasibility Study on Moving Vehicle

Experimental Setup. We conducted attacks in two moving scenarios: (1) Scenario A: Both the attacker and the victim 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. In this case, the main challenge is to overcome the jitters caused by the moving of the vehicles. (2) Scenario B: The victim is moving while the attacker is stationary. In this case, the main challenge is to aim the receiver with the laser transmitter in real-time.

Results. The results of scenario A are presented in Sec. VI-C. For Scenario B, however, we have not succeeded in spoofing the LiDAR detection result. Due to the fact that real-time aiming requires the receiver and laser transmitter to change the angle and height according to the position of the victim LiDAR. For our current aiming device, It is difficult to be achieved, especially when the laser (905nm) is invisible to human eyes. Although we have succeeded in manual aiming the laser transmitter to the victim LiDAR by using a large light spot and a 10x zoom CCD camera (for observing the light spot), manually aiming of the receiver is still challenging since the working signal of the LiDAR is too weak to be observed by the instrument such as CCD camera or night vision equipment. We assume a servo system (as shown in Fig. 28) can help address the real-time tracking and aiming problem but it is more like an engineering problem. We leave it as future work.

J. The Results of Simulation Evaluation

Optimization-based Hiding Attacks. The results shown in Fig.29(a). demonstrate the effectiveness of Opt-Hide simulation. When against VLP-16 and PointPillars, the ASRs for



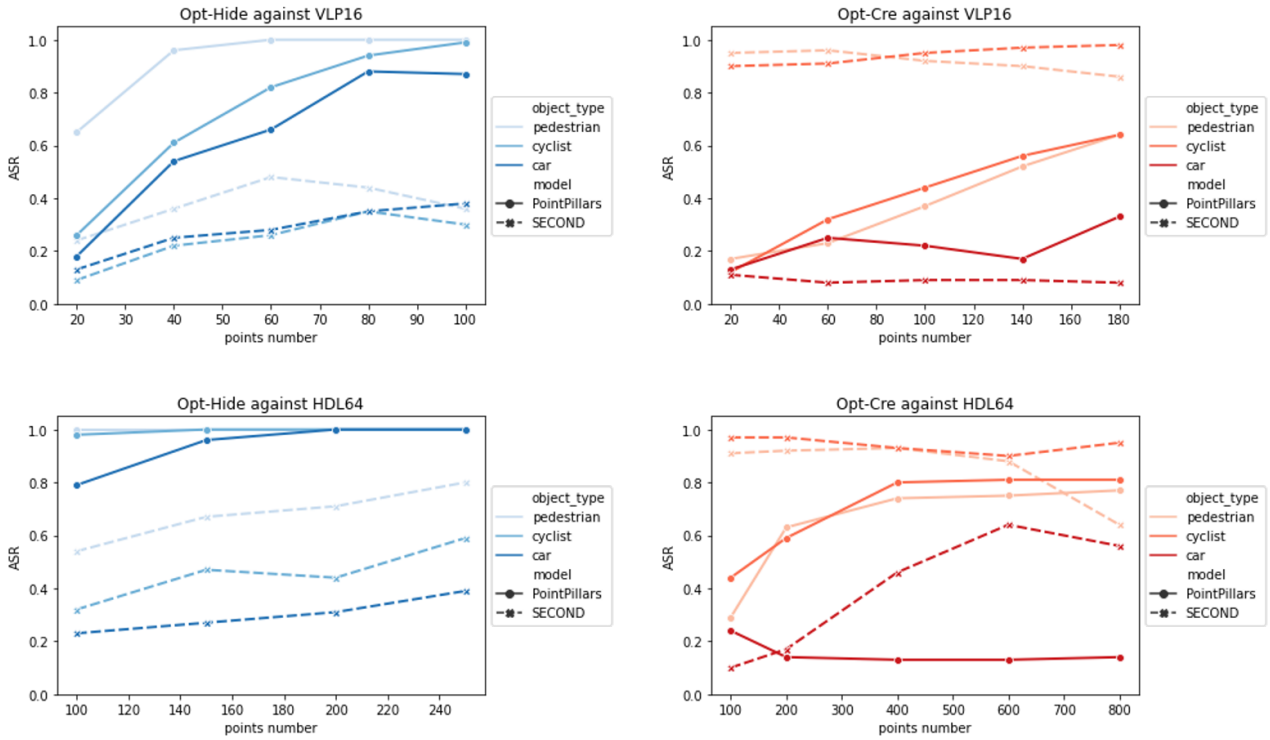
$$h_{receiver} = h_{lidar} + d * \tan(-1°)$$

Fig. 28: **A servo system to adjust the height of the receiver according to the height of the victim LiDAR and the distance between the receiver and LiDAR.**

pedestrians, cyclists, and cars can achieve 100%, 99%, and 87%. When against VLP-16 and SECOND, the ASRs for pedestrians, cyclists, and cars can achieve 36%, 30%, and 38%. When against HDL64E and PointPillars, the ASRs for pedestrians, cyclists, and cars can achieve 100%, 100%, and 100%. When against HDL64E and SECOND, the ASRs for pedestrians, cyclists, and cars can achieve 80%, 59%, and 39%. Generally speaking, the ASRs increase roughly along with the number of spoofing points increasing.

Optimization-based Creating Attacks. We select the position of the road in near front of (10 meters) the victim LiDAR as the center coordinates of our forged object, and the experimental results are shown in Fig. 29(b). When against VLP-16 and PointPillars, the ASRs for pedestrians, cyclists, and cars can achieve 56%, 59%, and 17%. When against VLP-16 and SECOND, the ASRs for pedestrians, cyclists, and cars can achieve 92%, 98%, and 17%. When against HDL64E and PointPillars, the ASRs for pedestrians, cyclists, and cars can achieve 77%, 81%, and 14%. When against HDL64E and SECOND, the ASRs for pedestrians, cyclists, and cars can achieve 64%, 95%, and 56%.

K. LiDAR Key Features Survey



(a) Opt-Hide Simulation. The ASR of hiding a front-near benign object with different number of spoofing points against VLP-16 and HDL64E on two 3D object detectors (PointPillars and SECOND).

(b) Opt-Create Simulation. The ASR of creating a front-near spoofing object with different number of points against two types of LiDAR (VLP-16 and HDL64E) on two 3D object detectors (PointPillars and SECOND).

Fig. 29: **The overall results of simulation attack.**

TABLE IV: **Survey of part of the LiDARs on the market.**

LiDAR	Company	Laser Number	Laser Vertical Distribution		Offical Scanning Sequence		Corrected Scanning Sequence	
			Vertical View	Vertical Resolution	single firing cycle	full cycle	single firing cycle	full cycle
VLP-16	Velodyne	16	-15°∼+15°	2°	2.304μs	55.296μs	2.304μs	55.296216μs
RS-16	Robosense	16	-15°∼+15°	2°	2.8μs	55.5μs	2.8μs	55.5522μs
HDL-32E	Velodyne	32	-30.67° ∼+10.67°	1.33°	1.152μs	46.080μs	\	\
Ultra Puck	Velodyne	32	-25°∼+15°	0.33°	2.304μs	55.296μs	\	\
Puck Hi-Res	Velodyne	16	-10°∼+10°	1.33°	2.304μs	55.296μs	\	\
RS-32	Robosense	32	-25°∼+15°	0.33°	1.44μs	55.5μs	\	\
HDL-64E	Velodyne	64	-24.9°∼+2°	0.4°	3.5μs, 1.2μs, 1.2μs, 1.3μs every 4 laser	57.6μs (dual return mode)	\	\
					2.4μs, 1.2μs, 1.2μs, 1.2μs every 4 laser	48μs (single return mode)	\	\