BadLiDet: A Simple Backdoor Attack against LiDAR Object Detection in Autonomous Driving

Shuai Li[1,2], Yu Wen[1✉], Huiying Wang[1,2], Xu Cheng[3]
[1] Institute of Information Engineering, Chinese Academy of Sciences, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, China
[3] School of Computer Science, Peking University, China
Email: {lishuai, wenyu, wanghuiying}@iie.ac.cn, chengxu@mprc.pku.edu.cn

***Abstract*—Autonomous vehicles (AVs) widely deploy LiDAR-based 3D object detection to accurately perceive and understand the surrounding environment. The mainstream LiDAR detection systems primarily adopt deep neural networks (DNNs) to achieve satisfactory performance. However, annotating large amounts of LiDAR data and training complex LiDAR detection models are time-consuming and resource-intensive. A common practice for some individual developers and self-driving companies is to outsource data annotation or model training tasks to third-party platforms, which could expose a natural attack injection point. In this paper, we propose BadLiDet, a simple yet effective backdoor attack against LiDAR object detection in autonomous driving. Specifically, we present a model-agnostic attack strategy that enables an attacker to inject a shape-independent backdoor into the victim model by poisoning its training data. Afterward, the attacker can choose an ordinary object in different shapes as the trigger and place it around a specific location to easily deceive the LiDAR detection system. Our extensive experiments on five representative models and a public dataset demonstrate that BadLiDet can achieve a high attack success rate while preserving the utility of victim models. We further show the effectiveness of BadLiDet on the continuous attack scenario collected from a high-fidelity simulator. Moreover, the end-to-end simulation evaluation on an open-source self-driving platform shows that BadLiDet can cause a 100% vehicle collision rate.**

***Index Terms*—backdoor attack, LiDAR object detection**

I. Introduction

Autonomous Driving (AD) technology has received significant attention in recent years. LiDAR has become the standard equipment of many high-autonomy AVs as it can not only provide high-accuracy 3D point clouds about the driving environment but also operate effectively in various lighting and weather conditions. As a core task in LiDAR-based perception systems, LiDAR object detection aims to locate and classify the obstacles on the road from input point clouds, enabling AVs to precisely understand the status of objects in 3D space and make safety-critical driving decisions for safe navigation. Therefore, correct detection of obstacles in the presence of deliberate adversaries is essential to ensure the safety and reliability of AD systems.

It is known that training reliable LiDAR object detectors requires massive well-annotated training data and computing resources, which is extremely labor-intensive and time-consuming, especially for collecting and annotating large-scale and diverse LiDAR point clouds. Therefore, it is common for individual AD system developers and self-driving companies to outsource the data annotation and/or model training tasks to third-party platforms (*e.g.,* Scale [4] and Apollo Fuel [3]). This practice inevitably provides ample opportunity for adversaries to compromise LiDAR perception systems. For instance, the malicious employees of data annotation platforms may poison the data used for training the LiDAR object detector to degrade its performance in the deployment stage.

In this paper, we focus on a prominent threat to DNN models called backdoor attacks, where the attacker can embed a hidden malicious function into the model by poisoning its training data. Specifically, we aim to make the *backdoored victim detector* trained from the poisoned data fail to recognize a chosen object of *target class* when a trigger presents in the driving scenario, and still behave normally in clean scenarios. Most existing backdoor attacks concentrate on the image recognition domain, and there is little research in the point cloud domain. The authors in [23], [14] customized backdoor triggers for point cloud data by inserting additional point clusters around target objects. A concurrent work [27] further extends this idea to attack LiDAR object detection. Although those methods achieve good attack performance, all of them adopt the *static* trigger pattern (*e.g.,* shape) to some extent and could suffer from attack failure when actually captured trigger points in the inference phase are inconsistent with the ones in the training phase, which is a common issue in image domain [16] and also exists in point cloud domain [27].

The above limitation naturally raises an important question: **Can we use a trigger object in arbitrary shapes to attack LiDAR object detection?** The shape-independent property brings two advantages: (1) attackers can flexibly choose different common objects as triggers according to their different needs. (2) the attack can remain effective under diverse driving conditions such as different distances and directions, where the pattern of the captured triggers is always changing. Although such an attack is promising, there are still some challenges to be solved. The main challenge is how to mitigate the reliance of the backdoored model on specific trigger shapes. Another challenge is how to ensure the generated poisoning data are effective for different types of object detectors. This issue should be considered since the existing detection models exhibit great diversity in their model designs and it is not always practical for attackers to have access to model information like [27].

To answer the above questions, we propose BadLiDet, a simple yet effective backdoor attack against LiDAR object detection. Specifically, we first design a new trigger pattern called birandom point clusters to represent possible physical objects, whose shape and number of points are both random during the generation process. In this way, we can make the victim detectors build strong correlations between target objects and the generic features (*e.g.,* location) of inserted triggers rather than a concrete trigger pattern (*e.g.,* shape and point's number). Then, based on an observation that modern object detectors usually have a large receptive field to guarantee good detection performance for different objects, we carefully choose a position from the upper area near target objects to place generated triggers. In this manner, we can easily establish a similar backdoor for different object detectors without any extra model knowledge.

To validate our attack, we conduct both digital-world and simulated physical-word evaluations. The results on a public dataset show that LiDAR object detection is highly vulnerable to our BadLiDet, involving five representative object detectors and three different trigger locations. The attack success rate is more than 95% while the detection performance loss of backdoored models is within 1.5% in most cases. The results on a high-fidelity simulator demonstrate that our BadLiDet is still highly effective and robust in various practical attack scenarios. We further carry out an end-to-end attack evaluation using the simulator and an open-source self-driving platform Baidu Apollo [2] to understand the practical safety impact of our attack on the AD system, and the results show that BadLiDet can cause a 100% collision rate.

In summary, we make the following key contributions:

- We propose BadLiDet, a simple and effective backdoor attack against LiDAR object detection with shape-independent and model-agnostic features.
- We perform comprehensive experiments to evaluate the performance of BadLiDet on five representative 3D object detectors and a public real-world dataset.
- We validate the effectiveness of BadLiDet under diverse attack scenarios and the safety impact on an industry-grade AD system using a high-fidelity simulator.

II. Background

A. LiDAR-based 3D Object Detection

The goal of LiDAR object detection systems is to accurately locate and classify potential obstacles in 3D space from the collected point clouds. Let $P = \{p_i\}_{i=1}^{m}$ be a frame of LiDAR point cloud, where each point p_i consists of the 3D coordinates (x_i, y_i, z_i) and the reflection intensity r_i. A 3D detector F_θ takes $\mathcal{P}$ as input and outputs a series of 3D bounding boxes to represent the detected objects. Each bounding box defines an object's center location, size, orientation, and confidence. To reduce false positives, some boxes with confidence scores below a threshold T_{conf} will be filtered out.

Recently, DNN-based LiDAR object detection systems have made remarkable progress. The existing detectors can be basically categorized into two groups: (1) one-stage methods that directly predict 3D bounding boxes from the extracted feature maps. (2) two-stage ones that first generate some coarse proposals and then perform fine-grained proposal refinement. Researchers also proposed different input representations (*e.g.,* voxel, pillar, and raw point) and feature extraction methods (*e.g.,* 3D sparse CNN, PointNet, and Transformer) to process LiDAR point clouds. According to whether predefined anchors are used during proposal generation, current detectors can be further classified into anchor-based and anchor-free methods. It is well-known that one-stage detectors are more computationally efficient and suitable for real-time applications like autonomous driving compared with two-stage counterparts. This paper mainly studies the backdoor robustness of five typical one-stage detectors.

B. Backdoor Attack

In a typical backdoor attack setting [9], the attacker aims to implant hidden behaviors called backdoors into a victim model by poisoning its training data, so that he can manipulate the victim model to behave unexpectedly when encountering specific conditions called triggers while performing normally on regular inputs. The core idea behind backdoor attacks is to induce the victim model to establish a strong connection between the trigger and the adversary-specified attack behavior. The backdoor adversary usually has access to a small set of samples that are used for training the victim model. He needs to embed the trigger into those samples and change the labels of trigger-embedded samples to reflect the victim model's behavior on poisoned samples. A trigger can be described as its pattern (*e.g.,* shape, color, and size) and its location in the sample. How to design a domain-specific trigger and attack behavior is essential to extend backdoor attacks to new data domains, model tasks, and attack scenarios.

III. Threat Model

Attack Surfaces. In this paper, we focus on data-poisoning based backdoor attacks which have expansive threat scenarios (*e.g.,* using third-party samples, annotation services, or training facilities). Those scenarios still exist during the development stage of LiDAR perception systems due to the limited data or computing resources of individual developers. For instance, a common practice for developers is to collect point cloud data from self-driving car users [27] or adopt third-party annotation services to annotate their data samples [12]. Therefore, a malicious data provider or annotation service vendor can easily poison the dataset and lead to a backdoor attack [10].

Attacker's Capabilities. We consider a backdoor adversary who has access to a small ratio of training data of the victim model and can manipulate the data and labels of those data samples. Accordingly, we assume that the attacker has no knowledge of victim models (*e.g.,* model architecture) and has no control over the training process. This is one of the most difficult settings for backdoor attacks [15].

Attacker's Objectives. The adversary's goal is to jeopardize the victim AV's safety by changing the perception results of the LiDAR detection system. Specifically, the attacker aims

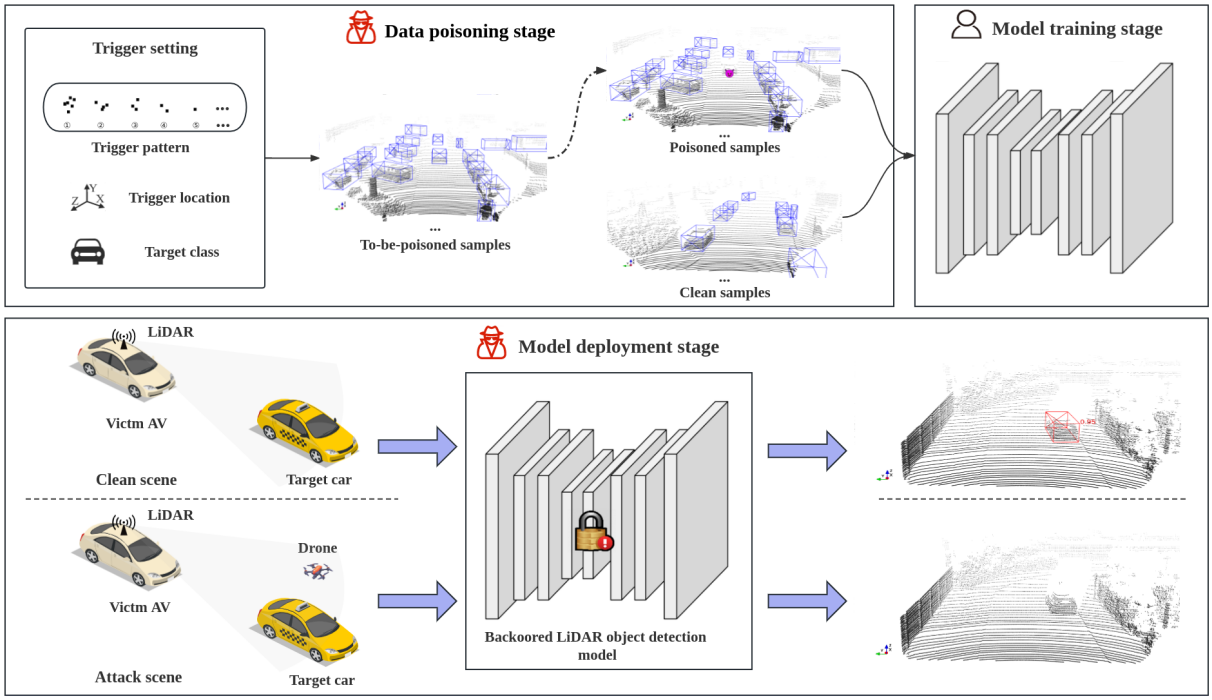


Fig. 1. The overview of our proposed BadLiDet.

to make victim AVs equipped with the backdoored object detector miss detection of the chosen target object when a trigger object is present in the scene, while maintaining normal detection ability in clean scenes to bypass potential model inspection and guarantee stealthiness. Here the trigger objects could be in any shape as long as they could reflect lasers.

In this paper, we mainly focus on hiding a target vehicle from the LiDAR detection systems. In particular, we consider a practical driving scenario where there is a car in front of the victim AV, and the attacker wants to remove the car from the detection results of the coming victim AV by placing a common object around the target. Such an attack may lead to a rear-end collision with serious consequences. The attack motivation could be malicious business competition.

IV. Methodology

A. Attack Overview

Figure 1 shows the overview of our proposed attack method. BadLiDet consists of three stages: the first stage of data poisoning and the last stage of backdoor activation are accessible to attackers, and the second stage of model training is controlled by the developer. The attacker first prepares basic attack settings, including the design of trigger patterns, the selection of trigger location, and the choice of attack target. Then, the attacker inserts the triggers into some clean samples and changes the labels of those samples. After the victim model is trained on the poisoned data and actually deployed on AVs, the attacker can easily fool the backdoored model by placing a common object with any shape (*e.g.,* a drone) around a target car. As long as these objects can reflect lasers, the backdoor behavior can be triggered.

B. Attack Design

Trigger Pattern Setting. In this paper, we represent a trigger using its pattern and its location in LiDAR point clouds. The trigger pattern further includes its shape, size, and number of points. We follow the basic trigger design of existing works that use an additional point cluster as the trigger since it can be easily performed using real-world physical objects. However, we found that existing triggers either use point clusters with a fixed shape and point number or adopt a pre-specified object to produce point clusters with a single pattern, which could limit their robustness and flexibility in real-world attack scenes. Considering the dynamic driving scenario, there exist many uncontrollable factors that affect the functionality of triggers. For instance, when the victim AV approaches the target car, both the shape and point number of the captured trigger points are always changing. The inconsistencies between the generated trigger point clusters and the ones used in the training phase may lead to attack failure.

To alleviate the above problems, we develop a shape-independent trigger pattern that enables the adversary not to care about the shape and point number of the captured trigger point cluster during the attack. Our idea is to introduce more randomness into the trigger point cluster to reduce model dependence on specific patterns. Specifically, we propose to use birandom point clusters (BPC) as our triggers to represent arbitrary objects. The shape and point number of those point clusters are both randomly generated during the whole process to simulate possible arbitrary trigger points captured by the

victim LiDAR. To ensure the feasibility and rationality of trigger objects in the physical world, we use two parameters *ms* and *mp* to limit the maximum size and maximum point number of a birandom point cluster P^t, which can be described as $|P^t| \leq mp$ and $\max_{p_a, p_b \in P^t} \|p_a - p_b\|_2 \leq ms$. In this manner, the victim model is expected to be robust against different trigger variations in terms of shape and point number.

Trigger Location Selection. In this paper, we represent the trigger location as the center of a trigger point cluster. To find a suitable location to place the trigger in a model-agnostic manner, we make an observation that LiDAR object detectors usually need to capture sufficient contextual information around an object to get better detection performance [7]. Thus we choose a position from the area near the target vehicle as our trigger location to generate unified poisoned samples for different detectors. In particular, we adopt the area above a target vehicle as the candidate trigger area due to two reasons. First, the trigger object is not easily occluded by other objects in this area. Second, this area allows a trigger (*e.g.,* drone) to continually follow and attack the target vehicle. We divide the trigger area into several sub-regions, including *top, left, right, front, and behind* regions. The attackers can flexibly choose a trigger location from those sub-regions according to their demands such as easier to align with the target car. Note that the trigger location is defined in the local coordinate of the target object to keep the consistent relative location.

Poisoned Sample Generation. Given the well-defined trigger and an attack target class, the attacker randomly selects a fraction of point cloud samples from the training dataset and inserts the trigger into those selected samples. There are usually many target objects in a driving scene, so we use a parameter *mo* to limit the maximum number of objects that can be poisoned. We emphasize that the injected point clusters are kept random (*i.e.,* re-generated with a random shape and random point number) for each to-be-poisoned target object. Here we define the poisoning ratio *pr* at the object level instead of the frame level, which is calculated as the percentage of poisoned objects among all objects in training data. Recall that the attacker aims to hide the target object when a trigger is present, so we need to delete the bounding box annotations of those trigger-embedded objects to establish the backdoor connection. The generated poisoned sample and the remaining clean samples constitute the poisoned training set.

Backdoor Behavior Activation. After the victim model successfully infects the backdoor and passes the model inspection, the attacker can attack any victim AVs equipped with the backdoored detector. The attacker can choose some common objects (*e.g.,* drone or road sign) as triggers without caring about their shapes. By placing the selected object around a specified trigger location of the target vehicle, the attacker can easily fool the LiDAR object detection system deployed by the victim AV. To make the attack more stealthy to humans, the adversary can even directly produce some trigger points via LiDAR spoofing technology [6].

V. Experiments in digital world

In this section, we conduct a series of experiments in digital world to answer the following research questions:

RQ1: Can our method be an effective backdoor attack against different object detectors?

RQ2: Can our method be robust against additional trigger variations beyond shape?

RQ3: What is the effect of different parameters in our method on the attack performance?

A. Experimental Settings

Dataset. We evaluate BadLiDet on the KITTI [8] dataset, which is a commonly-used benchmark dataset. The KITTI 3D object detection benchmark contains 7,481 training point clouds, which are further divided into *train* split (3,712 samples) and *val* split (3,769 samples) following the common practice. The *train* split includes 14,357 cars in total.

TABLE I
The details of victim 3D object detectors.

Detector	Representation	Backbone	Anchor	Stage
SECOND [24]	Voxel	3D Sparse CNN	✓	1
PointPillars [13]	Pillar	2D CNN	✓	1
CenterPoint [26]	Voxel	3D Sparse CNN	×	1
IA-SSD [28]	Point	PointNet++	×	1
VoxSeT [11]	Voxel	Transformer	✓	1

3D Object Detectors. We choose 5 representative LiDAR-based 3D object detectors, which comprehensively cover the mainstream model architectures. The details of those detectors are summarized in Table I. To ensure the fairness and consistency of the experiments, we use the open-source toolbox OpenPCDet [21] to train all detectors under the default model configurations with 4 NVIDIA V100 GPUs.

Evaluation Metrics. We use the mean Average Precision (mAP) to evaluate the detection performance (model utility), which ignores the grouping (*e.g.,* different difficulty levels) within the category. We adopt Attack Success Rate (ASR) and Attackable Objects Rate (AOR) to measure the attack performance. Specifically, we define them as follows:

- **mAP$_{\text{all}}$:** The mAP$_{all}$ is the average value of all categories on the clean validation set. It includes mAP$^c_{all}$ for the clean model and mAP$^p_{all}$ for the poisoned model.
- **mAP$_{\text{car}}$:** The mAP$_{car}$ is only defined in the car category. Here we use mAP$_{car}$ to reflect the attack impact on the selected target class. Similar with mAP$_{all}$, mAP$_{car}$ is also composed of mAP$^c_{car}$ and mAP$^p_{car}$.
- **ASR:** Given some trigger-embedded target objects, ASR is defined as the fraction of those objects that are not detected by the model. ASR also includes ASR^c for clean models and ASR^c for poisoned models.
- **AOR:** For each target object, we will generate some different random triggers for attack evaluation. We call a target object attackable if there exists at least one trigger that can successfully activate the backdoor. AOR is defined as the ratio of those attackable objects among all test objects, which also includes AOR^c and AOR^p.

TABLE II
THE COORDINATES OF THREE TRIGGER LOCATIONS.

Direction	X-axis	Y-axis	Z-axis
Top trigger	$0m$	$0m$	$0.7m$
Left trigger	$0m$	$(w/2+0.4)m$	$0.5m$
Behind trigger	$-(l/2+0.5)m$	$0m$	$0.6m$

TABLE III
THE MAP OF CLEAN AND POISONED MODELS ON THE KITTI DATASET.

Detector	Clean		Top trigger		Left trigger		Behind trigger	
	mAP_{car}	mAP_{all}	mAP_{car}	mAP_{all}	mAP_{car}	mAP_{all}	mAP_{car}	mAP_{all}
SECOND	81.49	68.07	81.16	68.23	80.79	67.67	79.94	67.24
PointPillars	79.91	65.97	79.57	65.75	79.41	65.49	78.78	65.42
CenterPoint	80.80	66.62	80.59	66.74	80.27	66.11	79.31	65.69
IA-SSD	82.44	71.21	82.03	70.42	81.99	70.09	82.26	71.69
VoxSeT	81.74	69.67	81.40	69.65	80.94	68.88	80.80	68.49

Attack Settings. BadLiDet has the following parameters: maximum trigger size ms, maximum trigger point number mp, maximum poisoned object number in a scene mo, poisoning rate pr and trigger location. We empirically set mp based on ms: $mp = (10 * ms)^2$. We instantiate three trigger locations from three different sub-regions to evaluate our BadLiDet. Unless otherwise mentioned, we use the following default parameters. We set $ms = 0.3m$, $mp = 9$, $mo = 1$ and $pr = 5\%$. Table II shows the detailed coordinate settings of three trigger locations, where the origin is the roof center of the target car, l and w denote the length and width of the target. In addition, the 3D detector relies on confidence threshold T_{conf} and 3D IoU threshold T_{iou} to judge whether a labeled object is detected. We set $T_{conf} = 0.5$ for all detectors when evaluating the attack performance of BadLiDet, which is also consistent with some open-source AD platforms [1], [2]. T_{iou} is set to 0.1 to ensure the attack does cause a real disappearance instead of a slight box offset when calculating ASR and AOR.

For each considered 3D object detector, we repeat the poisoning experiment multiple times and report the average results. We use the full clean validation data to evaluate the model utility and randomly select about 2,000 target cars from the validation set to evaluate the attack performance. To ensure the reliability of the results, we only choose those objects that can be detected by both clean and corresponding poisoned detectors. For each target car, we generate a birandom point cluster at the trigger location to perform the attack and the generation process is repeated 100 times. The average ASRs and AORs are reported to reduce the randomness.

B. Overall Attack Performance (RQ1)

BadLiDet maintains model utility. Table III compares both mAP_{car} and mAP_{all} of five object detectors under clean and three different backdoor attack settings. We can observe that the largest drop of mAP_{all} and mAP_{car} in all combinations of detector and trigger location are 1.18% and 1.55% respectively. In most cases, the poisoned detector has a similar mAP compared with the corresponding clean detector. Those results indicate that our BadLiDet can maintain the utility of the 3D object detection model.

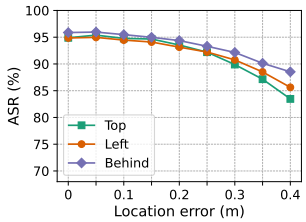


Fig. 2. Robustness to location errors.

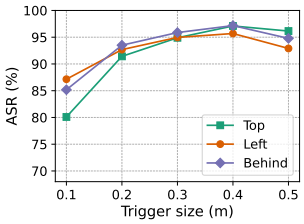


Fig. 3. Robustness to trigger size.

BadLiDet achieves high ASR. Table IV reports both ASR and AOR of five object detectors under clean and three different backdoor attack settings. We can see that our attack can achieve over 94% ASR and over 99% AOR in all trigger locations for all poisoned detectors while the ASR and AOR are close to 0% for all clean detectors. The results suggest that our attack is highly effective for different object detectors and the backdoor behavior is injected into the model via BadLiDet and it is not an existing vulnerability of the detector.

C. Attack Robustness Analysis (RQ2)

When the attacker leverages a trigger object to perform attacks in the physical world, there exist many uncontrollable factors that affect the attack's effectiveness. We identify two additional variations which include location error and trigger size to evaluate the robustness of BadLiDet. Here we choose PointPillars as the victim model to analyze the results.

Impact of location error. It is challenging for an attacker to precisely place the adopted object at the desired trigger location in the physical world. We simulate possible location errors by shifting the birandom point cluster by different distances toward random directions. Figure 2 shows the ASR when we vary the shifting distance from $0.05m$ to $0.4m$. Overall, the ASR of BadLiDet decreases monotonically when the shifting distance increases. We can also observe that BadLiDet keeps satisfactory attack performance in all trigger locations under different location errors. Even when the shifting distance reaches $0.4m$, the ASR is still over 83% in all cases. The above results demonstrate that BadLiDet is robust to different location errors. Thus the attacker does not have to place the trigger object at a specific location with high precision to activate the backdoor, which greatly reduces the difficulty of attack deployment in the physical world.

Impact of trigger size. It is possible for an attacker to adopt a trigger object whose size is not consistent with the one used in the training phase. We simulate different object sizes by varying the *ms* of the birandom point cluster from $0.1m$ to $0.5m$. Figure 3 shows the ASR when generating birandom point clusters with different sizes to attack the victim detector. Overall, the ASR of BadLiDet decreases as the trigger size decreases when the size is less than $0.3m$, which indicates that triggers with fewer points will have weaker backdoor activation ability than triggers with more points when both of them use almost the same number of poisoned samples during backdoor learning. Even when the object size is $0.1m$, the

TABLE IV
THE ASR AND AOR OF POISONED AND CLEAN MODELS ON THE KITTI DATASET.

Detector	Top trigger				Left trigger				Behind trigger			
	ASR^p	ASR^c	AOR^p	AOR^c	ASR^p	ASR^c	AOR^p	AOR^c	ASR^p	ASR^c	AOR^p	AOR^c
SECOND	97.04	0	99.34	0	97.39	0	99.69	0	98.10	0	99.54	0
PointPillars	95.21	0.14	99.28	0.15	94.96	0.07	99.54	0.15	95.87	0.12	99.49	0.15
CenterPoint	97.24	0	99.34	0	95.72	0	99.54	0	94.11	0	99.29	0
IA-SSD	96.91	0	99.85	0	94.77	0	99.95	0	97.30	0	99.90	0
VoxSeT	98.58	0.02	99.95	0.05	97.78	0.03	99.86	0.15	97.81	0	99.79	0

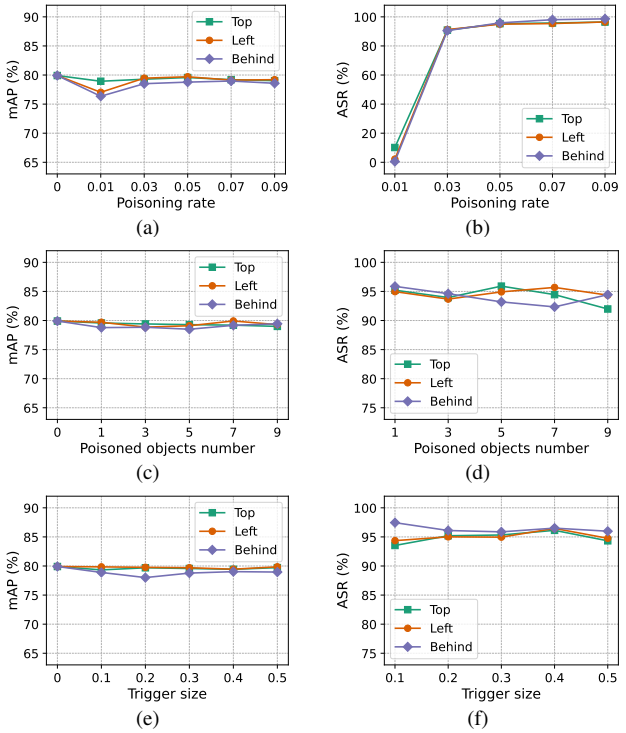


Fig. 4. The impact of different parameters on model utility and attack effectiveness of BadLiDet in KITTI dataset.

ASR is still over 80% in all cases[1]. In addition, a larger trigger size can not always bring higher attack effectiveness. When the size increases to 0.5m, the ASR is even slightly lower than the baseline in some cases. A possible reason is that a few point clusters generated by a large trigger are equivalent to a slight location error of the ones generated by a small trigger. The above results indicate that BadLiDet is robust to the different trigger sizes within a reasonable range, so the attacker also does not need to care about the size of the adopted object.

D. Ablation Study (RQ3)

In this set of experiments, we explore the effect of different parameter values in BadLiDet. We focus on the following three important parameters: poisoning rate, the number of poisoned objects in a scene, and trigger size. We choose PointPillars as victim detector and mainly report mAP_{car} and ASR of poisoned models.

[1]When the number of points is 2 for a trigger with size 0.1m, the attack success rate is over 91%.

Impact of poisoning rate. Figure 4(a) and 4(b) shows the experimental results when we vary pr from 1% to 9% with a step size of 2%. We observe that the ASR first increases and then saturates as the pr increases in all trigger locations. For PointPillars, 3% of total target objects are needed to achieve good attack performance. In addition, once the breakthrough point is reached, the growth trend behind it is not very significant, which implies the attacker does not rely on a higher pr to achieve good performance. As for the model utility, we can see that the mAP_{car} does not present a continuous downward trend with increasing pr, which means a higher pr does not necessarily lead to a lower mAP_{car}. Furthermore, the drop of mAP_{car} of the poisoned model is negligible compared with the baseline when the pr is between 3% and 9%.

Impact of poisoned objects number in a scene. Figure 4(c) and 4(d) show the experimental results when we vary mo from 1 to 9 with a step size of 2. We can see that BadLiDet achieves similar ASR under different mo values. At the same time the mAP_{car} of poisoned models are comparable to the clean baseline in all cases. The above results suggest that BadLiDet is insensitive to the mo parameter. The attacker can choose a small mo value (*e.g.,* 1) to reduce the risk of attack exposure in practice if he has access to more samples to poison.

Impact of trigger size. Figure 4(e) and 4(f) shows the experimental results when we vary ms from $0.1m$ to $0.5m$ with a step size of $0.1m$. We can observe that the ASR does not change much under different ms values. At the same time the mAP_{car} of poisoned detectors are also similar to the clean baseline in most cases. Those results indicate that the performance of BadLiDet is agnostic to this parameter. For real-world attack deployment, the attacker can choose a small ms value to keep stealthiness.

VI. EXPERIMENTS IN SIMULATED PHYSICAL WORLD

In this part, we perform extensive simulation experiments to answer the following research questions:

RQ4: Can our attack still be effective in a simulator where we simulate the real-world attack process?
RQ5: Can our attack still be robust against different trigger variations and driving conditions?
RQ6: How effective is BadLiDet compared with baselines?
RQ7: Can our attack really cause a rear-end collision to the full-stack AD system?

A. Experimental Settings

In previous experiments, we assume an ideal attack setting where an attacker could directly manipulate and pass the LiDAR point clouds to a backdoored detector. In addition, we only consider discrete LiDAR frames for attack evaluation. To better reflect the practical attack effectiveness of BadLiDet, we use a high-fidelity simulator to simulate real-world attack processes, which include collecting the consecutive point cloud sequences using a LiDAR sensor and performing attacks using a physical object. We choose LGSVL simulator [17] to create various driving scenarios. LGSVL is a popular autonomous driving simulator used to facilitate the development and testing of AD systems, which is also adopted by many researchers for attack verification [6], [25].

Simulator Settings. We use an open-source map CubeTown as the driving environment. A victim AV equipped with 64-line LiDAR is used for data collection. The height of LiDAR from the ground surface is set to be 1.8m. We utilize the tools from Baidu Apollo to parse the LiDAR point clouds and Python API from LGSVL to control the victim AV.

Attack Settings. We adopt a 3D object as the trigger and place it around the chosen location to perform an attack. By default, we set the shape of the trigger as a circle to simulate possible traffic signs in the real world. The size of the trigger is limited to $0.3m$, which is consistent with the one used in the training phase. Here we use a pole to hold the trigger object. We take the *left* trigger location as an example to perform the attack and report the experimental results. We consider an attack scenario where the victim AV is directly behind the target car at the beginning. Figure 5(a) and 5(c) show the original benign scene and corresponding attack scene. For each scenario simulation, we control the victim AV to approach the target car from a distance of $30m$ to $5m$ at the speed of $2m/s$ and collect about 120 continuous LiDAR frames. Here we define the ASR as the percentage of the frames that miss detection of the target object in all consecutive frames. We choose PointPillars as the victim detector due to its popularity.

Compared Baselines. To demonstrate the advantage of the trigger design in our BadLiDet, we compare it with the following three trigger patterns. Note that they are originally designed for the 3D point classifiers, we generalize those methods to our BadLiDet for a fair comparison. In particular, we replace the trigger pattern of BadLiDet with compared ones and keep other settings unchanged.

- **RS [23], [14]:** RS is generated by randomly sampling a fixed number of points distributed on a sphere. We set the diameter of the sphere as $0.3m$ and the number of points in each point cluster as 32.
- **RP [23]:** RP is generated by randomly sampling a fixed number of points distributed in a ball. We set the diameter of the ball as $0.3m$ and the point's number as 32.
- **HS [23]:** HS is generated by randomly sampling some points distributed on a half sphere with random orientation, which is derived from RS by keeping points having a positive inner product with a random vector.

B. Attack Performance (RQ4)

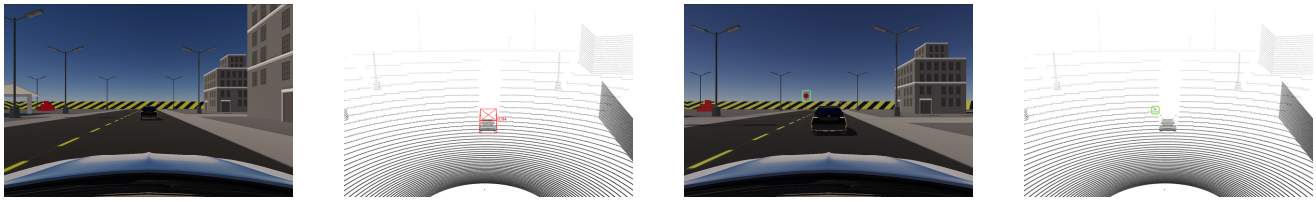
As shown in Figure 5(b), the target car can be correctly detected with high confidence by the victim AV in a benign environment. Figure 5(d) show the detection result after we place a trigger object at the specific location around the target car. We can see that the trigger successfully creates a small point cluster around the chosen location and the target car is not detected by the backdoored detector. We totally collect 123 LiDAR frames for this attack scenario and the attack success rate is 97.56%. For the benign scenario, we totally collect 126 LiDAR frames and the backdoored model can always detect the target car. The above results demonstrate that BadLiDet can not only achieve high attack effectiveness but also maintain normal detection performance under a more realistic experimental setting.

C. Attack Robustness Evaluation (RQ5)

Effect of location error. When performing attacks in the real world, it is difficult for an attacker to place the trigger object perfectly at the desired location. To measure the robustness of BadLiDet to the location error of a trigger object, we move the trigger in Figure 5(c) towards random directions with $0.3m$ for several times. The average ASR is 92.91% on all collected 621 LiDAR frames. Figure 6(a) and 6(e) show the new location of the trigger object after we shift it along the left side with $0.3m$ and the corresponding detection result. We can see that the target car is still not detected by the victim AV. The above results imply that BadLiDet is robust to a moderate location error of trigger objects, which facilitates the attack deployment in the physical world.

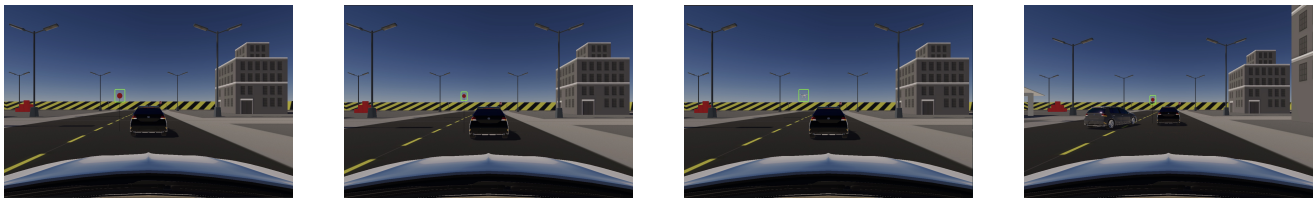
Effect of trigger size. In the practical attack deployment, an attacker may adopt a trigger object whose size is different from the one used in the training phase. To evaluate the robustness of BadLiDet to the size of a trigger object, we replace the trigger in Figure 5(c) with other two circle objects whose sizes are $0.2m$ and $0.4m$. We collect 251 and 258 LiDAR frames for the two triggers. The corresponding average ASRs are 94.02% and 97.67%, respectively. The new trigger object with a size of $0.2m$ is shown in Figure 6(b) and the corresponding detection result is shown in Figure 6(f). We can observe that our attack is still effective although we adopt a smaller trigger. The above results imply that BadLiDet is robust to a small size variation of trigger objects, which increases the diversity of trigger selection.

Effect of trigger shape. The design of BadLiDet makes the attacker not need to consider the shape of a trigger object when performing attacks in practice. To demonstrate this point, we replace the trigger in Figure 5(c) with three other objects of different shapes (*i.e.,* triangle, rectangle, and a drone). We totally collect about 250 LiDAR frames for each of the three triggers. The corresponding ASRs are 96.92%, 96.15%, and 95.87%, respectively. An example attack scenario where we employ a drone as the trigger is shown in Figure 6(c). The detection result in Figure 6(g) shows that the target is invisible to victim AV. The above results imply that BadLiDet is robust



(a) Original scene (b) Original detection result (c) Adding a trigger object (d) Detection after attack

Fig. 5. Attack using a trigger object in the simulator. The red bounding boxes represent the detected objects. The trigger is highlighted with green rectangles.



(a) Trigger at a different location (b) Trigger with a different size (c) Trigger with a different shape (d) Passing-by vehicle

(e) Detection result (f) Detection result (g) Detection result (h) Detection result

Fig. 6. Attack robustness evaluation in the simulated physical world.

to the shape of trigger objects, which significantly increases the flexibility of trigger selection.

Effect of driving direction. The driving direction of the victim AV is usually unpredictable to an attacker during the practical attack process. To study the robustness of BadLiDet to this uncertainty, we also control the victim AV in Figure 5(c) to approach the target car from different driving directions (*i.e.,* left side and right side of the target car). We collect 599 LiDAR frames in total and the average ASR is 95.16%. The results imply that BadLiDet is robust to different driving directions of victim AV.

Effect of victim vehicle's speed. In practical driving scenarios, the victim AV may approach the target car at different speeds. To illustrate the robustness of BadLiDet to this factor, we control the victim AV with different speeds (*i.e.,* $5m/s$ and $10m/s$). We totally collect 142 and 68 LiDAR frames for both attack scenarios. The average ASR is 98.59% and 98.53% respectively. The results imply that BadLiDet is robust to different driving speeds of victim AV.

Effect of passing-by vehicle. In real-world driving environments, there are usually many other obstacles around the target car. To explore whether the trigger functionality would be affected by the surrounding environment, we consider a scenario in Figure 6(d) where another vehicle is passing by the target car. We collect 130 LiDAR frames in total. The ASR for the target car is 97.69%. Additionally, the backdoored detector can always detect the passing-by vehicle with a 100% detection rate. The visualization result is shown in Figure 6(h). The results imply that BadLiDet is robust to small environment changes around the target object.

TABLE V
COMPARING BADLIDET WITH BASELINES.

Driving direction	Trigger shape	Left trigger			
		BPC	RS	RP	HS
Behind	Circle	97.46	13.93	34.62	73.75
	Triangle	97.92	13.08	30.76	47.74
	Rectangle	96.67	13.85	26.92	54.62
	Drone	95.83	11.87	22.61	57.39

Effect of target vehicle's motion state. In the end, we consider the most challenging attack scenario where the target car is moving during the attack and the trigger object should follow it to maintain the attack effect. Here we adopt a drone as the trigger due to its high mobility. The drone is set to move with the target at a similar speed. We confirm that the drone always has location errors relative to the perfect trigger position during the attack process. We totally collect 248 LiDAR frames and the ASR is 91.53%. The results imply that BadLiDet is still effective even when the target is moving, which further increases the attack practicality.

D. Comparison with Baselines (RQ6)

In this experiment, we perform attacks in the simulator using four trigger objects with different shapes. As shown in Table V, our BadLiDet can achieve very high ASR (*i.e.,* over 95%) and similar attack performance under different trigger shapes, while other compared baselines only obtain quite low ASR (*i.e.,* lower 50%) in most cases. The poor performance on baselines reveals that the fixed trigger pattern cannot adapt to dynamic changes in the driving environment.

E. End-to-End Attack Evaluation (RQ7)

Previous evaluation only considers the attack effectiveness at the model level (*i.e.,* object detection). However, the AD system contains various modules to make driving decisions jointly. In this experiment, we investigate the impact of our attack at the system level and assess whether BadLiDet can lead to a rear-end collision to full-stack AD systems.

Specifically, we conduct an end-to-end attack evaluation on Baidu Apollo and LGSVL simulator where the victim AV is controlled by the AD system. We enable all necessary modules, including localization, transform, perception, planning, prediction, routing, and control. We consider a scenario in Figure 5(a) where a victim AV approaches the target car from behind direction. We define an attack as successful if the victim AV crashes into the target car. Here we chose CenterPoint as the victim detector because the latest Baidu Apollo platform supports the deployment of CenterPoint trained from Paddle3D [5]. We train one clean model and three poisoned models with *top*, *left*, and *behind* triggers on the KITTI dataset. We run the simulation 50 times for each model and place a matching trigger object for three backdoored detectors.

The experimental results show that the Baidu Apollo, equipped with a clean detector, can always detect the target car and make the correct braking decision to avoid a collision. However, the AD system with those poisoned detectors always miss detection of the target car and cause a 100% collision rate in all considered trigger locations and different trigger shapes[2]. Those results demonstrate that our BadLiDet could actually lead to serious traffic accidents to AVs in practice attack deployment.

VII. Discussion

A. Possible Defense Strategies

Sensor Fusion. A natural idea is to exploit multi-sensor fusion to obtain more robust perception results, where the complementary sensors (*e.g.,* camera and radar) can provide additional evidence about the existence of an object. However, the camera or radar based perception systems are also vulnerable to malicious attack [29], [20]. The attacker could simultaneously attack all perception systems through different attack vectors to achieve the same attack objective. Moreover, how to effectively aggregate the inconsistent detection results from different sensors is still an open question.

Method Fusion. Although DNN-driven LiDAR perception models usually get more accurate detection results, we believe the clustering-based traditional method is naturally robust to object hidden attacks because it can physically determine the presence of an object from LiDAR point clouds. A good choice is to adopt both two methods to boost the detection recall. However, physical-based methods usually suffer from many false positives and may be susceptible to object creation attacks (*e.g.,* LiDAR spoofing). How to better fuse the advantages of the two methods and balance precision and recall of the perception system are still open questions.

[2]Attack demos are available at https://sites.google.com/view/badlidet.

B. Limitations and Future Work

Lack of Real-world Evaluation. Due to the restricted resources, we cannot collect LiDAR frames from real-world attack scenarios. However, the attack vector adopted by BadLiDet has been shown to be feasible in the real world [30]. In the future, we will validate the effectiveness of our attack with common objects and even LiDAR spoofing devices in the physical world.

Dynamic Backdoor Attack. Although BadLiDet shows good robustness to different location errors, it still belongs to the *static* backdoor. A more attractive alternative is *dynamic* backdoor [18], where the adversary could place the trigger at random locations to perform the attack. In our future work, we will explore the dynamic location based backdoor attack against LiDAR perception systems.

VIII. Related Work

Adversarial Attacks on LiDAR Perception Systems. The existing adversarial attacks explore the vulnerabilities of LiDAR perception tasks deployed in AD systems to achieve the specific attack expectations (*e.g.,* object hiding or object creating). Those attacks can be divided into two classes according to the means of launching an attack: laser-based ones [6], [19] and object-based ones [22], [30]. The signal-based methods strategically inject fake signals to victim sensor through extra attack devices. However, it is quite difficult to dynamically aim at the victim sensor with high precision and consistently generate some points with specific pattern in complex driving environments. The object-based methods generate adversarial points by placing 3D objects at some locations in the physical world. However, they either require placing a 3D-printed object with a specific shape in one location, or making some common objects appear around multiple locations (*e.g.,* 6). It is quite challenging to accurately build such objects of a specific shape or simultaneously control multiple ordinary objects. In addition, uncommon shapes or multiple objects appearing simultaneously are susceptible to human suspicion in practice. Moreover, the generated adversarial examples are usually *sample-specific* or *scene-specific*, inevitably weakening their attack impact. Different from all the above studies, we study the vulnerability of LiDAR perception through the lens of backdoor attacks. Our attack can be easily performed by placing a common object in arbitrary shape around the target object. It can also bring more broader security implications by enabling the adversary to attack different target objects and any AVs equipped with the backdoored detector.

Backdoor Attacks on 3D Point Clouds. There are only a few works studying backdoor attacks on 3D point cloud domain [14], [23]. Specifically, Xiang *et al.* [23] proposed to insert some triggers with optimal spatial location and local geometry into the training data to implant a backdoor. Li *et al.* [14] also investigated the effectiveness of rotation-based triggers. However, all of them focus on point cloud classification tasks whose goal is classification of a single object, whereas 3D object detection needs to recognize multiple

objects in a scene and output a series of 3D bounding boxes to represent them. In addition, they mainly target the synthetic point clouds derived from CAD models whereas we use the raw LiDAR point clouds collected from the real world.

We notice a concurrent work [27] which has the same research topic with us. We highlight the following differences with it. First, BadLiDet only requires access to a small ratio of training data while they need additional model information. Second, they suffer from a large performance degradation when performing attacks using trigger objects with different shapes while our BadLiDet is designed to resist this issue. Third, they do not achieve consistent attack performance for different object detectors as our BadLiDet.

IX. Conclusion

In this paper, we present a simple yet backdoor attack against LiDAR object detection in autonomous driving. In particular, we design a shape-independent and model-agnostic backdoor trigger to resolve the unique challenges of practical attack deployment in terms of flexibility and robustness. We demonstrate the effectiveness of our attack through extensive experiments in both the digital world and the simulated physical world. The end-to-end simulation experiments on a popular AD system show that our attack can cause a 100% vehicle collision rate.

References

[1] Autoware.AI. https://www.autoware.ai/.

[2] Baidu Apollo. http://apollo.auto.

[3] Open Lidar Model Training Service. https://github.com/ApolloAuto/apollo/tree/r7.0.0/docs/Apollo_Fuel/Perception_Lidar_Model_Training.

[4] Scale AI. https://scale.com/.

[5] Paddle3D. https://github.com/PaddlePaddle/Paddle3D, 2022.

[6] 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.

[7] Lue Fan, Ziqi Pang, Tianyuan Zhang, Yu-Xiong Wang, Hang Zhao, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Embracing single stride 3d object detector with sparse transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8458–8468, 2022.

[8] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[9] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

[10] Xingshuo Han, Guowen Xu, Yuan Zhou, Xuehuan Yang, Jiwei Li, and Tianwei Zhang. Physical backdoor attacks to lane detection systems in autonomous driving. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2957–2968, 2022.

[11] Chenhang He, Ruihuang Li, Shuai Li, and Lei Zhang. Voxel set transformer: A set-to-set approach to 3d object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8417–8427, 2022.

[12] M14 Intelligence. Autonomous vehicle data annotation market analysis. https://www.researchandmarkets.com/reports/4985697/autonomous-vehicle-data-annotation-market-analysis, 2020.

[13] 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.

[14] Xinke Li, Zhirui Chen, Yue Zhao, Zekun Tong, Yabang Zhao, Andrew Lim, and Joey Tianyi Zhou. Pointba: Towards backdoor attacks in 3d point cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16492–16501, 2021.

[15] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[16] Yiming Li, Tongqing Zhai, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shutao Xia. Rethinking the trigger of backdoor attack. *arXiv preprint arXiv:2004.04692*, 2020.

[17] Guodong Rong, Byung Hyun Shin, Hadi Tabatabaee, Qiang Lu, Steve Lemke, Mārtiņš Možeiko, Eric Boise, Geehoon Uhm, Mark Gerow, Shalin Mehta, et al. Lgsvl simulator: A high fidelity simulator for autonomous driving. In *2020 IEEE 23rd International conference on intelligent transportation systems (ITSC)*, pages 1–6. IEEE, 2020.

[18] Ahmed Salem, Rui Wen, Michael Backes, Shiqing Ma, and Yang Zhang. Dynamic backdoor attacks against machine learning models. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pages 703–718. IEEE, 2022.

[19] 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.

[20] Zhi Sun, Sarankumar Balakrishnan, Lu Su, Arupjyoti Bhuyan, Pu Wang, and Chunming Qiao. Who is in control? practical physical layer attack and defense for mmwave-based sensing in autonomous vehicles. *IEEE Transactions on Information Forensics and Security*, 16:3199–3214, 2021.

[21] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2020.

[22] 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.

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

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

[25] Kaichen Yang, Tzungyu Tsai, Honggang Yu, Max Panoff, Tsung-Yi Ho, and Yier Jin. Robust roadside physical adversarial attack against deep learning in lidar perception modules. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, pages 349–362, 2021.

[26] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021.

[27] Yan Zhang, Yi Zhu, Zihao Liu, Chenglin Miao, Foad Hajiaghajani, Lu Su, and Chunming Qiao. Towards backdoor attacks against lidar object detection in autonomous driving. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, pages 533–547, 2022.

[28] Yifan Zhang, Qingyong Hu, Guoquan Xu, Yanxin Ma, Jianwei Wan, and Yulan Guo. Not all points are equal: Learning highly efficient point-based detectors for 3d lidar point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18953–18962, 2022.

[29] Yue Zhao, Hong Zhu, Ruigang Liang, Qintao Shen, Shengzhi Zhang, and Kai Chen. Seeing isn't believing: Towards more robust adversarial attack against real world object detectors. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 1989–2004, 2019.

[30] 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.