

Revisiting Physically Realizable Adversarial Object Attack against LiDAR-based Detection: Clarifying Problem Formulation and Experimental Protocols

Luo Cheng¹², Hanwei Zhang³⁴, Lijun Zhang², Holger Hermanns³

¹University of Chinese Academy of Sciences, Beijing, China

²Key Laboratory of System Software (Chinese Academy of Sciences) and State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China

³Universität des Saarlandes, Saarland, Germany

⁴Institute of Intelligent Software, Guangzhou, China

Abstract

Adversarial robustness in LiDAR-based 3D object detection is a critical research area due to its widespread application in real-world scenarios. While many digital attacks manipulate point clouds or meshes, they often lack physical realizability, limiting their practical impact. Physical adversarial object attacks remain underexplored and suffer from poor reproducibility due to inconsistent setups and hardware differences. To address this, we propose a device-agnostic, standardized framework that abstracts key elements of physical adversarial object attacks, supports diverse methods, and provides open-source code with benchmarking protocols in simulation and real-world settings. Our framework enables fair comparison, accelerates research, and is validated by successfully transferring simulated attacks to a physical LiDAR system. Beyond the framework, we offer insights into factors influencing attack success and advance understanding of adversarial robustness in real-world LiDAR perception.

Introduction

Adversarial phenomena (Kurakin, Goodfellow, and Bengio 2018; Zhang et al. 2022) pose a significant threat to deep learning models. In particular, ensuring adversarial robustness in LiDAR-based 3D object detection is crucial due to its widespread use in real-world applications across various industries, including autonomous vehicles, delivery robots, warehouse automation, and drones (Aung et al. 2024). While adversarial attacks against 3D point cloud models by inserting, deleting, or shifting points (Huang et al. 2022; Wen et al. 2019; Zhang et al. 2024) or modifying the reconstructed meshes (Zhang et al. 2021) from 3D point cloud, have advanced our understanding of model vulnerabilities, many of these methods are not physically realizable. In practice, such manipulations are often impractical or easily overlooked. While idealized studies of adversarial robustness help identify model flaws, evaluating physical attacks is crucial for assessing real-world robustness and ensuring AI systems meet safety standards like those in the EU AI Act¹.

Despite growing interest, physical attacks on LiDAR-based 3D object detection remain underexplored compared

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

¹Regulation (EU) 2024/1689 on Artificial Intelligence (AI Act), <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ:L-202401689>



Figure 1: Top row shows the simulated scenario in CarLA; bottom row shows the real-world setup with a small car and a 3D-printed adversarial object. Left column presents the camera view of the scene, including the car with the adversarial object and the LiDAR-equipped car. Right column shows the corresponding LiDAR point cloud perception by the LiDAR-equipped car.

to digital attacks (Zhang, Hou, and Yuan 2024). Existing physical attacks fall into two types: spoofing attacks, which inject fake points into LiDAR streams via lasers or reflectors (Cao et al. 2019b; Sun et al. 2020), and adversarial objects, *i.e.* crafted physical artifacts to mislead perception systems (Tu et al. 2020; Abdelfattah et al. 2021). We view adversarial object attacks as a serious threat, but research is limited due to poor reproducibility. Few studies share code or setups, and hardware differences hinder fair comparison. To address this, we propose a device-agnostic, standardized framework to enable reproducible benchmarking and accelerate research progress.

Our standardized, device-agnostic framework abstracts the key components of physically realizable adversarial object attacks. By adjusting elements like mesh shape, loss constraints, and object placement, our framework can encompass existing attack methods. To enable fair comparison, we provide open-source code with evaluation protocols in both simulation and dataset settings. Figure 1 validates that

our simulation setting closely approximates the real world. We 3D-print a simulated adversarial object and demonstrate its real-world effectiveness on a physical LiDAR-based detection system. Our framework supports various loss functions, optimization strategies, and attack scenarios (white/black-box), and is compatible with multiple detection models. In this work, we follow a widely adopted setup, using it as a case study to analyze key factors affecting attack success and to provide foundational insights into physical adversarial objects. Thus, our contributions go beyond introducing a framework with benchmarkable settings, evaluation protocols, and open-source code to advance research on physical adversarial objects. We also provide insights and preliminary knowledge on real-world LiDAR robustness.

Related Work

Robustness of LiDAR-based 3D Object Detection

The robustness of LiDAR-based 3D object detection typically includes stability under varying conditions (Song et al. 2024) and adversarial robustness, *i.e.*, resilience against deliberate attacks (Zhang, Hou, and Yuan 2024), with recent focus on the latter. Zhang, Hou, and Yuan (2024) benchmark detectors by perturbing 3D points and categorize them as voxel-based (Yan, Mao, and Li 2018; Lang et al. 2019; Deng et al. 2021), point-based (Shi, Wang, and Li 2019), or mixed-feature (Shi et al. 2020), as well as one-stage (Yan, Mao, and Li 2018; Lang et al. 2019) or two-stage (Shi et al. 2020; Shi, Wang, and Li 2019) architectures. We follow this categorization but evaluate robustness against physically realizable attacks by introducing adversarial objects into the scene, better reflecting real-world threats in critical applications like autonomous driving.

Adversarial Attacks in Physical World

Physical-world adversarial attacks have received increasing attention due to their real-world safety implications (Ren, Huang, and Yan 2021), spanning image classification (Li, Schmidt, and Kolter 2019), pedestrian detection (Thys, Van Ranst, and Goedemé 2019), infrared imaging (Zhou et al. 2018), and LiDAR-based perception (Tu et al. 2020; Cao et al. 2019b). Among these, attacks on LiDAR systems are particularly important given their widespread use in autonomous driving.

Attacks on LiDAR systems can be broadly categorized into two types: spoofing attacks and adversarial objects. Spoofing attacks (Cao et al. 2019a) target the LiDAR sensor by injecting fake points to hinder detection, simulate fake vehicles, or disrupt vehicle networks and communication (Ju et al. 2022). However, their effectiveness is often limited in real-world scenarios due to environmental factors like weather, sensor noise, and difficulties in reproducing results outside controlled conditions. We consider adversarial objects to pose a greater threat in real-world applications.

Adversarial Object. PhyAdv (Tu et al. 2020) showed that placing an adversarial object on a vehicle’s rooftop can effectively hide it from LiDAR detectors. Similarly, Abdelfattah et al. (2021) extended adversarial techniques from 2D

to 3D detection by using RGB differentiable rendering for adversarial object generation. Inspired by object placement, (Zhu et al. 2021) proposed an attack framework that places reflective objects in physical space to mislead LiDAR sensing systems. Wang et al. (2023) developed an adversarial obstacle generation algorithm to deceive deep 3D detection models by creating fake obstacles within LiDAR’s detection range using only a few points. These studies demonstrate the feasibility and severity of adversarial object threats in realistic settings. However, inconsistent techniques, varied setups, and limited model coverage hinder the development of general insights into adversarial robustness in LiDAR-based perception. To address this, we build on prior work with a generalizable framework and comprehensive evaluation.

Physically Realizable Adversarial Object

Problem Formulation: Adversarial Object. Let \mathbf{P} denote an unordered set of 3D points and f denote a LiDAR-based 3D object detection which maps a point cloud \mathbf{P} to a set of 3D object bounding boxes \mathbf{B} associated with the corresponding probability scores $vs.$ of the predicted class labels y . Let $\mathbf{M} := \{\mathbf{v}_i\}$ denote a mesh object constructed by the vertices \mathbf{v}_i . Let $G(\mathbf{P}, \mathbf{M}, \mathbf{t})$ denote a function that integrates a mesh \mathbf{M} into a point cloud \mathbf{P} at given position and orientation \mathbf{t} , producing the resulting point clouds \mathbf{P}' . Möller-Trumbore intersection algorithm (Möller and Trumbore 2005) is a solution for G , which computes ray-triangle intersections between LiDAR rays and the mesh faces.

We aim to find an adversarial mesh \mathbf{M}_{adv} against the target LiDAR-based 3D object detection f over the 3D points \mathbf{P} from the distribution \mathcal{D} , satisfying

$$\arg \min_{\mathbf{M}_{adv}} \sum_{\mathbf{P} \sim \mathcal{D}} \mathcal{L}(f, G(\mathbf{P}, \mathbf{M}_{adv}, \mathbf{t})), \quad (1)$$

$$s.t. \quad \Phi(\mathbf{M}_{adv}) = True \quad (2)$$

where $\mathcal{L}(f, G(\mathbf{P}, \mathbf{M}_{adv}, \mathbf{t}))$ is the misdetection loss and $\Phi(\mathbf{M})$ is a boolean function that determines if a mesh \mathbf{M} is physically feasible.

Positioning of Object. The placement of the object, *i.e.* \mathbf{t} , depends on the overall design and physical context. Several studies place the adversarial object on top of the car, as this position ensures greater visibility across diverse scenarios and poses a more significant threat.

Physically Feasible Constraint. Implementing directly the boolean function Φ to check the physical feasibility of a mesh is challenging. To enforce this constraint during optimization, two main strategies are used: (1) reparameterizing the adversarial mesh to implicitly include the constraint; (2) incorporating a measure of physical feasibility into the loss function.

An example of parameterization is from PhyAdv (Tu et al. 2020), which reformulates the problem using learnable local displacement vectors $\Delta \mathbf{v}_i \in \mathbb{R}^3$ for each vertex. To enforce a box constraint and ensure physical feasibility with given size and translation limits, we denote the reparameterization as

$$\mathbf{v}_i = \tau(\mathbf{v}_i^0, \Delta \mathbf{v}_i), \quad (3)$$

where \mathbf{v}_i^0 is the initial vertex. On the other hand, surface smoothness can serve as a proxy for measuring physical feasibility of a mesh. Accordingly, the Laplacian loss for estimating surface smoothness is defined as:

$$\phi(\mathbf{M}_{adv}) := \sum_{\mathbf{v}_i \in V} \left\| \mathbf{v}_i - \frac{1}{\|N(i)\|} \sum_{j \in N(i)} \mathbf{v}_j \right\|_2^2, \quad (4)$$

where V is the set of vertices in \mathbf{M}_{adv} and $N(i)$ denotes the neighboring vertices of \mathbf{v}_i .

Adversarial Misdetection Loss. The misdetection loss (1) is critical to the effectiveness of adversarial objects and should be tailored to the threat objective and model architecture. Existing studies on physical adversarial objects often overlook this aspect, typically defaulting to loss functions tied to the initial stage of two-stage detectors. Since detection involves both localization and classification, their individual contributions to adversarial effectiveness remain underexplored. To address this, we examine different loss formulations and group them into three categories: mislocalization (ML) loss, misrecognition (MR) loss, and comprehensive misdetection (C) loss, aiming to enhance attack performance across architectures.

Mislocalization Loss (ML): Misdetection occurs when the detection model fails to correctly locate the bounding boxes, regardless of the predicted class. The misdetection loss can be formulated as

$$\mathcal{L}(f, \mathcal{D}, \mathbf{M}_{adv}) := \sum_{\mathbf{P} \sim \mathcal{D}} \sum_k IoU(\mathbf{B}, B_{gt}), \quad (5)$$

where \mathbf{B} is the proposed bounding boxes and B_{gt} denotes the given ground truth. A proposal is considered relevant if both its confidence score and its IoU with ground truth bounding box are greater than 0.1. This can be further extended by weighting each bounding box using its fixed prediction score.

Misrecognition Loss (MR): Misrecognition is also considered misdetection, meaning the object is correctly localized but not correctly recognized. A straightforward definition is

$$\mathcal{L}(f, \mathcal{D}, \mathbf{M}_{adv}) := \sum_{\mathbf{P} \sim \mathcal{D}} \sum_k -\log(1 - \sigma(s)). \quad (6)$$

where s is the prediction logit of the target class, $\sigma(\cdot)$ is the sigmoid function and $\sigma(s)$ is prediction score from the network. A weighted version is used in PhyAdv (Tu et al. 2020):

$$\mathcal{L}(f, \mathcal{D}, \mathbf{M}_{adv}) := \sum_{\mathbf{P} \sim \mathcal{D}} \sum_k -IoU(\mathbf{B}, B_{gt}) \log(1 - \sigma(s)), \quad (7)$$

where the predicted bounding box and its corresponding IoU value are kept fixed as weights for the confidence scores.

Comprehensive Misdetection Loss (C): To encompass both mislocalization and misrecognition in misdetection, we use (7) without fixing either the proposed bounding boxes or the prediction scores. However, it is ineffective in the second stage of a two-stage model when the prediction score is nearly one. Thus, for such a case, we propose to construct the loss as

$$\mathcal{L}(f, \mathcal{D}, \mathbf{M}_{adv}) := \sum_{\mathbf{P} \sim \mathcal{D}} \sum_k IoU(\mathbf{B}, B_{gt}) \cdot \sigma(s). \quad (8)$$

Logit vs. Score. Logit loss is simple yet effective for transferable adversarial attacks in images (Zhao, Liu, and Larson 2021). However, in 3D detection attacks, only the score is used as part of the loss. To fill the gap, we modify (8) as

$$\mathcal{L}(f, \mathcal{D}, \mathbf{M}_{adv}) := \sum_{\mathbf{P} \sim \mathcal{D}} \sum_k IoU(\mathbf{B}, B_{gt}) \cdot s, \quad (9)$$

and compare the performance between using the score and the logit of object prediction.

A Study Case. Our framework is designed to flexibly support different initialization shapes and placements of adversarial objects, enabling it to encompass a wide range of existing work. While variations in shape, position, and constraints can significantly alter the physical representation of an attack, we emphasize that the core challenge lies in the algorithmic components—specifically, how to formulate effective physically feasible constraints, define misdetection losses, and solve the optimization problem efficiently. These components remain relevant regardless of the object’s initial shape or placement. Therefore, for clarity and consistency, we adopt a standard setup: a mesh initialized as a sphere, positioned on top of a car.

To generate a physically realizable adversarial object \mathbf{M}_{adv} , we formulate an optimization problem,

$$\ell(f, \mathcal{D}, \mathbf{M}_{adv}) := \mathcal{L}(f, \mathcal{D}, \mathbf{M}_{adv}) + \lambda \phi(\mathbf{M}_{adv}), \quad (10)$$

that balances the misdetection loss and the physical feasibility constraints by a weighting parameter λ .

We generalize the attack algorithm as shown in Algorithm 1. We use Gradient Descent (GD) with fixed step size ϵ as the optimizer. The simple yet efficient framework enables attacks in both white-box and black-box settings and allows the exploration of various definitions of misdetection loss \mathcal{L} .

In line 1, we initialize the mesh as a isotropic sphere with parameters sphere level ν and scale b . The initial learnable displacement vector $\Delta \mathbf{v}_i$ is set to zero. In line 5 and 9, we estimate the adversarial loss of LiDAR-based 3D detection f on 3D points \mathbf{P} with adversarial mesh \mathbf{M}_{adv} , then calculate the gradient with respect to the learnable vector $\Delta \mathbf{v}_i$. In the black-box setting, f is the surrogate 3D detector, whereas in the white-box setting, g serves as the sole target detector. Gradient descent is applied on line 6 and 11. In a white-box setting, we update the adversarial mesh at each step. However, in a black-box setting, we update the adversarial mesh only when the misdetection loss on the targeted detection model g decreases.

Evaluation Protocol

Attack success rate and invisibility are crucial in attacks. For LiDAR-based 3D object detection, precision is measured via 3D bounding boxes or BEV, though not all evaluations consider both. Assessing adversarial object invisibility is also challenging. We propose a standardized evaluation protocol to thoroughly assess adversarial attack performance and detector robustness.

Algorithm 1: Adversarial Attack against LiDAR-based Detector

Input: g : Targeted LiDAR-based 3D Detector, \mathbf{D} : Given dataset
Input: f : (Optional) Surrogate LiDAR 3D Detector
Input: ν : Sphere level, \mathbf{b} : Scale, ϵ : Step size
Output: $\mathbf{M}_{adv} := \{\tau(\mathbf{v}_i^0, \Delta\mathbf{v}_i)\}$: An adversarial object defined by attacked vertices.

```

1:  $\mathbf{v}_i^0 \leftarrow Sphere(\nu, \mathbf{b})$ ,  $\Delta\mathbf{v}_i \leftarrow \mathbf{0}$ 
2:  $\mathbf{M}_{adv} \leftarrow \{\tau(\mathbf{v}_i^0, \Delta\mathbf{v}_i)\}$  w.r.t. (3)
3: while  $\mathbf{P} \in \mathbf{D}$  do
4:   if the attack is under white-box setting then
5:      $\mathbf{g} \leftarrow \nabla_{\Delta\mathbf{v}_i} \ell(g, \mathbf{P}, \mathbf{M}_{adv})$  w.r.t. (10)
6:      $\Delta\mathbf{v}_i \leftarrow \Delta\mathbf{v}_i - \epsilon \frac{\mathbf{g}}{\|\mathbf{g}\|_2}$ 
7:      $\mathbf{M}_{adv} \leftarrow \{\tau(\mathbf{v}_i^0, \Delta\mathbf{v}_i)\}$  w.r.t. (3)
8:   else if the attack is under black-box setting then
9:      $\mathbf{g} \leftarrow \nabla_{\Delta\mathbf{v}_i} \ell(f, \mathbf{P}, \mathbf{M}_{adv})$  w.r.t. (10)
10:    if  $\mathcal{L}(g, \mathbf{P}, \{\tau(\mathbf{v}_i^0, \Delta\mathbf{v}_i)\}) < \mathcal{L}(g, \mathbf{P}, \mathbf{M}_{adv})$  then
11:       $\Delta\mathbf{v}_i \leftarrow \Delta\mathbf{v}_i - \epsilon \frac{\mathbf{g}}{\|\mathbf{g}\|_2}$ 
12:       $\mathbf{M}_{adv} \leftarrow \{\tau(\mathbf{v}_i^0, \Delta\mathbf{v}_i)\}$  w.r.t. (3)
13:    end if
14:  end if
15: end while

```

Attack Success Rate. We consider two precisions: 3D mean average precision (3D mAP) and Bird’s Eye View mean average precision (BEV mAP), which projects 3D bounding boxes onto a 2D plane. Both are measured with moderate difficulty with an IoU threshold of 0.7. To calculate the attack success rate (ASR), we subtract the accuracy with the adversarial mesh (p_a) from the original precisions (p_o) and divide by the original precisions, i.e. $ASR := \frac{p_o - p_a}{p_o}$.

Invisibility. When evaluating the invisibility of an adversarial object, we consider three key aspects: i) *modification magnitude*, measured by the L_2 norm of changes to mesh vertices comparing to initialization; ii) *smoothness*, measured by the Laplacian smoothness (ϕ) of the vertices, reflecting physical realism; iii) *size of object*, measure by two metrics, i.e. *Area* and *Volume*. *Area* is the BEV area of the adversarial mesh, calculated as the union of triangle areas projected onto the BEV view, formalized as

$$Area(\mathbf{M}_{adv}) := \bigcup_{\{\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k\} \subset \mathbf{M}_{adv}} Proj_{xy}(\{\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k\}), \quad (11)$$

where $Proj_{xy}$ calculates the area of a triangle projected onto the BEV view using only the x and y axes. *Volume* is the total volume of the mesh, computed using the signed volume of tetrahedron algorithm, formalized as

$$Volume(\mathbf{M}_{adv}) := \left| \sum_{\substack{\{\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k\} \text{ is} \\ \text{a face in } \mathbf{M}_{adv}}} \frac{1}{6} (\mathbf{v}_i \cdot (\mathbf{v}_j \times \mathbf{v}_k)) \right|, \quad (12)$$

i.e. taking the absolute value of the sum of the signed volumes for all mesh tetrahedra defined by triangular faces. The first two aspects are standard measures of invisibility, while size is specifically tailored to adversarial objects.

Physical Realizable Setting. We structure our experimental setup across three levels: dataset-based evaluation, simula-

tion, and limited real-world testing. The core of our experimental protocol focuses on the dataset and simulation levels, which provide controlled, reproducible, and scalable environments for benchmarking. To assess real-world applicability, we conduct a small-scale physical experiment demonstrating the transferability of adversarial objects from simulation to reality, as shown in Figure 1. However, due to the complexity, cost, and lack of standardization in physical setups, we do not include real-world testing as part of our formal evaluation protocol.

Our standard procedure follows a train-test split: adversarial objects are generated using the training set and evaluated on the test set. While this structure is widely adopted, we emphasize that assessing physical realizability requires conditions that closely approximate the real world. Since conducting physical tests with actual vehicles is logically demanding and difficult to reproduce, we adopt a simulation-based approach as a practical surrogate. In our setting, the attacker generates adversarial objects using a dataset aligned in format with the simulation data but drawn from different samples. These objects are then evaluated within a simulation environment, allowing for consistent, scalable, and physically meaningful testing without the constraints of real-world deployment.

Experimental Settings

Dataset. We generate and test adversarial meshes using the KITTI (Geiger, Lenz, and Urtasun 2012) dataset, which provides 3D point clouds from a 64-line Velodyne HDL-64E LiDAR and object labels from the front camera of autonomous vehicles. Focusing on the “Car” class, we consider objects with over ten points per scene. Out of 28,742 labeled cars in our training and validation datasets, 25,844 meet this criterion. We evaluate the transferability of adversarial meshes using the KITTI-CARLA (CarLA) (Deschaud 2021) dataset. Since it only provides semantic labels, we first use the victim model to generate 3D bounding boxes from clean data. Ground truth boxes are those that overlap with “Car” points by more than 70%.

Networks. We select PointRCNN (PR) (Shi, Wang, and Li 2019), PointPillar (PP) (Lang et al. 2019), PV-RCNN (PVR) (Shi et al. 2020), Voxel RCNN (VR) (Deng et al. 2021), and SECOND (SC) (Yan, Mao, and Li 2018) as representative models for LiDAR-based 3D object detectors². The original BEV mAP on KITTI is 86.76% for PP, 85.67% for PR, 88.92% for PVR, 89.15% for VR, and 88.72% for SC, while the corresponding 3D mAP is 77.57% for PP, 78.67% for PR, 84.37% for PVR, 85.66% for VR, and 81.82% for SC.

Attacks. We use PhyAdv (Tu et al. 2020) as our baseline white-box attack with its original parameters: a maximum global offset of 0.1m in the x and y directions, no offset in the z direction ($\mathbf{c} = (0.1m, 0.1m, 0.0m)$), a Laplacian

²Codes and weights pre-trained on KITTI are from OpenPCDet(Team 2020). (<https://github.com/open-mmlab/OpenPCDet>)

loss weight $\lambda = 0.001$, and the Adam optimizer with a learning rate of $\gamma = 0.005$. The 3D adversarial mesh is initialized as a unit sphere with 162 vertices and 320 triangular faces, scaled by $\mathbf{b} = (0.7m, 0.7m, 0.5m)$. By default, our attack uses gradient descent with misdetection loss (9), a sphere level $\nu = 2$, and a maximum global offset $\mathbf{c} = (0.1m, 0.1m, 0.0m)$. The sphere level $\nu = 0, 1, 2, 3$ corresponds to vertex counts $\|V\| = 12, 42, 162, 642$. Step sizes are: $\epsilon = 0.005$ for PP, $\epsilon = 0.05$ for PR, and $\epsilon = 0.0005$ for PVR, VR, and SC.

Rendering. For LiDAR simulation, we use the KITTI setup with a Velodyne HDL-64E sensor and a vertical resolution of about 0.4 degrees, covering elevation angles from $+2^\circ$ to -24.8° . To focus on adversarial meshes above car rooftops, we keep only the top 10 lines, from $+2^\circ$ to approximately -2.19° . We use the algorithm from (Engelmann, Stückler, and Leibe 2017) for vehicle reconstruction. Since the code is not available, we train a TSDF using car models from the ApolloScape 3D dataset (Song et al. 2019). We then designate the top 0.2 meters of the reconstructed surface as the car’s rooftop and place the adversarial patch at its center.

Ablation for Key factors

We examine the impact of key factors, *i.e.*, the optimizer, initial mesh, and misdetection loss. For the ablation of initial mesh and misdetection loss, we selected PP, PR, and VR. This choice allows us to compare the differences between one-stage and two-stage networks, as well as point-based and voxel-based networks.

Influence of the Optimizer. To compare gradient descent and Adam optimizers in the white-box setting, we varied the step size ϵ for gradient descent and the learning rate γ for Adam across five 3D object detection models. For two-stage models, the loss was constructed in the second stage. The results, shown in Figure 2, indicate that Gradient Descent (GD) performs as well as or better than Adam. Consequently, GD is used for the remaining experiments. We also find that PR benefits from a larger step size, SC and VR perform better with a smaller step size, while PP has an optimal step size, and PVR shows little sensitivity to step size variations. In experiments, we use the optimal parameters for each model.

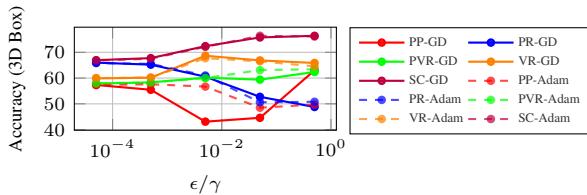


Figure 2: Ablation on step size ϵ for gradient descend or learning rate γ for Adam.

Influence of the Initial Mesh. In this section, we assess the impact of the initial mesh on attack performance by varying two parameters: the sphere level ν , which affects the

number of vertices, and the scale \mathbf{b} , which sets the mesh radius. We evaluate two scenarios: placing the sphere mesh on top of the car to test its effect on detection and optimizing the mesh to create an adversarial one. For two-stage models, we use (9) as the misdetection loss in \mathcal{L} . As shown in Figure 3, PP is the most sensitive to mesh scale, while PR is the least. The impact of the sphere level varies depending on the model during the attack. Our findings indicate that a larger initial mesh improves attack performance, but for physical reliability, we set $b = 0.7$. The optimal choice for ν is $\nu = 2$.

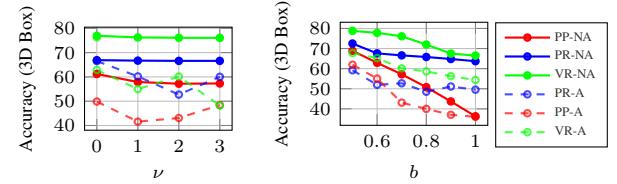


Figure 3: Ablation on sphere level ν and scale b . In the left figure, $\mathbf{b} = (0.7m, 0.7m, 0.5m)$; in the right figure, $\mathbf{b} = (bm, bm, 0.5m)$ with $\nu = 3$.

Influence of the Misdetection Loss. We examine various loss functions using GD with $\epsilon = 0.05$ on PR in a white-box setting, focusing on changes to \mathcal{L} in the loss function. As shown in Table 1, BEV mAP is more difficult to attack compared to 3D mAP, with an ASR gap of over $\sim 15\%$ for PP and VR, and about $\sim 10\%$ for PR. VR remains stable despite variations in misdetection loss. Generally, MR loss performs better than ML loss. There is no clear optimal choice, but MR losses from (7), (8) and (9) are strong options. Misdetection loss is usually more effective when constructed in Stage 1, though exceptions exist.

Loss	PP (2019)		PR (2019)		VR (2021)				
			Stage 1 (S1)		Stage 2 (S2)		Stage 1 (S1)		
	BEV	3D Box	BEV	3D Box	BEV	3D Box	BEV	3D Box	
ML	(5)	15.60	42.92	23.02	29.56	15.01	23.59	15.07	29.84
	(7)	16.26	44.32	21.56	30.07	21.16	29.83	15.19	30.17
	(8)	16.38	43.81	24.44	32.32	22.75	29.75	15.13	29.95
MR	(6)	<u>24.96</u>	<u>45.43</u>	24.29	34.61	24.45	34.00	15.14	30.09
	(7)	25.50	46.50	24.37	34.20	24.43	34.15	15.22	30.17
	(8)	23.25	43.95	27.14	38.36	23.89	<u>35.22</u>	15.19	30.07
	(9)	23.58	44.26	23.80	32.21	26.74	37.30	15.15	<u>30.16</u>
C	(7)	17.68	43.69	25.27	36.34	<u>25.56</u>	34.94	<u>15.20</u>	30.13
	(8)	16.91	43.10	25.73	<u>37.62</u>	25.00	34.77	15.16	29.95

Table 1: Comparison of misdetection loss designs. Bold indicates the best, and underlined denotes the second best.

Experimental Comparison

White-Box Setting. In this section, we demonstrate the performance of adversarial meshes generated by our framework in white-box settings and their transferability across

various models and datasets. We focus on two representative attacks using MR (8) and (9) since adversarial loss with logit shows better transferability in images (Zhao, Liu, and Larson 2021). We investigate if this holds for 3D objects.

Dataset	Method	PP (2019)		PR (2019)		PVR (2020)		VR (2021)		SC (2018)	
		BEV 3D Box	3D Box								
KITTI	Vanilla	7.09	26.29	10.16	15.34	6.53	21.50	1.18	11.16	1.31	9.62
	PhyAdv	12.20	36.25	15.50	22.35	2.34	6.99	13.11	21.47	3.34	12.66
	MR(8)	23.25	43.95	23.89	35.22	13.99	31.98	15.03	29.79	7.77	17.45
	MR(9)	23.58	44.26	26.74	37.30	8.60	30.69	15.04	29.83	7.77	17.15
CarLA	Vanilla	53.55	57.37	33.52	48.28	37.92	41.21	29.06	32.01	43.87	46.98
	PhyAdv	60.23	66.58	35.81	50.83	25.38	27.88	27.25	29.57	38.82	41.25
	MR(8)	67.85	72.35	41.81	56.31	35.54	38.66	29.02	32.02	43.00	46.09
	MR(9)	68.50	72.80	46.02	61.90	37.56	40.54	28.95	31.86	42.94	45.97

Table 2: ASR of adversarial meshes for different models in a white-box setting on the KITTI dataset, with transferability tested on CarLA. Bold indicates the best performance.

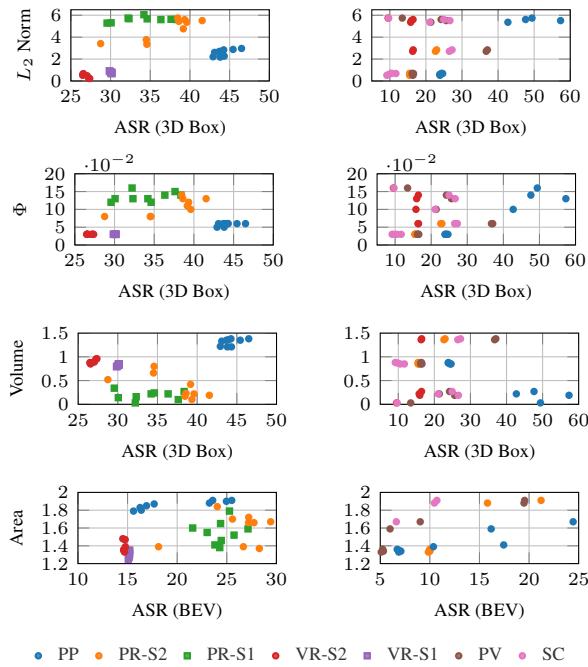


Figure 4: 3D and BEV box ASR plotted with metrics: Volume, L_2 norm, Laplacian smoothness (Φ), and projected area. Rows share y-axes. Left: white-box ASR (Table 1); Right: transferability ASR (Table 3). Colors indicate target models (used for both generation and testing in white-box; testing only in transferability).

The results for the KITTI dataset in Table 2 demonstrate that our attack significantly outperforms PhyAdv, especially on PP, PR, and PVR. Notably, on PVR, PhyAdv performs worse than Vanilla, which uses a spherical mesh placed on cars. For the CarLA dataset, Table 2 shows that our adversarial mesh transfers well across datasets, with models being highly affected by objects on top of cars. Vanilla achieves the

Source	S _{target}	Loss	PP (2019)		PR (2019)		PVR (2020)		VR (2021)		SC (2018)	
			BEV 3D Box	3D Box	BEV 3D Box	3D Box	BEV 3D Box	3D Box	BEV 3D Box	3D Box	BEV 3D Box	3D Box
PP	-	MR(8)	-	-	21.21	22.97	19.55	37.04	3.30	16.48	10.71	27.25
	-	MR(9)	-	-	15.81	22.65	19.49	36.70	3.19	16.34	10.46	26.49
	S1	MR(8)	17.44	49.40	-	-	3.14	13.45	0.89	9.53	1.66	9.68
	S1	MR(9)	16.18	47.62	-	-	5.99	24.21	1.52	16.44	3.92	24.94
PR	S1	MR(8)	24.44	57.32	-	-	9.03	25.63	1.75	15.97	6.62	26.66
	S1	MR(9)	10.36	42.72	-	-	4.23	21.37	0.84	15.78	1.93	21.06
	S2	MR(8)	7.08	24.64	10.01	15.60	5.32	16.56	-	-	1.79	11.69
	S2	MR(9)	6.87	24.34	9.85	15.51	5.10	16.23	-	-	1.68	10.55
VR	S2	MR(8)	6.88	23.84	9.96	15.72	5.26	16.24	-	-	1.65	9.79
	S2	MR(9)	6.71	23.90	9.98	15.49	5.25	16.43	-	-	1.65	9.18

Table 3: Transferability of adversarial meshes: generated against source models (columns) and tested on target models (rows). Bold indicates the best, and underlined denotes the second best.

highest ASR on PVR and SC, while our attack outperforms PhyAdv, particularly against PP and PR.

Transferability. According to Table 3, adversarial examples generated using PP show the highest transferability. Generally, BEV mAP is harder to compromise than 3D mAP, with VR being the most robust and PP the most vulnerable. Unlike image classification, using logits does not improve transferability for LiDAR-based detectors. Using MR (8) as the misdetection loss improves transferability when attacking the second stage of PR, but not for VR. Adversarial examples generated with VR show the lowest transferability, indicating that models easier to attack produce more transferable adversarial examples.

To explore how invisibility affects white-box attacks and transferability, we measure four invisibility metrics. In Figure 4, the first column of, derived from Table 1, shows no clear relationship between these metrics and ASR. However, the second column, from Table 3, reveals that while L_2 norm, Laplacian smoothness ϕ and 3D box volume have an inconsistent impact on transferable 3D ASR, a larger BEV area consistently improves transferable BEV ASR. Therefore, to enhance transferability, increasing the BEV area of the adversarial mesh during optimization is beneficial.

Black-Box Setting. According to Table 4, we present the results of using PP, PR, PVR, and VR as source models (f) and assessing their performance against these models, as well as SC, as target models (g) in a black-box setting. The results show that PP is the most effective source model. Conversely, PVR and VR as source models perform poorly against SC as the target model, while VR is the most difficult target model when PP and PR are used as sources.

CarLA Simulation Scenarios. We generate adversarial objects from 3D point clouds in the KITTI dataset, convert them into mesh models, and deploy them in the CarLA simulator (see Figure 5). Their impact is evaluated across pre-defined driving scenarios using the ScenarioRunner frame-

Surrogate	Stage	PP (2019)	PR (2019)	PVR (2020)	VR (2021)	SC (2018)
Model		BEV 3D Box				
PP	-	-	10.66	<u>16.44</u>	<u>7.81</u>	22.58
PR	S1	15.09	41.77	-	-	7.81
	S2	17.51	47.41	-	-	24.43
PVR	S1	7.07	26.50	10.14	16.34	-
	S2	6.83	25.37	10.22	16.51	-
VR	S1	7.13	26.45	10.20	15.99	6.48
	S2	7.11	26.44	10.16	16.18	6.38
						21.34
						-
						1.44
						-
						10.27
						-
						1.26
						8.98

Table 4: ASR of adversarial meshes in a black-box setting, with surrogate models in the columns and target models in the rows. Bold indicates the best, and underlined denotes the second best.



Figure 5: As shown in the figure. From left to right, we construct an adversarial vehicle for clean, vanilla, PhyAdv, MR(9), and MR(8), respectively.

	Clean				Noisy			
	CutIn	FLV0	FLV	OFLV	CutIn	FLV0	FLV	OFLV
Benign	0.61	0.71	0.67	0.35	0.73	0.70	0.70	0.51
Vanilla	0.58	0.52	0.56	0.36	0.41	0.34	0.37	0.33
PhyAdv	0.55	0.52	0.51	0.34	0.33	0.34	0.30	0.32
MR(8)	0.33	0.44	0.37	0.23	0.19	0.28	0.21	0.21
MR(9)	0.36	0.42	0.36	0.23	0.17	0.26	0.20	0.21

Table 5: The mAP@0.7 results on CutIn, FLVWO, FLV, and OFLV scenarios using PointPillar, with confidence $\delta = 0.4$. ‘Clean’ denotes CarLA-generated scenes without noise; ‘Noisy’ includes added simulation noise.

work³, focusing on four safety-critical cases: Cut-In, FLV, OFLV, and FLVWO. In CutIn, a vehicle suddenly merges into the ego vehicle’s lane, testing the system’s ability to perceive and react quickly. FLV (Follow Leading Vehicle) involves the ego vehicle trailing another under normal conditions, assessing persistent and reliable perception. In OFLV (Other Follow Leading Vehicle), the ego vehicle must overtake an obstructing car to continue following a farther target, challenging perception under occlusion. Finally, FLVWO (Follow Leading Vehicle With Obstacle) combines following behavior with unexpected obstacle avoidance, requiring both timely and persistent perception. Each scenario is tested under two conditions, *i.e.* Clean (no added noise) and Noisy (Gaussian noise with $\sigma = 0.2$), to simulate different sensor conditions. We compare five object types: no attack (benign), a basic sphere (vanilla), adversarial objects generated by phyADV, and two adversarial variants optimized for mis-

³https://github.com/carla-simulator/scenario_runner

detection loss (MR(8) and MR(9)). Quantitative results, as Table 5, show that MR(9) outperforms in CarLA simulation scenarios.

Challenges

Compared to digital adversarial attacks on LiDAR-based systems, physically realizable adversarial object attacks remain underexplored, particularly in terms of implementation challenges. Addressing this problem is crucial for safety-critical applications such as autonomous driving, where real-world robustness and compliance with safety regulations are essential. Several open challenges persist. For example, commonly used constraints such as bounding box limits and surface smoothness (*e.g.*, via Laplacian loss) do not adequately capture physical feasibility. There is a need for more effective constraints that better reflect 3D object plausibility while remaining differentiable for optimization. Additionally, while we explore various formulations for adversarial misdetection loss, practical implementation, especially for two-stage detectors, remains difficult. Backpropagating through refinement stages is non-trivial. Besides, current designs often rely only on coarse metrics like IoU, logits, or classification probabilities. Our framework includes a feasible implementation, but we believe improved loss formulations incorporating richer features could further enhance attack effectiveness.

Another underexplored issue is the reflectivity of adversarial objects. In real-world autonomous driving datasets such as KITTI and nuScenes, point clouds are typically stored in an $N \times 4$ format, where the first three channels represent spatial coordinates, and the fourth channel encodes reflectivity, *i.e.*, the intensity of the laser return. We neutralize reflectivity in our evaluation by setting it to zero across all points, ensuring a pure geometric baseline and isolating the effects of spatial structure on model behavior. Nevertheless, while reflectivity can enhance classification performance, its value distribution varies significantly across different LiDAR sensors, leading to two key challenges: (1) models trained on different datasets may learn inconsistent reflectivity patterns, and (2) existing work’s focus is on analyzing geometric structure, not sensor-specific intensity cues.

Conclusion

In this work, we propose a unified, device-agnostic framework for generating and evaluating physically realizable adversarial objects in LiDAR-based 3D object detection. Our framework supports diverse model architectures, attack settings, and loss designs, offering flexibility while enabling reproducible benchmarking. By abstracting the key components of physical attacks and validating their transferability from simulation to the real world, we demonstrate that simulation can serve as an effective proxy for physical feasibility. We also highlight underexplored factors, such as reflectivity and constraint formulation, that influence attack success. Alongside our open-source code and standardized evaluation protocol, we aim to accelerate progress and foster deeper understanding of adversarial robustness in real-world perception systems.

References

- Abdelfattah, M.; Yuan, K.; Wang, Z. J.; and Ward, R. 2021. Towards universal physical attacks on cascaded camera-lidar 3d object detection models. In *2021 IEEE International Conference on Image Processing (ICIP)*, 3592–3596. IEEE.
- Aung, N. H. H.; Sangwongngam, P.; Jintamethasawat, R.; Shah, S.; and Wuttisittikulwij, L. 2024. A review of lidar-based 3d object detection via deep learning approaches towards robust connected and autonomous vehicles. *IEEE Transactions on Intelligent Vehicles*.
- Cao, Y.; Xiao, C.; Cyr, B.; Zhou, Y.; Park, W.; Rampazzi, S.; Chen, Q. A.; Fu, K.; and Mao, Z. M. 2019a. Adversarial sensor attack on lidar-based perception in autonomous driving. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, 2267–2281.
- Cao, Y.; Xiao, C.; Yang, D.; Fang, J.; Yang, R.; Liu, M.; and Li, B. 2019b. Adversarial objects against lidar-based autonomous driving systems. *arXiv preprint arXiv:1907.05418*.
- Deng, J.; Shi, S.; Li, P.; Zhou, W.; Zhang, Y.; and Li, H. 2021. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 1201–1209.
- Deschaud, J.-E. 2021. KITTI-CARLA: a KITTI-like dataset generated by CARLA Simulator. *arXiv e-prints*, arXiv:2109.00892.
- Engelmann, F.; Stückler, J.; and Leibe, B. 2017. SAMP: shape and motion priors for 4d vehicle reconstruction. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 400–408. IEEE.
- Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, 3354–3361. IEEE.
- Huang, Q.; Dong, X.; Chen, D.; Zhou, H.; Zhang, W.; and Yu, N. 2022. Shape-invariant 3D Adversarial Point Clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15335–15344.
- Ju, Z.; Zhang, H.; Li, X.; Chen, X.; Han, J.; and Yang, M. 2022. A survey on attack detection and resilience for connected and automated vehicles: From vehicle dynamics and control perspective. *IEEE Transactions on Intelligent Vehicles*, 7(4): 815–837.
- Kurakin, A.; Goodfellow, I. J.; and Bengio, S. 2018. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, 99–112. Chapman and Hall/CRC.
- Lang, A. H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; and Beijbom, O. 2019. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12697–12705.
- Li, J.; Schmidt, F.; and Kolter, Z. 2019. Adversarial camera stickers: A physical camera-based attack on deep learning systems. In *International conference on machine learning*, 3896–3904. PMLR.
- Möller, T.; and Trumbore, B. 2005. Fast, minimum storage ray/triangle intersection. In *ACM SIGGRAPH 2005 Courses*, 7–es.
- Ren, H.; Huang, T.; and Yan, H. 2021. Adversarial examples: attacks and defenses in the physical world. *International Journal of Machine Learning and Cybernetics*, 12(11): 3325–3336.
- Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; and Li, H. 2020. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10529–10538.
- Shi, S.; Wang, X.; and Li, H. 2019. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 770–779.
- Song, X.; Wang, P.; Zhou, D.; Zhu, R.; Guan, C.; Dai, Y.; Su, H.; Li, H.; and Yang, R. 2019. ApolloCar3d: A large 3d car instance understanding benchmark for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5452–5462.
- Song, Z.; Liu, L.; Jia, F.; Luo, Y.; Jia, C.; Zhang, G.; Yang, L.; and Wang, L. 2024. Robustness-aware 3d object detection in autonomous driving: A review and outlook. *IEEE Transactions on Intelligent Transportation Systems*.
- Sun, J.; Cao, Y.; Chen, Q. A.; and Mao, Z. M. 2020. Towards robust {LiDAR-based} perception in autonomous driving: General black-box adversarial sensor attack and countermeasures. In *29th USENIX Security Symposium (USENIX Security 20)*, 877–894.
- Team, O. D. 2020. OpenPCDet: An Open-source Toolbox for 3D Object Detection from Point Clouds. <https://github.com/open-mmlab/OpenPCDet>.
- Thys, S.; Van Ranst, W.; and Goedemé, T. 2019. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 0–0.
- Tu, J.; Ren, M.; Manivasagam, S.; Liang, M.; Yang, B.; Du, R.; Cheng, F.; and Urtasun, R. 2020. Physically realizable adversarial examples for lidar object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13716–13725.
- Wang, J.; Li, F.; Zhang, X.; and Sun, H. 2023. Adversarial Obstacle Generation against LiDAR-Based 3D Object Detection. *IEEE Transactions on Multimedia*.
- Wen, Y.; Lin, J.; Chen, K.; Chen, C.; and Jia, K. 2019. Geometry-aware generation of adversarial point clouds. *arXiv preprint arXiv:1912.11117*.
- Yan, Y.; Mao, Y.; and Li, B. 2018. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10): 3337.
- Zhang, H.; Cheng, L.; He, Q.; Huang, W.; Li, R.; Sicre, R.; Huang, X.; Hermanns, H.; and Zhang, L. 2024. Eidos: Efficient, Imperceptible Adversarial 3D Point Clouds. *arXiv preprint arXiv:2405.14210*.
- Zhang, H.; Furon, T.; Amsaleg, L.; and Avrithis, Y. 2022. Deep Neural Network Attacks and Defense: The Case of Image Classification. *Multimedia Security*, 1: 41–75.
- Zhang, J.; Chen, L.; Liu, B.; Ouyang, B.; Xie, Q.; Zhu, J.; Li, W.; and Meng, Y. 2021. 3d adversarial attacks beyond point cloud. *arXiv preprint arXiv:2104.12146*.
- Zhang, Y.; Hou, J.; and Yuan, Y. 2024. A comprehensive study of the robustness for lidar-based 3d object detectors against adversarial attacks. *International Journal of Computer Vision*, 132(5): 1592–1624.
- Zhao, Z.; Liu, Z.; and Larson, M. 2021. On success and simplicity: A second look at transferable targeted attacks. *Advances in Neural Information Processing Systems*, 34: 6115–6128.
- Zhou, Z.; Tang, D.; Wang, X.; Han, W.; Liu, X.; and Zhang, K. 2018. Invisible mask: Practical attacks on face recognition with infrared. *arXiv preprint arXiv:1803.04683*.
- Zhu, Y.; Miao, C.; Zheng, T.; Hajiaghajani, F.; Su, L.; and Qiao, C. 2021. 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*, 1945–1960.