# Physical Adversarial Attack Against LiDAR-Based 3D Perception via 2D Patches

Yubo Zhu, *Graduate Student Member, IEEE*, Zijian Wang, *Student Member, IEEE*, Wei Tong, *Member, IEEE*, Xinyue Zhao, Ruizhe Song, Yunlong Mao, *Member, IEEE*, and Sheng Zhong, *Fellow, IEEE*

***Abstract*—Point cloud data, renowned for its capability in capturing the shapes and spatial structures of objects, has grown increasingly vital in domains such as scene reconstruction and virtual/augmented reality. Nevertheless, deep learning models for point clouds have been shown vulnerable to adversarial attacks. Existing physical attacks against LiDAR-based perception typically rely on specialized laser-emitting devices, custom-fabricated 3D objects, or predefined roadway-shadow templates, which limits their practicality or generality. In this paper, we present a model-guided digital-to-physical attack that fools LiDAR-based 3D perception with 2D specular reflective patches on existing objects. We exploit the weakness of LiDAR in handling specular surfaces, identify task-critical points, group them into a small number of connected regions on the LiDAR projection plane, and map these regions to patch locations on the object surface. We first validate the attack in the digital domain and then realize the patches in the physical world with low-cost reflective materials. Extensive experiments on benchmark datasets and real-world objects show substantial accuracy degradation across ten state-of-the-art models and strong transferability in black-box settings. The same connected return-suppression strategy also reduces Moderate 3D AP for cars, pedestrians, and cyclists on the KITTI detection benchmark. These results expose an effective, low-cost attack surface for LiDAR-based systems.**

***Index Terms*—Point cloud, physical adversarial attack, LiDAR.**

## I. Introduction

LIGHT Detection and Ranging (LiDAR) based perception is a core system of various applications, particularly in the fields of scene reconstruction and Mixed Reality (MR), helping these applications to capture depth information of surrounding objects in the form of 3D point clouds [13], [27], [42], [50]. Deep learning has exhibited considerable potential in successfully addressing various computer vision tasks. To enhance the comprehension of point clouds, multiple deep neural network (DNN) models are employed for object recognition tasks based on 3D point cloud data, successfully achieving high accuracy [6], [25], [29], [31], [43].

Received 15 February 2026; revised 2 September 2026 and 4 October 2026; accepted 19 October 2026. Date of publication 5 November 2026; date of current version 12 November 2026. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62372226, Grant 62272215, and Grant 62672235. The associate editor coordinating the review of this article and approving it for publication was Prof. Yanjiao Chen. *(Corresponding author: Wei Tong.)*

Yubo Zhu, Zijian Wang, and Xinyue Zhao are with the State Key Laboratory for Novel Software Technology and the School of Computer Science, Nanjing University, Nanjing 210023, China (e-mail: zyb@smail.nju.edu.cn; zj-wang@smail.nju.edu.cn; zhaoxinyue@smail.nju.edu.cn).

Wei Tong, Ruizhe Song, Yunlong Mao, and Sheng Zhong are with the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China (e-mail: weitong@outlook.com; Stream-Song82@gmail.com; maoyl@nju.edu.cn; zhongsheng@nju.edu.cn).

Digital Object Identifier 10.1109/TIFS.2026.3750721

Recent research has revealed that adversarial attacks can significantly reduce the accuracy of DNN-based object recognition models designed for 3D point cloud data, both in the digital world [4], [5], [11], [15], [38], [44], [49] and the physical world [1], [3], [7], [14], [16], [18], [21], [39], [41], [53]. While adversarial attacks in the digital world allow attackers to directly alter point clouds, such manipulations are often difficult to realize in real-world settings. Physical adversarial attacks are more challenging because they require transforming physical objects or the environment to manipulate the point clouds captured by LiDAR sensors and thereby deceive the perception system. Existing physical attacks against LiDAR-based perception are typically *active*, injecting spoofed laser pulses with specialized emitters [1], [14], or *passive*, reshaping returns without emitting signals. The latter includes custom-fabricated 3D adversarial objects [3], [7], [18], [39] and reflective roadway shadows that create a nonexistent vehicle [16]. Active devices and 3D fabrication incur high deployment cost. Shadow-template attacks reduce this cost, but the perturbation is built from a fixed geometric template and scene-specific priors, and the resulting effect can be tied to particular model representations. How to locate physically connected, task-critical surface regions and realize them with a small number of low-cost reflective patches remains underexplored.

To address these issues in physical adversarial attacks, it seems promising to investigate the feasibility of attacking 3D LiDAR-based perception by solely modifying the surfaces of objects. The main idea is rooted in the inherent vulnerability of LiDAR, where specular reflection surfaces typically reflect very few laser pulses directly back to the LiDAR sensors. Consequently, this can lead to inaccurate mapping of physical objects to point clouds [14], [16], [22], [51]. By manipulating the reflected laser pulses to the LiDAR sensors through modifications of the target object's surface using specular reflective materials, we can potentially utilize 2D patches to fool the 3D perception of LiDAR sensors. Furthermore, employing multiple patches in different areas of an object's surface may result in a more potent and stealthy attack strategy.

In this work, we introduce a digital-to-physical attack pipeline against LiDAR-based perception. Our goal is a low-cost physical attack that can be realized on an existing object with a small number of reflective patches. Because a 2D patch suppresses returns along the LiDAR line of sight rather than isolated 3D points, unconstrained point removal is not physically realizable. We therefore first identify task-critical points with a saliency-guided score, then greedily group them

1556-6021 © 2026 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: Nanjing University. Downloaded on November 23,2026 at 09:57:54 UTC from IEEE Xplore. Restrictions apply.

into a few connected regions on the LiDAR projection plane, and finally map these regions to candidate patch locations on the object surface. The procedure jointly accounts for attack effectiveness, region connectivity, and deployment cost. We next measure how 2D specular patches affect LiDAR returns under different reflection angles, and implement the selected regions in the physical world with plane mirrors. The same digitally generated patches are further transferred to other models and to real LiDAR captures.

To summarize, this paper makes the following contributions:

- **Model-guided digital-to-physical methodology.** We establish a systematic pipeline that translates model-guided point-cloud analysis into physically realizable surface patches. The pipeline identifies task-critical points, organizes them into a small number of connected regions in the LiDAR projection plane, and maps these regions to patch locations on an existing object's surface. It jointly considers attack effectiveness, region connectivity, physical realizability, and deployment cost.
- **Object-specific physical attack using 2D reflective patches.** We exploit passive specular materials to selectively suppress critical LiDAR returns from existing objects, providing a low-cost physical attack mechanism without specialized laser-emitting devices or custom-fabricated 3D objects.
- **Comprehensive evaluation.** We evaluate the proposed attack against ten state-of-the-art point cloud models, achieving significant accuracy degradation in both digital and physical domains, and further demonstrating strong transferability under black-box settings. We also extend the evaluation from classification to LiDAR object detection on the KITTI benchmark, where connected return suppression substantially reduces Moderate 3D AP for cars, pedestrians, and cyclists.
- **Broader implications.** Our findings expose an effective attack surface for LiDAR-based perception, emphasizing the urgent need for developing more robust and resilient defense mechanisms in safety-critical applications.

## II. Related Work

### A. Point Cloud Recognition

Extensive research in deep learning has been conducted on classification tasks using point clouds, primarily due to their ability to represent objects within a 3D scene, e.g., [28], [29], [34], [45].

VoxNet [28] utilizes a voxel-based descriptor to represent 3D data and employs a convolutional neural network (CNN) architecture for object recognition. MVCNN [34] uses view-based descriptors by converting 3D data into multiple images with different views and subsequently works with CNN architectures. PointNet [29] is the pioneering model capable of directly processing raw point cloud data while maintaining invariance to the input point cloud. It employs a max-pooling operation to extract global features of a fixed length. PointNet++ [30] is an enhanced version of PointNet that employs a hierarchical structure to capture both local and global features within point cloud data, resulting in improved performance. Apart from methods that process voxel-/view-based descriptors or raw point clouds, DGCNN [43] processes point clouds by transforming them into graphs. Recent advances include ASSANet [32], PointMLP [26], PointNeXt [33], PointVector [6], and Pix4Point [31], which further improve point cloud processing through efficient architectures, novel representations, and cross-modal pretraining.

### B. Adversarial Point Clouds

The concept of adversarial example was first proposed for images [2], [8], [17], [37]. The basic idea of adversarial attacks on point clouds is to find the critical points and manipulate these points to affect the accuracy of the classification models. Specifically, various methods can be employed for this manipulation, including perturbation [9], [10], [23], [47], adding points [47], and reducing points [48], [52]. Xiang et al. [47] proposed an attack that can craft adversarial point clouds against PointNet by generating or perturbing points. For point removal attacks, several common methods are used, such as random deletion, importance-based deletion, and structured deletion. Yang et al. [48] proposed a point cloud deletion algorithm based on the PointNet framework. Zheng et al. [52] proposed a more general importance scoring method, which computes saliency scores for 3D point cloud data based on the saliency map of 2D images. They approximated the process of deleting points as the process of moving points toward the center of the object.

### C. Physical Adversarial Attacks

In addition to adversarial attacks on 3D objects in the digital world, recent research has migrated attacks to the physical world, e.g., [1], [3], [7], [18], [24], [35], [39], [41], [53], [54]. Reference [1] explored the vulnerability of LiDAR to laser spoofing attacks and revealed that LiDAR-based systems can be fooled under effective laser spoofing attacks in vehicle detection scenarios. Reference [35] proposed a white-box attack method for laser spoofing attacks. They assumed that by limiting the number of adversarial laser points, as well as controlling their position and angle at which they are generated, a more precise attack could be achieved. Tu et al. [39] proposed a general method for generating adversarial objects. The adversarial objects were placed in fixed positions so that the point cloud data captured by other LiDAR detectors could not be classified correctly. Reference [24] explored the detection time problem of LiDAR-based systems and proposed an algorithm that maximizes the detection runtime through adversarial perturbation attacks. Reference [3] generated the adversarial example by applying a genetic algorithm with a simulated annealing strategy to fool LiDAR. Kobayashi et al. [16] proposed Shadow Hack, which models an adversarial shadow as a trapezoid and optimizes its dimensions in simulation so that KITTI-trained detectors falsely recognize the resulting empty region as a nonexistent vehicle.

Existing physical LiDAR attacks generally face limitations in either deployment cost or attack generality. Active laser attacks require specialized devices and precise signal control, while customized 3D adversarial objects incur substantial fabrication and deployment costs. Although passive attacks such

Authorized licensed use limited to: Nanjing University. Downloaded on November 23,2026 at 09:57:54 UTC from IEEE Xplore. Restrictions apply.

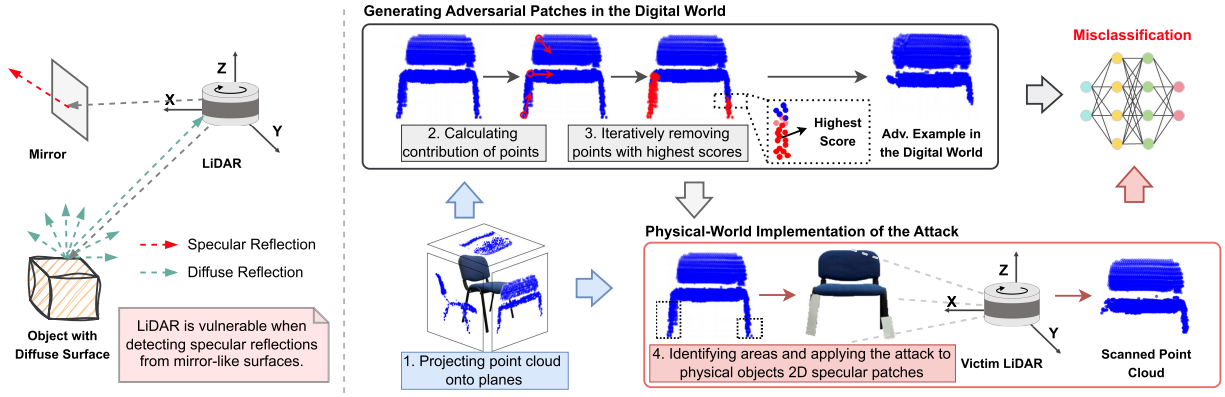



Fig. 1. Basic idea and the framework of our attacks.

as Shadow Hack significantly reduce these costs, their attack construction relies on predefined geometric templates and scenario-specific priors, and their effectiveness may depend on particular model representations.

Our work further explores passive manipulation in an object-centric setting. We establish a model-guided digital-to-physical methodology that identifies task-critical points on an existing object, organizes them into a small number of connected and physically realizable regions, and maps these regions to reflective patches on the object's surface. This formulation jointly considers attack effectiveness, region connectivity, physical realizability, and deployment cost, enabling low-cost manipulation of the perceived geometry and class of existing objects.

## III. Generating 2D Patches Against LiDAR Perception

Our method aims to manipulate LiDAR perception by removing regions of points in the captured point clouds using 2D specular reflective patches (e.g., mirrors, polished metal stickers), ultimately leading to misclassification by the victim model. We start by modeling the problem, establishing constraints, and exploring an effective method in the digital world that can be effectively implemented in the physical world. The framework of our attacks is presented in Fig. 1.

### A. Preliminaries and Overview

*1) LiDAR Sensors and Point Cloud Models:* A LiDAR sensor emits laser pulses and estimates the distance between the LiDAR and the object based on the time of receiving the reflected laser pulses. By emitting laser pulses in various directions, the LiDAR sensor can capture a point cloud that represents the detected objects. Usually, the horizontal and vertical fields of view (FOV) and the corresponding angle resolutions are used to measure the performance of a LiDAR sensor. We denote a point cloud captured by a LiDAR as $X \in \mathcal{X}$, where $X = \{x_1, x_2, \ldots, x_N\}$ is a collection of $N$ points. Each point $x_i$ consists of four dimensions, where the first three dimensions represent the 3D coordinates of the point and the fourth dimension represents reflectivity. Since most point cloud models do not consider reflectivity, we only focus on the first three dimensions and denote a point as $x_i = (x_{i1}, x_{i2}, x_{i3})$. We define $F : \mathcal{X} \to \mathcal{Y}$ as the classification function of the 3D point cloud object recognition model. Given the point cloud $X$, $F(X)$ produces its label $Y \in \mathcal{Y}$. The classification loss of the model $F$ is denoted by $\mathcal{L}$.

*2) Point Cloud Saliency:* We review how point cloud saliency [52] scores each point. Denote by $x_{\text{center}}$ the center position of all the points, and recall that $\mathcal{L}$ is the loss of the recognition model. Based on [52], the importance score of the $i$-th point $score_i$ can be defined as

$$score_i = -g_i \cdot r_i^{1+\alpha}, \tag{1}$$

where $\alpha$ is a parameter used to rescale the point clouds; $g_i$ is the the gradient of the loss function at the $i$-th point, which is defined as

$$g_i = \frac{\partial \mathcal{L}}{\partial r_i}, \tag{2}$$

where $r_i$ is the distance from the $i$-th point to $x_{\text{center}}$.

*3) Threat Model:* The goal of the attacker is to induce an untargeted misclassification by the victim model, without requiring control over the specific incorrect class. The attacker manipulates LiDAR perception by attaching several 2D specular reflective patches (e.g., mirrors or polished metal stickers) to the target object's surface, thereby suppressing returns from selected regions of the captured point cloud. The attack requires temporary physical access to the target object and control over the position, shape, and orientation of a small number of patches. The attacker also needs an approximate geometric model of the target object and its approximate pose relative to the LiDAR. To simplify attack generation, we assume that the attacker knows the type of LiDAR sensor being used [1], [14] and, in most cases, has white-box access to the victim point cloud model [1], [14], [47], [48]. We additionally evaluate the black-box setting in which the attacker does not know the victim model and examine the transferability of the attack. The attacker is not assumed to control the LiDAR hardware, scanning process, or downstream software.

*4) Methodology Overview:* To achieve the attack goal, our study is structured around two key aspects. (1) For a target object and its corresponding point cloud data, we explore the effect of removing critical points and design a physically

Authorized licensed use limited to: Nanjing University. Downloaded on November 23,2026 at 09:57:54 UTC from IEEE Xplore. Restrictions apply.

TABLE I
SUMMARY OF THE PRINCIPAL NOTATION

| Symbol | Meaning |
|---|---|
| $X, X_{\text{adv}}$ | Benign and adversarial point clouds |
| $F, \mathcal{L}$ | Victim classifier and classification loss |
| $x_i, r_i$ | Point $i$ and its radial distance from the cloud center |
| $\alpha_s$ | Saliency radial-rescaling exponent |
| $n, k, T$ | Total deletion budget, points per patch, and number of rounds |
| $S, S_t$ | All removed points and the region removed in round $t$ |
| $\alpha_m, \beta_m$ | Horizontal and vertical mirror-plane angles |
| $\Omega(\mathbf{n})$ | Reflection-induced invisible region for normal $\mathbf{n}$ |

realizable method for point removal in the digital world. This step is crucial in assessing how the changes in the targeted point cloud data could affect model accuracy. Before implementing these patches in the physical world, it provides guidelines for attackers to generate adversarial patches in the digital world. (2) We explore the effect of 2D specular reflections on LiDAR using mirrors and implement the generated digital-world patches in the physical world. Although we intuitively understand that specular reflection impacts laser reflection, thereby affecting LiDAR perception, we still require quantitative measurements to determine its impact on point distance and the extent to which it affects point cloud capture. By identifying the characteristics revealed in this analysis, we can subsequently select a more effective patch implementation method in the physical world.

### B. Adversarial 2D Patches Generation

*1) Problem Formulation:* In typical scenarios without specular reflection, LiDAR measures the distance from objects by emitting laser pulses and detecting the time it takes to receive their diffuse reflection. Since the speed of light is constant, the time needed in the process can be used to infer the distance. However, with the presence of specular reflection surfaces, such as mirrors, some laser pulses will be reflected off the surfaces, altering the time it takes for the LiDAR sensor to receive them. At specific angles, LiDAR may never receive the laser pulses reflected by the specular reflection surfaces. In addition, the laser pulses emitted to the target object may be reflected to other objects that may be far away from the target object or even out of the scan range of the LiDAR sensor via specular reflection surfaces, and these laser pulses will eventually be received by LiDAR after a significant delay. As a result, the LiDAR sensor could not capture the accurate point clouds of objects; usually, some parts will disappear in the point clouds of an object, leading to a misunderstanding of the subsequent deep learning model digesting the point cloud data. The objective of the attack is to generate adversarial 2D patches using specular reflective material, causing the LiDAR device to capture manipulated point clouds, thereby misleading the 3D deep object recognition model.

It is infeasible to try all possible positions and shapes of patches by placing specular reflective patches on the surface of the target object physically. Therefore, our approach first generates adversarial patches in the digital world, with the constraint that the patches can be realized physically. Given an original point cloud $X$ and its corresponding label $Y$, the goal is to find an adversarial example $X_{\text{adv}}$ based on $X$ such that $F(X_{\text{adv}}) \neq Y$, with the constraint that $X_{\text{adv}}$ can be realized by adding physical 2D specular reflective patches on the surface of the corresponding object. Based on the law of reflection, the points that represent the covered area of the object will disappear from the point cloud data when reflective patches are placed at specific angles. Therefore, the attack goal can be reduced to finding an adversarial example satisfying $X_{\text{adv}} \subset X$, which means to remove points from the point cloud and treat the removed areas as patches.

*2) Challenges and Method:* There are two major challenges in generating an adversarial example for our problem: 1) making the removed areas satisfy the physically realizable requirement of 2D patches; 2) choosing the optimal areas to remove points.

For the first challenge, we have observed that a patch not only reflects off the points representing the attached surface area from the LiDAR sensor but also obstructs all points behind the patch within the LiDAR's field of view. Therefore, we employ a projection operation to identify the points that need to be removed in the 3D point cloud. By projecting all the points on the lines of sight (LOS) of the LiDAR sensor to a plane, it allows us to ignore the depth information along the directions of the LOS of the LiDAR and simplify the strategy of choosing points to remove. When we project the point cloud onto a horizontal plane parallel to the plane of the patch, the points need to be deleted from a closed region within that plane. Without loss of generality, we define this projection plane as the plane $x = 0$ in the Cartesian coordinate system.

For the second challenge, a straightforward approach would be to enumerate all possible areas for removal and select the best one. However, this method is computationally infeasible due to the large search space of candidate areas. Furthermore, generating multiple areas with varying shapes and sizes would aggravate the difficulty. We adopt point cloud saliency maps [52], which can characterize the contribution of each point on the loss of the point cloud object recognition model, as a basic component, and a greedy strategy to identify a continuous and closed area of points. Without considering the physically realizable requirement, a straightforward approach could repeat the following steps: compute the scores of remaining points in each round based on Eq. 1, then select and delete the point with the highest score. However, such a method cannot apply to cases of using 2D specular reflective patches to remove points.

For each round of removing points, our method first computes the scores of all the remaining points. Then the method projects all the remaining points as the above and adopts a greedy strategy to choose which points should be deleted. Points with high importance scores are often surrounded by important areas. Therefore, we delete $k$ points nearest to the point with the highest score in each turn. Formally, we denote the $i$-th point by $x_i = (x_{i1}, x_{i2}, x_{i3})$. We define a function $dist_{x=0}$ as the distance between two points on the projection plane $x = 0$, the distance between $x_i$ and $x_j$ can be defined as

$$dist_{x=0}(x_i, x_j) = \sqrt{(x_{i2} - x_{j2})^2 + (x_{i3} - x_{j3})^2}. \quad (3)$$

Authorized licensed use limited to: Nanjing University. Downloaded on November 23,2026 at 09:57:54 UTC from IEEE Xplore. Restrictions apply.

**Algorithm 1** Remove Points on the Projection Plane

**Input:** The well-trained model with loss $\mathcal{L}$; point cloud input $X$; $k$: number of points to be deleted per round; $n$: number of points to be deleted in total.

$T \leftarrow \frac{n}{k}$;
**for** $p \leftarrow 1$ to $T$ **do**
  Center position: $x_{\text{center}} = median(x_{i1}, x_{i2}, x_{i3})$;
  The gradient of $\mathcal{L}$: $g_i = \frac{\partial \mathcal{L}}{\partial r_i}$;
  The importance score: $score_i = -g_i \cdot r_i^{1+\alpha}$;
  $x_v \leftarrow$ the points with the highest score;
  $dist^i_{x=0}(x_i, x_v) = \sqrt{(x_{i2} - x_{v2})^2 + (x_{i3} - x_{v3})^2}$;
  $I_{\text{delete}} \leftarrow \text{argmink}_{i=1,\ldots,n} dist^i_{x=0}$;
  Delete points $x_j$ $(j \in I_{\text{delete}})$ from $X$;
**end for**

Denote by $x_v$ the point with the highest score in the current round, and the points that need to be removed in this round are $k$ points nearest to the $x_v$, which the distance is measured using the distance function $dist_{x=0}$. In scenarios involving multiple patches, several areas of points on the plane will need to be removed. Assuming a total deletion of $n$ points is required, and with $k$ points being deleted in one round, we would consequently need $T = \frac{n}{k}$ patches in the real-world setting. The strategy is illustrated in Algorithm 1. Note that computing $g_i = \partial L / \partial r_i$ requires access to the model gradients. Therefore, the proposed patch generation algorithm is designed for white-box or surrogate white-box settings, and in our black-box evaluation we rely on transferability of such patches to victim models.

*3) Theoretical Interpretation and Physical Feasibility:*

*a) LiDAR reflection model:* Let the LiDAR emitter be $L = (0, 0, 0)$ and a surface point be $P = (x, y, z)$ with normal vector $\mathbf{n}$. The incident direction is $\mathbf{d} = \frac{P-L}{\|P-L\|}$, and the reflected direction follows the law of reflection:

$$\mathbf{r} = \mathbf{d} - 2(\mathbf{d} \cdot \mathbf{n})\mathbf{n}. \tag{4}$$

Assuming $\|\mathbf{d}\| = \|\mathbf{n}\| = 1$, the LiDAR can receive the reflected beam only if the reflected ray falls within its receiving cone, i.e.,

$$\angle(\mathbf{r}, -\mathbf{d}) \leq \delta \iff -\mathbf{r} \cdot \mathbf{d} \geq \cos\delta, \tag{5}$$

where $\delta$ denotes the half-angle of the LiDARs receiving field of view (FOV). A point $P$ becomes *invisible* when this constraint is violated, forming the reflection-based occlusion region

$$\Omega(\mathbf{n}) = \{P \mid \angle(\mathbf{r}(P), -\mathbf{d}(P)) > \delta\}.$$

*b) Optimization formulation:* We model the attack as the selective removal of points that violate the visibility condition. Given a point cloud $X$ and a classifier $F(\cdot)$ with ground truth label $y$, the objective is formulated as

$$\min_{S \subset X} L(F(X \setminus S), y), \quad \text{s.t. } S \subseteq \Omega(\mathbf{n}),\ |S| \leq n. \tag{6}$$

Here, $S$ is the subset of removed points, and $n$ denotes the maximum number of deletable points. For physical realizability, the removed region is modeled as a union of at most $T = \lceil n/k \rceil$ connected patches on the projection plane:

$$S = \bigcup_{t=1}^{T} S_t, \quad S_t \subseteq \Omega(\mathbf{n}_t), \quad \text{diam}_{x=0}(S_t) \leq \Delta, \tag{7}$$

where each $S_t$ corresponds to one reflective patch with bounded size $\Delta$ and normal $\mathbf{n}_t$. This formulation aligns each greedy deletion round in Algorithm 1 with one mirror patch in the physical world.

*c) Saliency-guided removal:* Empirically, points with high saliency values computed from the gradient-based importance scoring are often located near geometric edges or surface discontinuities. These regions are also more sensitive to specular reflections, where LiDAR returns are likely to vanish because of the angular misalignment between the incident and reflected beams. Hence, although the correspondence between saliency and reflection-induced point loss is not strictly deterministic, the saliency-guided removal strategy remains consistent with the physical characteristics of LiDAR sensing and provides a plausible connection between digital optimization and physical realizability.

*d) Area–point relationship:* The number of removed points is approximately proportional to the projected patch area along the line of sight. Let $\beta$ denote the angle between the patch *plane* and the beam, as defined in our measurement setup. Since the incidence angle between the patch normal and the beam is $\theta = 90^\circ - \beta$, the visible area projected onto the sensor is scaled by $\cos\theta = \sin\beta$, yielding

$$|S| \approx \rho A \sin\beta, \tag{8}$$

where $\rho$ is the point density determined by the LiDARs angular resolution and $A$ is the physical patch area. This relation connects the geometric configuration with the density of removed points observed in both digital simulation and real experiments.

### C. Physical-World Attack Realization

*1) Probing the Specular Reflection on LiDAR:* We use plane mirrors to explore the effects of specular reflection on LiDAR detection. Specifically, we explore the effects by conducting measurements in two settings. First, we configure the LiDAR system to emit unidirectional laser pulses to measure the influence of plane mirrors on the distance perception of LiDAR. This setting allows us to obtain precise measurements of how a plane mirror reflection alters the perceived distance for a single point in the point clouds. Second, we configure a more general setting where the LiDAR sensor uses its full horizontal and vertical fields of view (FOV). This setting allows the LiDAR system to capture the point clouds representing a whole object, which is similar to a typical LiDAR detection use case.

Unidirectional laser pulses: The measurement experiments are performed in a rectangular area measuring approximately 8 meters by 6 meters, with a ceiling height of about 3 meters from the floor. A mirror is placed in the center of the floor. The LiDAR device is placed approximately 2 meters away from the mirror, directly facing it, as illustrated in the left of Fig. 3. To explore the effects of mirror reflections with various angles, we adjust two angles, $\alpha$ and $\beta$, for the mirror, as depicted in the right of Fig. 3. The angle $\alpha$ represents the horizontal angle between the plane of the mirror and the direction of the laser from the LiDAR device. Adjusting the angle $\alpha$ enables the laser to be reflected towards various surrounding objects. The angle $\beta$ is defined as the vertical

Authorized licensed use limited to: Nanjing University. Downloaded on November 23,2026 at 09:57:54 UTC from IEEE Xplore. Restrictions apply.

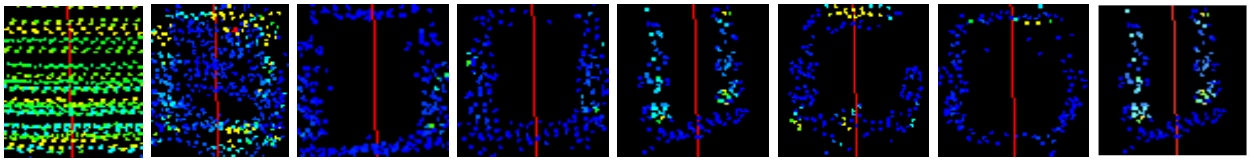



Fig. 2. The first figure depicts a controlled experiment in the right of Fig. 4, while the other images show the LiDAR's view from different mirror angles with $(\alpha, \beta)$.

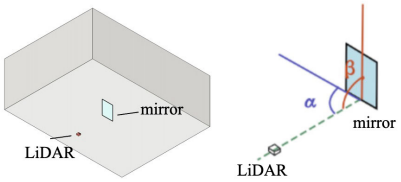



Fig. 3. Experimental setup for specular reflection with unidirectional laser pulses. The left is the experimental scene and the right is to illustrate the angles in the experiments.

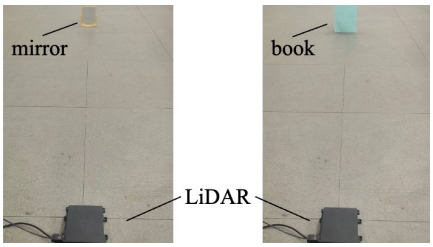



Fig. 4. Experimental setup to investigate the impact of specular reflection for laser pulses from various directions.

TABLE II
SINGLE LASER SPECULAR REFLECTION ON DIST. ESTIMATION

| $\alpha$,$\beta$ | ground truth | $90^\circ, 90^\circ$ | $120^\circ, 90^\circ$ | $135^\circ, 90^\circ$ |
|---|---|---|---|---|
| distance(m) | 2.014 | 2.026 | 9.353 | 7.533 |
| $\alpha$,$\beta$ | $150^\circ, 90^\circ$ | $90^\circ, 120^\circ$ | $90^\circ, 135^\circ$ | $90^\circ, 150^\circ$ |
| distance(m) | 7.483 | 6.768 | 4.817 | 5.065 |

angle between the plane of the mirror and the direction of the laser from the LiDAR device. Configuring the angle $\beta$ allows the laser to be reflected towards the ceiling of the space. The actual distance from LiDAR to the mirror is 2.014 meters. We choose several settings with different $\alpha$ and $\beta$ combinations for distance measurement. The experimental results are detailed in Table II, which shows the following conclusion: under the case of unidirectional laser pulses, the measured distance will be overestimated at a specific angle, due to reflections in the closed environment.

Full horizontal and vertical FOV: For the second setting, we have placed the LiDAR device in the center of the room and a book about 2 meters in front of the LiDAR as a control case. The setup of the scene and the point clouds generated from the LiDAR scanning are shown in Fig. 4. Then instead of the book, a mirror is placed with different angles. Similar to the first setting, we choose various angles for setting up the mirrors. The point clouds generated from the LiDAR scanning are depicted in Fig. 2 (The red line in the picture indicates the area directly in front of the LiDAR). From these images, it is evident that when the mirror is at a specific angle relative to the LiDAR, the laser either travels farther than it actually should or never returns. Consequently, the point cloud representing the mirror's location disappears.

*2) Physically Implementation of Patches:* We utilize PointNet as a white-box model to identify the areas that need to be removed. We employ a method similar to Algorithm 1, but with a different condition for termination. Specifically, for physical attacks, instead of limiting the number of total points to be removed, we iteratively execute the point removal process until the PointNet model misclassifies the input. Subsequently, we eliminate these areas from LiDAR perception by applying specular reflection with several plane mirrors as adversarial patches for the target object.

Based on the measurements conducted above, we are able to choose proper angles of placing the mirrors to interfere with LiDAR perception. It is important to note that if multiple areas in the point cloud require removal, we typically employ multiple plane mirrors. However, in cases where these areas are connected or very close together, we can use a single piece of plane mirror to cover them collectively.

## IV. EVALUATION

We evaluate the experiments from three aspects. 1) Verify the results on the benchmark dataset to validate the effectiveness of the proposed algorithm. 2) Implement the attack and evaluate the feasibility of the attack in the physical world. 3) Investigate the transferability of the attack against black-box victim models.

### A. Dataset and Training Details

For the experiments in the digital world, the benchmark datasets used are ModelNet40 [46] and ScanObjectNN [40]. The ModelNet40 dataset consists of CAD models spanning 40 categories. We adopt the official split, using 9843 models for the training set and 2468 models for the testing and attack sets. Models are trained with the Adam optimizer, a learning rate of 0.001, a weight decay of 0.05, and a batch size of 32 for 100 epochs. The ScanObjectNN dataset is a real-world 3D dataset containing approximately 2.9$k$ samples across 15 distinct classes. Our experiments are conducted on PB_T50_RS, the most challenging and widely used variant of ScanObjectNN. Models are trained with the Adam optimizer, a learning rate of 0.002, a weight decay of 0.05, and a batch

Authorized licensed use limited to: Nanjing University. Downloaded on November 23,2026 at 09:57:54 UTC from IEEE Xplore. Restrictions apply.

TABLE III
BENCHMARK ACCURACY (%) GROUPED BY ARCHITECTURAL FAMILY. LOWER ACCURACY INDICATES A STRONGER ATTACK

| Dataset | Family | Model | Clean | Random | Drop-Area | | Our Attack | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | $k = 5$ | $k = 50$ | $k = 5$ | $k = 50$ | $k = 150$ |
| ModelNet40 | Pointwise MLP | PointNet | 90.59 | 89.71 | 64.34 | 64.34 | **35.25** | 47.89 | 51.86 |
| | Hierarchical | PointNet++ | 92.10 | 91.09 | 61.18 | 61.18 | **46.07** | 48.54 | 52.23 |
| | Convolution | PointConv | 90.68 | 88.13 | 71.23 | 71.07 | **56.24** | 60.82 | 61.83 |
| | Graph | DGCNN | 89.14 | 86.30 | 78.16 | 78.16 | **50.53** | 60.78 | 65.60 |
| | Graph | DeepGCN | 92.67 | 90.07 | 84.93 | 84.93 | **68.72** | 72.85 | 75.24 |
| | Hierarchical | ASSANet | 91.21 | 90.07 | 68.72 | 68.72 | 69.00 | 64.87 | **63.74** |
| | MLP | PointMLP | 92.14 | 90.07 | 79.25 | 79.25 | **57.58** | 66.09 | 69.69 |
| | Hierarchical | PointNeXt | 92.42 | 91.25 | 79.74 | 79.74 | 71.35 | **70.26** | 72.33 |
| | Hierarchical | PointVector | 92.63 | 91.09 | 83.91 | 83.91 | 72.33 | **72.04** | 72.08 |
| | Cross-modal | Pix4Point | 91.65 | 90.92 | 83.51 | 83.51 | **62.68** | 72.41 | 75.28 |
| ScanObjectNN | Pointwise MLP | PointNet | 66.17 | 64.68 | 36.88 | 36.88 | **34.66** | 36.36 | 36.50 |
| | Hierarchical | PointNet++ | 83.31 | 83.73 | 56.97 | 57.01 | 53.02 | 48.96 | **48.68** |
| | Convolution | PointConv | 82.96 | 80.90 | 71.26 | 69.88 | **61.96** | 64.03 | 65.74 |
| | Graph | DGCNN | 82.58 | 78.49 | 72.69 | 72.76 | **64.75** | 65.37 | 65.54 |
| | Graph | DeepGCN | 80.71 | 77.31 | 71.69 | 71.76 | **64.47** | 65.37 | 66.27 |
| | Hierarchical | ASSANet | 80.60 | 75.08 | 59.47 | 59.56 | 45.21 | **43.02** | 43.38 |
| | MLP | PointMLP | 84.80 | 81.05 | 74.57 | 74.53 | 67.10 | **66.34** | 66.66 |
| | Hierarchical | PointNeXt | 82.41 | 80.33 | 69.08 | 69.08 | 60.44 | **58.15** | 58.61 |
| | Hierarchical | PointVector | 85.15 | 82.93 | 75.36 | 75.40 | 66.51 | **63.11** | 64.19 |
| | Cross-modal | Pix4Point | 85.63 | 83.27 | 77.37 | 77.38 | 72.79 | 72.24 | **71.65** |

size of 32 for 250 epochs. The total deletion budget is $n = 300$, and the realizable attack removes $k = 50$ projected neighbors per round for $T = 6$ rounds.

### B. Baselines

We consider two baseline attacks: the first attack randomly deletes $k$ points in each round (denoted by Drop-Random in the figures); the second attack randomly chooses one point and deletes $k$ points nearest to it on the projection plane $x = 0$, including the chosen point itself, in each round (denoted by Drop-Area in the figures). We use the accuracy of the victim model on the attack set to measure the performance. We note that, in our experiments, a lower accuracy indicates a better attack performance, as the attacker's goal is to induce misclassification in the victim model by manipulating the inputs.

### C. LiDAR and Physical Objects

We utilize plane mirrors as the specular reflection patches to remove the points from the point clouds captured by the LiDAR sensor. We use a Livox HAP 3D LiDAR, which has a detection range of 150 meters, $120° \times 25°$ FOV, and $0.18° \times 0.23°$ angle resolution, to collect the point cloud data in the experiments. We select five chairs, five monitors, and five plants as the target objects for the experiments in the physical world. For the physical-world experiments, we also collect the point clouds of adversarial objects and use them to attack point cloud object recognition models to evaluate the transferability of the attack under the black-box attack setting.

### D. Models

For both the digital-world and physical-world experiments, four state-of-the-art models, including PointNet [29], PointNet++ [30], DGCNN [43], PointConv [45], DeepGCN [19], ASSANet [32], PointMLP [26], PointNeXt [33], PointVector [6] and Pix4Point [31] are used as victim models in our experiments.

*1) Attack Performance on the Benchmark Dataset:* In the first set of experiments, we set the total number of points to be removed as $n = 300$. We vary the value of $k$, which represents the number of points removed in each round, from 5 to 300. The results are shown in Table III. From these results, it is observed that our attack outperforms two baseline attacks significantly. For the baseline attacks, it can be observed that the random point drop method (Drop-Random) has negligible effects on the accuracy of the models. If we randomly drop a cluster of points located on the projection plane $x = 0$ (Drop-Area), there is a noticeable decrease in the accuracy of the models. This is because a point cluster in a specific area often carries more significant features of the point cloud.

The architecture grouping in Table III reveals a consistent but non-uniform sensitivity. PointNet, which relies on pointwise MLPs with a single global max-pooling, suffers the largest ModelNet40 accuracy drop at $k = 50$ (42.70 percentage points), whereas graph-based and recent hierarchical models generally retain more accuracy. This is consistent with local redundancy: architectures that aggregate neighborhoods at several stages can preserve evidence outside one missing region, while a global critical-point representation is more directly affected.

In addition, we have observed that as the number of points removed in each round increases, the attack performance gets worse for most models. This decline is attributed to the reduced number of rounds available for removing point clusters, which limits the capability of the attacker to optimize the attack. However, opting for a smaller value of $k$ does not necessarily imply better performance for our attack. For instance, if we select $k = 5$ with $n = 300$, we would require $n/k = 60$ patches to execute the attack. A higher number of patches notably escalates the difficulty of implementing the attack in the physical world. Based on the results in Table III,

Authorized licensed use limited to: Nanjing University. Downloaded on November 23,2026 at 09:57:54 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
RESULTS ON TRANSFERABILITY FOR BOTH THE DIGITAL AND PHYSICAL WORLDS: ACCURACY OF THE VICTIM MODELS UNDER ATTACKS

| | White-Box → Black-Box (ModelNet40) | White-Box → Black-Box (ScanObjectNN) | Physical World (Black-Box Settings) |
|---|---|---|---|
| PointNet | - | - | - |
| PointNet++ | 48.54% → 50.12% | 48.96% → 48.96% | 2 / 14 |
| PointConv | 60.82% → 65.36% | 64.03% → 64.05% | 2 / 14 |
| DGCNN | 60.78% → 61.12% | 65.37% → 65.37% | 3 / 13 |
| DeepGCN | 72.85% → 71.02% | 65.37% → 65.75% | 3 / 14 |
| ASSANet | 64.87% → 63.61% | 43.02% → 43.03% | 3 / 13 |
| PointMLP | 66.09% → 62.56% | 66.34% → 66.34% | 3 / 15 |
| PointNeXt | 70.26% → 68.67% | 58.15% → 58.15% | 2 / 15 |
| PointVector | 72.04% → 68.92% | 63.11% → 63.12% | 5 / 13 |
| Pix4Point | 72.41% → 71.06% | 72.24% → 50.59% | 4 / 15 |

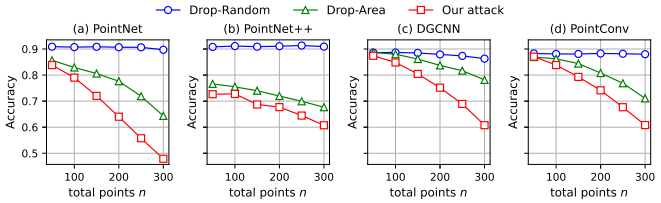



Fig. 5. Results of varying the attack intensity, i.e., the total number of points to be removed $n$, with $k = 50$ points removed in each round on ModelNet40 dataset.

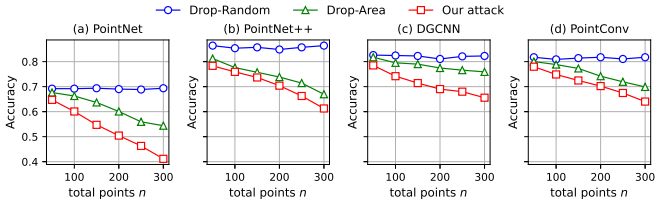



Fig. 6. Results of varying the attack intensity, i.e., the total number of points to be removed $n$, with $k = 50$ points removed in each round on ScanObjectNN dataset.

we choose $k = 50$ for the following experiments to balance the difficulty of implementation and the attack performance.

In the second set of experiments, we set the attack to remove $k = 50$ points in each round and vary the total number of points to be removed, $n$, from 50 to 300. As shown in Fig. 5 and Fig. 6, the proposed attack outperforms baseline attacks across various attack intensities. For ModelNet40 dataset, when $n = 300$, the accuracy decreases from 91.31% to 47.89% for PointNet, from 92.24% to 60.73% for PointNet++, from 90.75% to 60.77% for DGCNN, and from 89.33% to 60.81% for PointConv. For ScanObjectNN dataset, the accuracy decreases from 69.36% to 41.13% for PointNet, from 86.40% to 61.27% for PointNet++, from 82.27% to 65.57% for DGCNN, and from 81.75% to 64.03% for PointConv. It is evident that our attack significantly diminishes the classification performance of various state-of-the-art point cloud object recognition models, making high-accuracy models perform poorly.

*a) Attempted targeted classification:* We replace the untargeted objective in Eq. 1 with the negative cross-entropy of a predetermined target. For each clean-correct input, we select the runner-up class from the clean prediction and fix it as the attack target before optimization. Under the same deletion budget ($n = 300$, $k = 50$), we optimize connected deletion regions toward this target on the full 2,468-sample ModelNet40 test set. The resulting targeted ASRs are 18.10% on PointNet and 9.59% on DGCNN, which are too low to constitute a reliable targeted attack.

*b) Comparison with iterative saliency-topk and runtime:* Table VI compares all methods on the full ModelNet40 test set under the same deletion budget of $n = 300$ and $k = 50$. On PointNet, whose clean accuracy is 90.59%, iterative Saliency-TopK reduces the accuracy to 53.81%, whereas our connected-region method further reduces it to 52.47%. The same ordering is observed on DGCNN: from a clean accuracy of 89.14%, Saliency-TopK and our method reduce the accuracy to 59.08% and 58.14%, respectively. Although Saliency-TopK selects points without a connectivity constraint, it evaluates point-wise saliency and may produce spatially scattered deletions with redundant effects. In contrast, our method exploits local spatial structure to remove coherent geometric evidence, resulting in consistently stronger attack performance while maintaining physical realizability. The attack-generation time on PointNet is 45.7 ms per sample for our method, compared with 48.6 ms for Saliency-TopK, 24.9 ms for Drop-Area, and 19.2 ms for Drop-Random. Since patch optimization is performed offline, this additional cost does not affect LiDAR inference latency.

*2) Attack Performance on LiDAR Object Detection:* Table III and Table VII answer complementary questions: the former tests breadth across classification architectures, whereas the latter tests whether connected return suppression affects a downstream detector in complete driving scenes. We evaluate a pretrained PointPillars detector on the full 3,769-frame KITTI validation split. For each labeled object, a PointNet surrogate selects returns within its 3D box; the selected connected region is then removed from the full scene before PointPillars inference. This is a black-box transfer setting because detector gradients and parameters are not used. The clean Moderate 3D $AP_{R40}$ values of 78.38%, 51.45%, and 62.93% for Car, Pedestrian, and Cyclist closely match the published OpenPCDet PointPillars reference values, validating the evaluation alignment. Our attack lowers them to 68.95%, 32.20%, and 36.00%, respectively. Averaged over the three classes, our method reduces AP by 18.53 points, compared with 6.62 for Drop-Random, 14.32 for Drop-Area,

Authorized licensed use limited to: Nanjing University. Downloaded on November 23,2026 at 09:57:54 UTC from IEEE Xplore. Restrictions apply.

TABLE V

PARTIAL RESULTS ON THE PHYSICAL-WORLD OBJECTS. FOR THE 2ND TO 5TH COLUMNS OF THE TABLE, IN EACH CELL, THE FIRST SYMBOL INDICATES WHETHER THE BENIGN POINT CLOUD CAN BE CORRECTLY CLASSIFIED BY THE CORRESPONDING MODEL (✓: CORRECT; ✗: INCORRECT). THE SECOND SYMBOL IN EACH CELL INDICATES THE OUTCOME OF THE ATTACK ON THE VICTIM MODEL (✓: SUCCESS; ✗: FAIL; -: NO ATTACK HAS BEEN CONDUCTED BECAUSE THE BENIGN POINT CLOUD HAS NOT BEEN CORRECTLY CLASSIFIED)

| | PointNet | Pointnet++ | DGCNN | PointConv | # of points removed | # of mirrors |
|---|---|---|---|---|---|---|
| chair01 | ✔/✔ | ✔/✔ | ✖/- | ✔/✔ | 250 | 2 |
| chair02 | ✔/✔ | ✔/✔ | ✔/✔ | ✔/✔ | 350 | 4 |
| chair03 | ✔/✔ | ✔/✔ | ✔/✖ | ✔/✔ | 150 | 2 |
| chair04 | ✔/✔ | ✔/✔ | ✔/✔ | ✔/✔ | 100 | 2 |
| chair05 | ✔/✔ | ✔/✔ | ✔/✔ | ✔/✖ | 150 | 3 |
| monitor01 | ✔/✔ | ✔/✔ | ✔/✔ | ✔/✔ | 350 | 3 |
| monitor02 | ✔/✔ | ✔/✔ | ✔/✔ | ✔/✔ | 100 | 2 |
| monitor03 | ✔/✔ | ✖/- | ✖/- | ✔/✔ | 300 | 4 |
| monitor04 | ✔/✔ | ✔/✖ | ✔/✔ | ✖/- | 250 | 1 |
| monitor05 | ✔/✔ | ✔/✔ | ✔/✔ | ✔/✔ | 200 | 2 |
| plant01 | ✔/✔ | ✔/✔ | ✔/✔ | ✔/✔ | 250 | 2 |
| plant02 | ✔/✔ | ✔/✔ | ✔/✔ | ✔/✔ | 100 | 2 |
| plant03 | ✔/✔ | ✔/✔ | ✔/✖ | ✔/✖ | 150 | 2 |
| plant04 | ✔/✔ | ✔/✔ | ✔/✔ | ✔/✔ | 50 | 1 |
| plant05 | ✔/✔ | ✔/✖ | ✔/✖ | ✔/✔ | 150 | 3 |

TABLE VI

UNTARGETED ATTACK ACCURACY AND RUNTIME ON THE FULL 2,468-SAMPLE MODELNET40 TEST SET WITH POINTNET

| Metric | Drop-Random | Drop-Area | Saliency-TopK | Our Attack |
|---|---|---|---|---|
| Acc. (%) | 88.65 | 65.10 | 53.81 | **52.47** |
| Time (ms/sample) | 19.2 | 24.9 | 48.6 | 45.7 |

TABLE VII

POINTPILLARS RESULTS ON ALL 3,769 KITTI VALIDATION FRAMES USING THE OFFICIAL OPENPCDET KITTI EVALUATOR. VALUES ARE MODERATE 3D $AP_{R40}$ (%)

| Method | Car | Pedestrian | Cyclist |
|---|---|---|---|
| Clean | 78.38 | 51.45 | 62.93 |
| Drop-Random | 75.61 | 45.35 | 51.94 |
| Drop-Area | 72.38 | 36.18 | 41.24 |
| Saliency-TopK | 74.30 | 43.13 | 49.92 |
| Our Attack | **68.95** | **32.20** | **36.00** |

and 8.47 for Saliency-TopK. Thus, the connected attack is consistently strongest across all three classes on the full split. These official-evaluator results demonstrate a substantial digital detection impact without claiming physical detection validation.

*3) Attack Performance on Physical Objects:* For attacking the LiDAR perception with physical objects, instead of limiting the total number of points need to be removed, the attack algorithm terminates only when it achieves the success attack instead, which the object is misclassified by the model. We choose to remove an area containing $k = 50$ points in each iteration.

In Fig. 7, for the physical-world attacks, our method removes 50 points in each round and terminates when the attack successfully fools the victim model. Since we treat PointNet trained on the ModelNet40 dataset as a white-box in our experiments, the attack achieves a 100% success rate against PointNet. The complete experimental results are presented in Fig. 7.

To evaluate the stealthiness of the physical attack, we measure the number of removed points, the quantity of plane mirrors used, and the total area covered by the mirrors.

TABLE VIII

RESULTS ON THE STEALTHINESS OF THE PHYSICAL ATTACKS. $n$ IS THE NUMBER OF POINTS NEED TO BE REMOVED; $n/N$ IS THE RATIO OF THE NUMBER OF REMOVED POINTS TO THE TOTAL NUMBER OF POINTS THE POINT CLOUD. EACH COLUMN WITH THREE NUMBERS THAT ARE (MIN, MAX, AVG) OF THE CORRESPONDING METRIC

| Object | Removed Points $n$ | Ratio $n/N$ (%) | # Mirrors | Mirror Area (1e-2 $m^2$) |
|---|---|---|---|---|
| Chair | 100, 350, 200 | 2.8, 12.5, 7.3 | 2, 4, 2.6 | 2.0, 11.0, 6.5 |
| Monitor | 100, 350, 240 | 4.3, 18.5, 9.5 | 1, 4, 2.4 | 1.0, 8.0, 5.0 |
| Plant | 50, 250, 140 | 6.1, 25.3, 14.7 | 1, 3, 2.0 | 1.5, 5.0, 2.7 |

Table VIII shows that the proposed attack can achieve its goal at an affordable cost. Theoretically, for every round where 50 points are deleted, a mirror is required to cover them. If $n'$ points need to be removed, $\frac{n'}{k}$ mirrors are required. An interesting observation is that the proposed attack needs fewer plane mirrors to implement the attack, as the small critical areas of the point cloud have a high probability of being combined and covered by one plane mirror together.

*4) Cost and Deployment Overhead Analysis:* Beyond attack effectiveness, practical deployment cost is a critical factor in evaluating the real-world feasibility of physical adversarial attacks. While prior LiDAR-based attacks often rely on specialized laser-emitting devices, synchronized control modules, or custom-fabricated 3D adversarial objects, the proposed method uses only commercially available reflective materials, such as plane mirrors.

As reported in Table. VIII, the number of mirrors required to successfully attack an object is relatively small, typically ranging from 1 to 4, with an average of approximately 2–3 mirrors depending on the object category. The total reflective area remains limited, generally within a few $\times 10^{-2}$ $m^2$. In our physical experiments, each mirror used costs only a few US dollars. Consequently, the total material cost per attacked object is generally below $20. No specialized hardware, electronic components, synchronization devices, or active light sources are required.

Authorized licensed use limited to: Nanjing University. Downloaded on November 23,2026 at 09:57:54 UTC from IEEE Xplore. Restrictions apply.

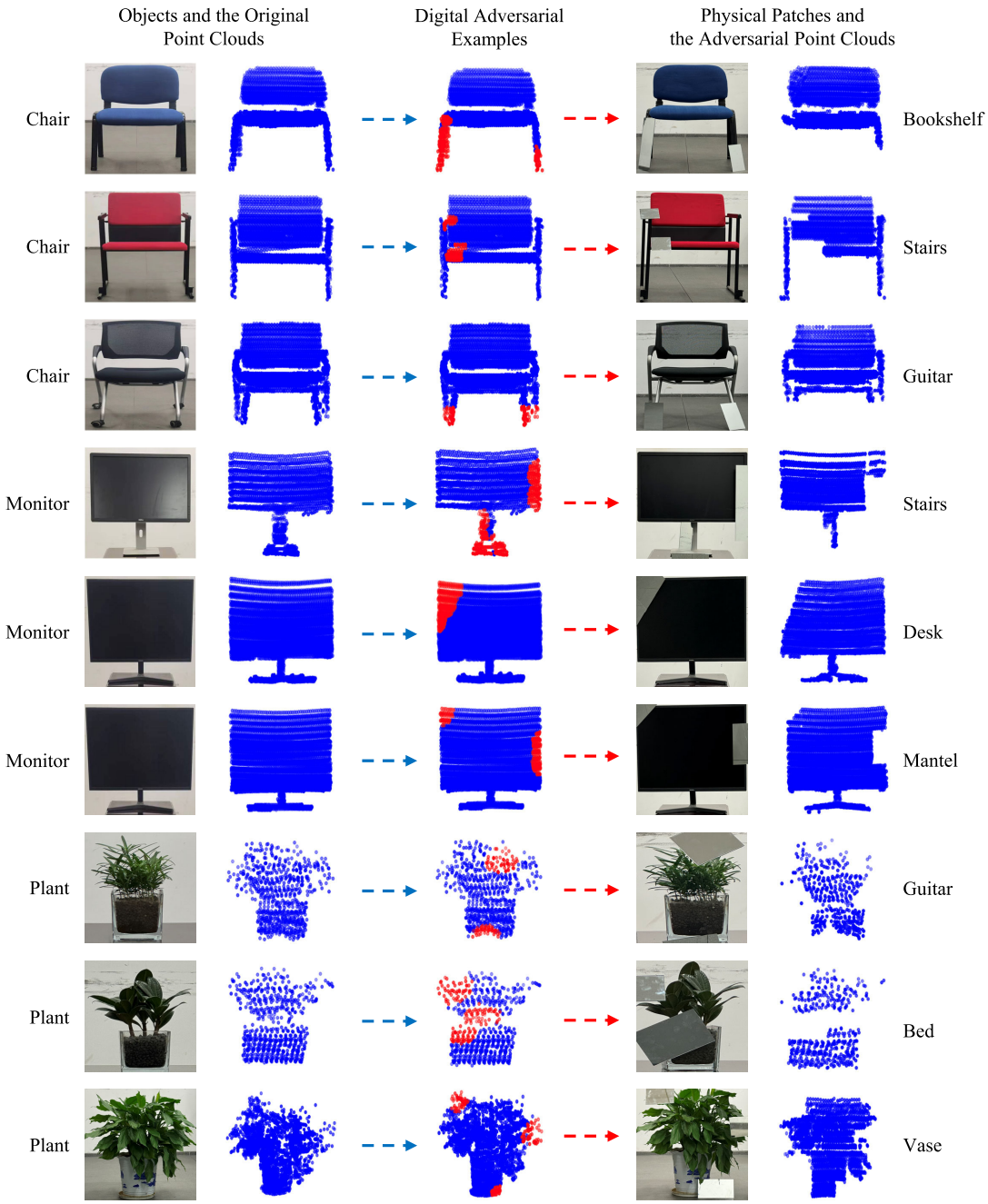



Fig. 7. Partial physical attack results. The left two columns are the target objects and their original point clouds; the middle column is the adversarial example generated in the digital world; the right two columns are the physical attacks where the plane mirrors are used to interfere with the LiDAR's capture of point clouds and the adversarial point clouds.

For comparison, representative physical LiDAR attacks in the literature involve substantially higher deployment overhead. Laser spoofing attacks require high-power laser emitters and precise synchronization modules, whose cost can easily exceed $1,000 and demand careful calibration. Attacks based on custom 3D adversarial objects require fabrication, printing, and material processing, typically resulting in costs ranging from hundreds of dollars depending on object size and complexity. Optical projection or shadow-based attacks require controlled lighting environments and projection equipment, introducing additional setup constraints and cost.

*5) Transferability:* We evaluate the transferability of the proposed attack. For black-box settings, we use PointNet trained as the model to generate patches for adversarial point clouds and evaluate the attack performance on other models.

The results on the transferability in the digital world are shown in Table. IV. We observe that black-box attacks on these three models match the performance of white-box attacks

Authorized licensed use limited to: Nanjing University. Downloaded on November 23,2026 at 09:57:54 UTC from IEEE Xplore. Restrictions apply.

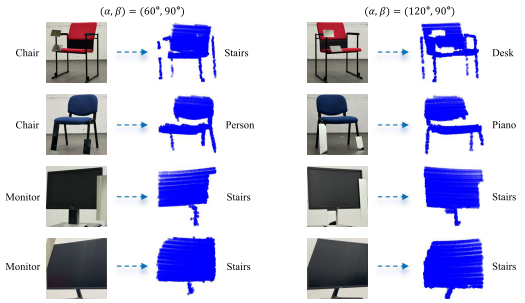



Fig. 8. Physical attack results. The angles are set to $(\alpha,\beta) = (60°, 90°)$ and $(\alpha,\beta) = (120°, 90°)$ respectively.

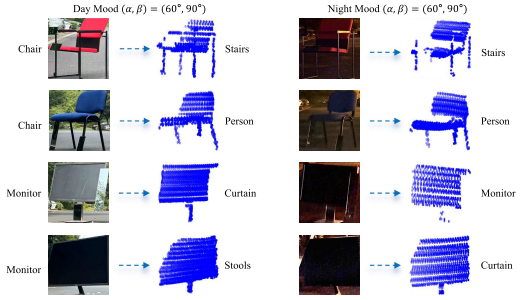



Fig. 9. Physical attack results. The angles are set to $(\alpha,\beta) = (60°, 90°)$. We collect data during the day and at night respectively.

in most cases. The performance gap increases as $n$ increases. We speculate the reason could be that for the first few rounds, different models may choose similar points to remove, and the differences become larger in the later rounds. Unlike model-specific additive perturbations, our attack removes coherent regions corresponding to discriminative geometric structures, such as edges, corners, and protruding parts. These structures are commonly used by different architectures for classification. Table IV shows the results in the physical world, which also demonstrate the good transferability of the proposed attack. Specifically, we evaluate the attack on three types of objects (5 of each), and we find that the attack achieves success rates ranging from 75% to 100%, 75% to 100%, and 60% to 80% on chairs, monitors, and plants, respectively, across three different models.

We also explore the methods with different angles, lighting conditions, and outdoor environment. Without changing the position of plane mirrors, we adjust the position of LiDAR or lighting conditions. Fig. 8 shows that $\alpha = 60°, \beta = 90°$ and $\alpha = 120°, \beta = 90°$ are set for some objects. Fig. 9 shows that $\alpha = 60°, \beta = 90°$ is set in outdoor environment, which are captured during the day and at night. This reflects the effective attack performance in different environments.

*6) Viewpoint Robustness:* To cover a wider range of LiDAR–patch angles, we add a digital sweep $\theta = \alpha_m - 90°$ from $-90°$ to $90°$ in $5°$ steps, where $\theta = 0°$ is normal incidence. Patches are selected once at $\theta = 0°$ with $n = 300$ and $k = 50$ and then kept fixed.

A ray that strikes a patch follows the law of reflection in Section III-B.3. A point from the $\theta = 0°$ connected region is removed if its reflected ray falls outside the receiver FOV,

$$\angle(\mathbf{r}, -\mathbf{d}) > \delta \quad \Longleftrightarrow \quad 2i > \delta,$$

where $i = |\theta|$ and $\delta = 15°$. We soften this cutoff by

$$\lambda(\theta) = \text{clip}\left(\frac{2|\theta| - 5°}{20°}, 0, 1\right)$$

and remove

$$N_{\text{del}}(\theta) = \text{round}\big(300 \cos\theta \cdot \lambda(\theta)\big)$$

points from that region. The factor $\cos\theta$ is the projected patch area.

The sweep uses 256 ModelNet40 samples and PointNet. As shown in Fig. 10, accuracy at $\theta = 0°$ equals the 88.67% clean baseline. It falls to 82.03%, 64.45%, and 53.13% at + 5°, + 10°, and + 15°, and recovers to 58.59%, 64.84%, 76.17%, and 87.89% at + 30°, + 45°, + 60°, and + 90°. The negative side is nearly symmetric (81.64%, 63.28%, and 51.56% at −5°, −10°, and −15°). The attack is strongest near the first angles at which the reflected beam leaves the receiver FOV, and it does not persist at grazing incidence. These results quantify geometric sensitivity to viewpoint change.

*7) Defense Evaluation:* We apply an eight-neighbor interpolation defense that restores every attacked input to 1,024 points. Table IX shows that the repair changes PointNet accuracy by at most 0.16 points, and that it raises DGCNN accuracy under Saliency-TopK from 59.08% to 63.29% while further lowering accuracy under our connected attack from 58.14% to 56.12%. These results show that our method remains robust against interpolation-based repair.

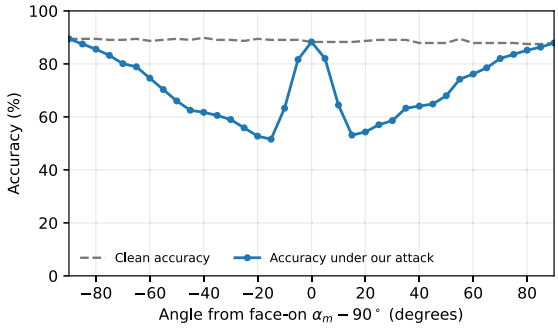


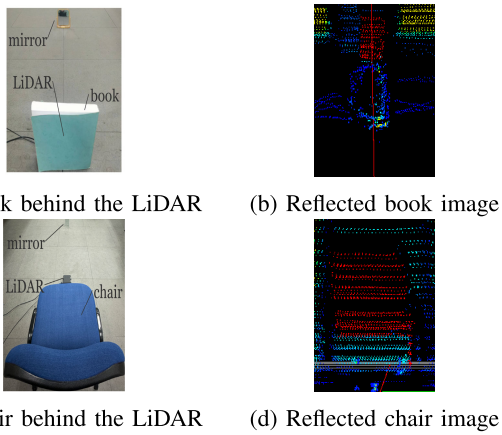

Fig. 10. Geometric angular-sweep simulation on PointNet.

TABLE IX

EIGHT-NEIGHBOR INTERPOLATION ON THE FULL 2,468-SAMPLE MODELNET40 TEST SET. THE INTERPOLATOR RESTORES EVERY ATTACKED INPUT TO 1,024 POINTS. Δ IS REPAIRED MINUS ATTACKED ACCURACY; A POSITIVE Δ MEANS THE DEFENSE RECOVERS ACCURACY

| Victim | Method | Acc. (%) | +Interp. (%) | Δ |
|---|---|---|---|---|
| PointNet | Saliency-TopK | 53.81 | 53.85 | +0.04 |
| | Our Attack | **52.47** | **52.51** | +0.04 |
| DGCNN | Saliency-TopK | 59.08 | 63.29 | +4.21 |
| | Our Attack | **58.14** | **56.12** | −2.02 |

Authorized licensed use limited to: Nanjing University. Downloaded on November 23,2026 at 09:57:54 UTC from IEEE Xplore. Restrictions apply.




(a) Book behind the LiDAR (b) Reflected book image

(c) Chair behind the LiDAR (d) Reflected chair image

Fig. 11. Place objects out of the LiDAR's field of view and capture their point cloud imaging with the help of mirror reflections.

## V. Discussion

### A. Adding Points by Mirrors

We have made an interesting observation: the device sensor is capable of detecting objects reflected by the mirror, even when the object is behind the LiDAR.

As shown in Fig. 11a, we have placed the mirror approximately 1.2 meters in front of the LiDAR device, with a book placed closely behind it. The image captured by the LiDAR device is shown in Fig. 11b. A rectangle of points appears directly in front, indicating a distance of nearly 2.5 meters. We mark the point cloud with red in the image. This distance precisely corresponds to the sum of the distance from the LiDAR to the mirror (~1.2 meters) and the distance from the mirror to the book (~1.3 meters). In addition, we have also placed a chair behind the LiDAR device, as shown in Fig. 11c. In Fig. 11d, the captured image can clearly displays the chair. The conclusion about distance is the same as what we observed in the experiment with the book.

These findings provide evidence that we can create an object within the LiDAR device's field of view using mirrors. Such a mechanism broadens the capability of manipulating the point cloud, allowing for both adding and removing points instead of solely removing points, when specular reflective patches are used to attack 3D LiDAR perception.

### B. Potential Defenses

Given the demonstrated vulnerability of LiDAR perception to 2D specular reflective patches, it is critical to consider possible countermeasures. We outline several potential defense directions:

- Sensor-Level Mitigation. LiDAR devices could integrate additional hardware-based mechanisms to identify abnormal specular reflections. For example, combining LiDAR with polarization filters or multiple laser wavelengths may help differentiate between diffuse and specular surfaces. In addition, incorporating complementary sensors such as cameras or radar can provide redundant information to cross-check suspicious missing regions in the point cloud.
- Learning-Based Robust Models. Recent studies provide several promising directions for improving point-cloud robustness. PWAVEP uses spectral graph wavelets to identify and purify adversarial perturbations [20], while PointDP employs diffusion-driven purification to reconstruct clean point clouds before classification [36]. CausalPC improves robustness by identifying the causal effects of structural information while reducing the influence of adversarial noise [12]. These methods demonstrate the potential of input purification and robust feature learning.
- Geometric Consistency Checks. LiDAR perception systems can exploit geometric priors of the environment. Since 2D reflective patches often lead to physically implausible point clouds (e.g., large missing surfaces or unnatural reflections), models can validate inputs against scene-level consistency. Multi-frame temporal aggregation is another promising approach: by comparing point clouds captured across different viewpoints or timestamps, the system can reduce the impact of a single manipulated frame.
- Adversarial Detection. Another line of defense is to design detectors that explicitly identify adversarial attacks. For instance, statistical analysis of reflection intensity, abnormal point density distributions, or sudden distance discontinuities can be employed to flag possible patch-based manipulations. Such detection can serve as a warning layer, triggering fallback strategies (e.g., switching to conservative decision modes in autonomous driving).

While our work focuses on revealing a novel attack vector, these defense directions highlight opportunities for strengthening LiDAR-based perception. We hope that our findings not only shed light on the security weaknesses but also motivate future research into developing robust multi-sensor fusion frameworks, adversarially robust training pipelines, and real-time detection methods against physical-world manipulations.

### C. Limitations and Future Work

Our work has paved the way to utilize 2D specular reflective patches for manipulating LiDAR 3D perception in object recognition. However, the current proposed approach has limitations in terms of applicability and attack capability. Regarding applicability, the proposed attack targets the recognition task specifically and may not be suitable for manipulating models with other tasks, such as detection, segmentation, or surface reconstruction. Exploring attacks against these tasks under the framework of using 2D specular reflective patches could be promising. As for attack capability, the approach in this work exploits the property that 2D specular reflective patches can remove some points from the point cloud to fabricate the attack. As mentioned in the conclusion of our paper and in Section V-A of this supplemental material, future research could investigate whether mirrors can be used to add points to the point cloud for attacking LiDAR 3D perception.

### D. Potential Negative Impacts

Our work investigates a practical physical attack on LiDAR 3D perception using 2D specular reflective patches. Since the

Authorized licensed use limited to: Nanjing University. Downloaded on November 23,2026 at 09:57:54 UTC from IEEE Xplore. Restrictions apply.

proposed attack can be realized with commonly used items, such as plane mirrors, misusing this attack may pose a threat to applications and systems relying on LiDAR 3D perception. Therefore, our goal in proposing this attack and revealing its potential harm is to demonstrate the current deficiencies in LiDAR 3D perception and to further motivate research into more robust LiDAR 3D perception systems in real-world applications.

## VI. Conclusion

In this paper, we propose an effective physical adversarial attack against LiDAR-based 3D object recognition by manipulating point clouds through point removal. Specifically, our attack exploits the vulnerability of LiDAR sensors on specular surfaces and makes the physical realization feasible by utilizing easily accessible and inexpensive 2D specular reflective patches, such as plane mirrors. Extensive experiments have been conducted on benchmark dataset in the digital world and on objects in the physical world to validate the effectiveness, stealthiness, and transferability of the attack. Our work provides an important step towards understanding the vulnerability of LiDAR-based object recognition and paves the way of utilizing low-dimensional items to manipulate high-dimensional perception. Future work would extend the ability of specular reflective patches to not only remove points but also create additional points in point clouds (please refer to supplementary materials for some exploratory results), and also investigate defenses against such physical adversarial attacks.

## References


[1] Y. Cao et al., "Adversarial sensor attack on LiDAR-based perception in autonomous driving," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2019, pp. 2267–2281.

[2] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 39–57.

[3] J. Chen, D. Liao, S. Xiang, and H. Zheng, "LiDAttack: Robust black-box attack on LiDAR-based object detection," 2024, *arXiv:2411.01889*.

[4] J. Chen, X. Li, M. Xu, C. Li, and T. Li, "Rethinking gradient-based adversarial attacks on point cloud classification," 2025, *arXiv:2505.21854*.

[5] L. Cheng et al., "Eidos revisited: Expanding efficient, imperceptible adversarial attacks on 3D point clouds," *J. Syst. Archit.*, vol. 169, Dec. 2025, Art. no. 103595.

[6] X. Deng, W. Y. Zhang, Q. Ding, and X. M. Zhang, "PointVector: A vector representation in point cloud analysis," 2023, *arXiv:2205.10528*.

[7] Y. Miao, Y. Dong, J. Zhu, and X.-S. Gao, "Isometric 3D adversarial examples in the physical world," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 19716–19731.

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015.

[9] A. Hamdi, S. Rojas, A. Thabet, and B. Ghanem, "AdvPC: Transferable adversarial perturbations on 3D point clouds," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 241–257.

[10] B. He et al., "Generating transferable 3D adversarial point cloud via random perturbation factorization," in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, no. 1, pp. 764–772.

[11] Q. Huang, X. Dong, D. Chen, H. Zhou, W. Zhang, and N. Yu, "Shape-invariant 3D adversarial point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 15335–15344.

[12] Y. Huang, M. Zhang, D. Ding, E. Jiang, Z. Wang, and M. Yang, "CausalPC: Improving the robustness of point cloud classification by causal effect identification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 19779–19789.

[13] S. Jia, X. Gong, F. Liu, and L. Ma, "AI-powered LiDAR point cloud understanding and processing: An updated survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 8, pp. 11249–11275, Aug. 2025.

[14] Z. Jin, X. Ji, Y. Cheng, B. Yang, C. Yan, and W. Xu, "PLA-LiDAR: Physical laser attacks against LiDAR-based 3D object detection in autonomous vehicle," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2023, pp. 1822–1839.

[15] J. Kim, B.-S. Hua, D. T. Nguyen, and S.-K. Yeung, "Minimal adversarial examples for deep learning on 3D point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 7797–7806.

[16] R. Kobayashi, K. Nomoto, Y. Tanaka, G. Tsuruoka, and T. Mori, "Invisible but detected: Physical adversarial shadow attack and defense on LiDAR object detection," in *Proc. 34th USENIX Secur. Symp. (USENIX Secur. 25)*, 2025, pp. 7369–7386.

[17] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," 2016, *arXiv:1611.01236*.

[18] Y. Lao, Y. Zhang, Z. Wang, C. Wang, Y. Xue, and W. Shao, "DisorientLiDAR: Physical attacks on LiDAR-based localization," 2025, *arXiv:2509.12595*.

[19] G. Li et al., "DeepGCNs: Making GCNs go as deep as CNNs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, pp. 6923–6939, Jun. 2023.

[20] H. Li, R. Liu, H. Liu, C. Wang, L. Yin, and J. Xu, "PWAVEP: Purifying imperceptible adversarial perturbations in 3D point clouds via spectral graph wavelets," in *Proc. ACM Web Conf.*, Apr. 2026, pp. 2893–2904.

[21] Y. Li, Y. Li, X. Dai, S. Guo, and B. Xiao, "Physical-world optical adversarial attacks on 3D face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 24699–24708.

[22] Y. Li, C. Wen, F. Juefei-Xu, and C. Feng, "Fooling LiDAR perception via adversarial trajectory perturbation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 7898–7907.

[23] D. Liu, R. Yu, and H. Su, "Adversarial shape perturbations on 3D point clouds," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 88–104.

[24] H. Liu, Y. Wu, Z. Yu, Y. Vorobeychik, and N. Zhang, "SlowLiDAR: Increasing the latency of LiDAR-based detection using adversarial examples," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 5146–5155.

[25] H. Liu, J. Jia, and N. Z. Gong, "PointGuard: Provably robust 3D point cloud classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6186–6195.

[26] X. Ma, C. Qin, H. You, H. Ran, and Y. Fu, "Rethinking network design and local geometry in point cloud: A simple residual MLP framework," 2022, *arXiv:2202.07123*.

[27] I. H. Mahir, D. A. Bachour, K. Abedrabboh, D. Perez-Astudillo, and L. Al Fagih, "Processing airborne LiDAR point cloud for solar cadasters: A review," *Appl. Energy*, vol. 382, Mar. 2025, Art. no. 125325.

[28] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 922–928.

[29] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.

[30] C. R. Qi, L. Yi, H. Su, and L. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.

[31] G. Qian, A. Hamdi, X. Zhang, and B. Ghanem, "Pix4Point: Image pretrained standard transformers for 3D point cloud understanding," 2024, *arXiv:2208.12259*.

[32] G. Qian, H. Hammoud, G. Li, A. Thabet, and B. Ghanem, "ASSANet: An anisotropic separable set abstraction for efficient point cloud representation learning," 2021, *arXiv:2110.10538*.

[33] G. Qian et al., "PointNeXt: Revisiting PointNet++ with improved training and scaling strategies," 2022, *arXiv:2206.04670*.

[34] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 945–953.

[35] J. Sun, Y. Cao, Q. A. Chen, and Z. M. Mao, "Towards robust LiDAR-based perception in autonomous driving: General black-box adversarial sensor attack and countermeasures," in *Proc. 29th USENIX Secur. Symp. (USENIX Secur.)*, 2020, pp. 877–894.


Authorized licensed use limited to: Nanjing University. Downloaded on November 23,2026 at 09:57:54 UTC from IEEE Xplore. Restrictions apply.

[36] J. Sun, J. Wang, W. Nie, Z. Yu, Z. Mao, and C. Xiao, "A critical revisit of adversarial robustness in 3D point cloud recognition with diffusion-driven purification," in *Proc. 40th Int. Conf. Mach. Learn.*, 2023, pp. 33100–33114.

[37] C. Szegedy et al., "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*.

[38] K. Tang et al., "Imperceptible adversarial attacks on point clouds guided by point-to-surface field," in *Proc. ICASSP - IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2025, pp. 1–5.

[39] J. Tu et al., "Physically realizable adversarial examples for LiDAR object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13716–13725.

[40] M. A. Uy, Q. Pham, B. Hua, T. Nguyen, and S. Yeung, "Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1588–1597.

[41] J. Wang, F. Li, S. Lv, L. He, and C. Shen, "Physically realizable adversarial creating attack against vision-based BEV space 3D object detection," *IEEE Trans. Image Process.*, vol. 34, pp. 538–551, 2025.

[42] M. Wang, R. Huang, W. Xie, Z. Ma, and S. Ma, "Compression approaches for LiDAR point clouds and beyond: A survey," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 21, no. 7, pp. 1–31, Jul. 2025.

[43] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, Oct. 2019.

[44] J. Wu et al., "Gradient-based sparse voxel attacks on point cloud object detection," *Pattern Recognit.*, vol. 160, Apr. 2025, Art. no. 111156.

[45] W. Wu, Z. Qi, and L. Fuxin, "PointConv: Deep convolutional networks on 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9621–9630.

[46] Z. Wu et al., "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.

[47] C. Xiang, C. R. Qi, and B. Li, "Generating 3D adversarial point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9136–9144.

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

[49] R. Yao, A. Zhang, Y. Zhou, J. Zhao, B. Liu, and A. E. Saddik, "Adversarial geometric attacks for 3D point cloud object tracking," *IEEE Trans. Multimedia*, vol. 27, pp. 3144–3157, 2025.

[50] K. You, T. Chen, D. Ding, M. S. Asif, and Z. Ma, "RENO: Real-time neural compression for 3D LiDAR point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2025, pp. 22172–22181.

[51] Y. Zhang, J. Hou, and Y. Yuan, "A comprehensive study of the robustness for LiDAR-based 3D object detectors against adversarial attacks," *Int. J. Comput. Vis.*, vol. 132, no. 5, pp. 1–33, May 2024.

[52] T. Zheng, C. Chen, J. Yuan, B. Li, and K. Ren, "PointCloud saliency maps," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1598–1606.

[53] X. Zhu, S. Huang, Z. Hu, J. Li, J. Zhu, and X. Hu, "Physical adversarial examples for person detectors in thermal images based on 3D modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 10, pp. 8973–8989, Oct. 2025.

[54] Y. Zhu, C. Miao, T. Zheng, F. Hajiaghajani, L. Su, and C. Qiao, "Can we use arbitrary objects to attack LiDAR perception in autonomous driving?" in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2021, pp. 1945–1960.

Authorized licensed use limited to: Nanjing University. Downloaded on November 23,2026 at 09:57:54 UTC from IEEE Xplore. Restrictions apply.