# A New Adversarial Perspective for LiDAR-Based 3D Object Detection

**Shijun Zheng**[1,2], **Weiquan Liu**[3*], **Yu Guo**[1,2], **Yu Zang**[1,2*], **Siqi Shen**[1,2], **Cheng Wang**[1,2]

[1]Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, China
[2]Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, School of Informatics, Xiamen University, China
[3]College of Computer Engineering, Jimei University, China
{zhengshijun, gouyu}@stu.xmu.edu.cn, {siqishen, cwang}@xmu.edu.cn, wqliu@jmu.edu.cn, zangyu7@126.com

## Abstract

Autonomous vehicles (AVs) rely on LiDAR sensors for environmental perception and decision-making in driving scenarios. However, ensuring the safety and reliability of AVs in complex environments remains a pressing challenge. To address this issue, we introduce a real-world dataset (ROLiD) comprising LiDAR-scanned point clouds of two random objects: water mist and smoke. In this paper, we introduce a novel adversarial perspective by proposing an attack framework that utilizes water mist and smoke to simulate environmental interference. Specifically, we propose a point cloud sequence generation method using a motion and content decomposition generative adversarial network named PCS-GAN to simulate the distribution of random objects. Furthermore, leveraging the simulated LiDAR scanning characteristics implemented with Range Image, we examine the effects of introducing random object perturbations at various positions on the target vehicle. Extensive experiments demonstrate that adversarial perturbations based on random objects effectively deceive vehicle detection and reduce the recognition rate of 3D object detection models.

**Extended version** — http://arxiv.org/abs/2412.13017

## Introduction

LiDAR is an important support for modern autonomous driving systems in achieving scene perception due to its ability to capture accurate depth information (Xia et al. 2021). Learning-based LiDAR point cloud methods (Xia et al. 2023, 2024) have recently become popular with the development of deep learning. However, previous studies have shown that deep neural network models are vulnerable to adversarial attacks, causing the model to produce erroneous outputs (Szegedy et al. 2014; Huang et al. 2022). Malicious attacks on deep network models can significantly compromise the security of autonomous driving systems.

Currently, 3D adversarial attacks are primarily focused on the digital domain, often overlooking the potential for adversarial attacks in real-world scenarios. In the digital domain, adversarial attack methods include perturbing points, adding independent points or adversarial clustering (Xiang, Qi, and Li 2019), and deleting points (Zheng et al. 2019). In addition, using gradients to implement adversarial attacks is also a major method (Ma et al. 2020), but the generated adversarial examples are prone to outliers. Optimization-based methods (Zheng et al. 2023; Wen et al. 2020) can generate adversarial examples with better geometric properties, but this method performs slowly. In the frequency domain, the imperceptibility and transferability of adversarial examples can be improved based on high-frequency and low-frequency information. However, the adversarial examples generated by these methods are not operable in the real world.

The methods for addressing adversarial attacks in real-world scenarios can be divided into two categories. The first is a sensor-level attack. Attackers use malicious laser pulses to simulate LiDAR return pulses and inject attack signals into LiDAR signals (Cao et al. 2019). The second type is a model-level LiDAR attack, which is also the main adversarial attack method. For example, adding adversarial objects

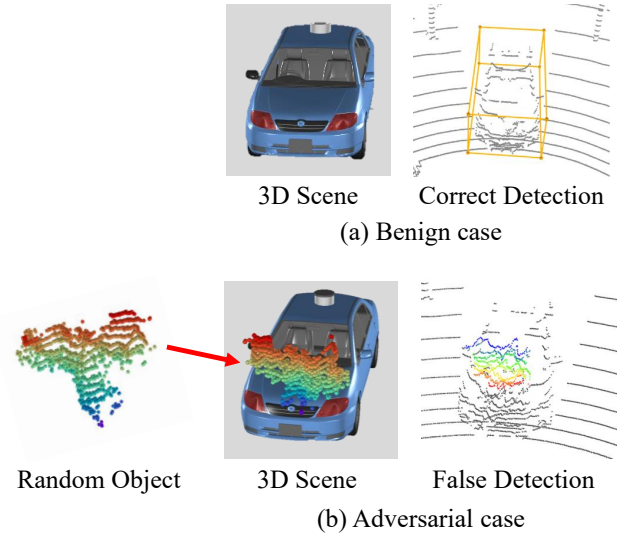


Figure 1: We use random objects to camouflage the target vehicle and fool the LiDAR detector. In benign cases, the 3D object detector correctly detects the vehicle. In adversarial cases, the detector fails to identify the vehicle when random objects are superimposed on it.

*Corresponding Author.

to the roof of a vehicle makes it invisible to LiDAR detectors, thereby deceiving the detection system (Tu et al. 2020). Adversarial objects are generated optimally using adversarial attack methods in the digital domain, and then 3D printing is used to produce this adversarial example in the real world. Such adversarial examples placed on real roads can mislead autonomous vehicles equipped with LiDAR (Cao et al. 2021). These methods seek to generate optimal adversarial objects, which generally have a fixed appearance and are not prone to deformation, such as geometric objects (Tu et al. 2020), traffic cones (Cao et al. 2021), etc.

In this paper, different from existing 3D adversarial attacks on real scenes, we focus on the adversarial nature of random objects to LiDAR. We explore the adversarial capabilities of two random objects: water mist and smoke. To this end, we introduce an adversarial attack framework based on random object interference from a new perspective. First, since random objects have complex physical meanings, they are difficult to describe quantitatively and qualitatively. Therefore, we designs a motion and content separation generative adversarial network method for point cloud sequence generation. This method is used to generate point cloud sequences of random objects, which simulate the distribution of random objects during LiDAR scanning. Second, to verify the attack capability of random objects, we use different methods to superimpose simulated random objects on different positions of vehicle targets in public autonomous driving datasets to generate adversarial data. These generated adversarial data are then used to attack state-of-the-art 3D object detection methods.

Overall, the contributions of this paper can be summarized as follows:

- We are the first to delve into the adversarial nature of random objects for LiDAR perception in real-world scenarios. By employing two random objects, namely water mist and smoke, we demonstrate adversarial attacks on 3D object perception in real scenes.
- We are the first to propose a generative adversarial network framework (PCS-GAN) for point cloud sequence generation. This framework is used to generate random objects that simulate the data characteristics of LiDAR scanning.
- We construct a LiDAR point cloud dataset (ROLiD) of random objects, including water mist and smoke data. The dataset will be released for public research.
- Our method effectively attacks state-of-the-art 3D detectors on KITTI and nuScenes, with attack success rates greater than 80% for most models.

## Related Work

**Adversarial attacks in the digital domain.** The 3D point cloud adversarial attack method was originally extended from the image adversarial attack method. Attackers seek to generate adversarial examples with high attack success rates and better imperceptibility. Xiang et al. (Xiang, Qi, and Li 2019) conducted pioneering research on adversarial attacks targeting 3D point clouds, extending the $C\&W$ attack to this domain. Wen et al. (Wen et al. 2020) argued that regularization terms used to constrain perturbation size fail to ensure imperceptibility. To address this limitation, they proposed a local curvature consistency measure for evaluating point cloud similarity from a geometric perspective and developed a geometry-aware attack method. In addition, generating high-quality adversarial samples with minimal cost remains a crucial research focus (Kim et al. 2021; Zheng et al. 2023). For example, Zheng et al. (Zheng et al. 2023) introduced a local region adversarial attack method on 3D point clouds. In recent years, to improve the transferability and imperceptibility of adversarial examples, researchers have focused increasingly on attacks in the graph spectral domain (Hu, Liu, and Hu 2022; Tao et al. 2023). Hu et al. (Hu, Liu, and Hu 2022) proposed a new point cloud attack paradigm, namely graph spectral domain attack (GSDA), which generates adversarial samples by perturbing the transformation coefficients corresponding to different geometric structures in the graph spectral domain.

**Adversarial attacks in real scenarios.** Implementing adversarial attacks in real-world settings is notably more complex and presents significant security challenges for autonomous driving systems. Tu et al. (Tu et al. 2020) proposed an adversarial attack method to generate adversarial objects of different geometric shapes, and placing them on top of the car can fool the LiDAR detector, causing the car to be invisible. Cao et al. (Cao et al. 2021) explored the security vulnerabilities of multi-sensor autonomous driving systems by formulating adversarial attacks as an optimization problem. To this end, the authors introduced MSF-ADV (Cao et al. 2021) to generate physically realizable, adversarial 3D-printed objects that can mislead autonomous driving systems. The attack algorithm was evaluated using an industrial-grade autonomous driving system in the real world and achieved a success rate of more than 90%. Considering the geometric characteristics of 3D objects and the invariance of physical transformation, Miao et al. (Dong et al. 2022) introduced Gaussian curvature into the regularization term to generate natural and robust 3D adversarial samples in the physical world. Zhu et al. (Zhu et al. 2021) proposed an adversarial attack method to find specific attack locations in the real world. Placing any object with a reflective surface, such as a commercial drone, at these attack locations can mislead LiDAR sensing systems. This method is also the first to study the impact of adversarial position on LiDAR perception models.

In summary, compared with existing adversarial attack methods in the real scenarios, our work is oriented towards adversarial attacks on random objects such as water mist and smoke. We generate random objects through algorithms and simulate their distribution. We do not use optimization methods to find optimal adversarial objects. Such adversarial objects have a fixed geometric appearance and are not easily deformed. At the same time, we explore the attack performance when adding random object perturbations at different locations of the target object. In the process of adding perturbation, we consider the occlusion problem of the target object and simulated the physical characteristics of real LiDAR scanning.

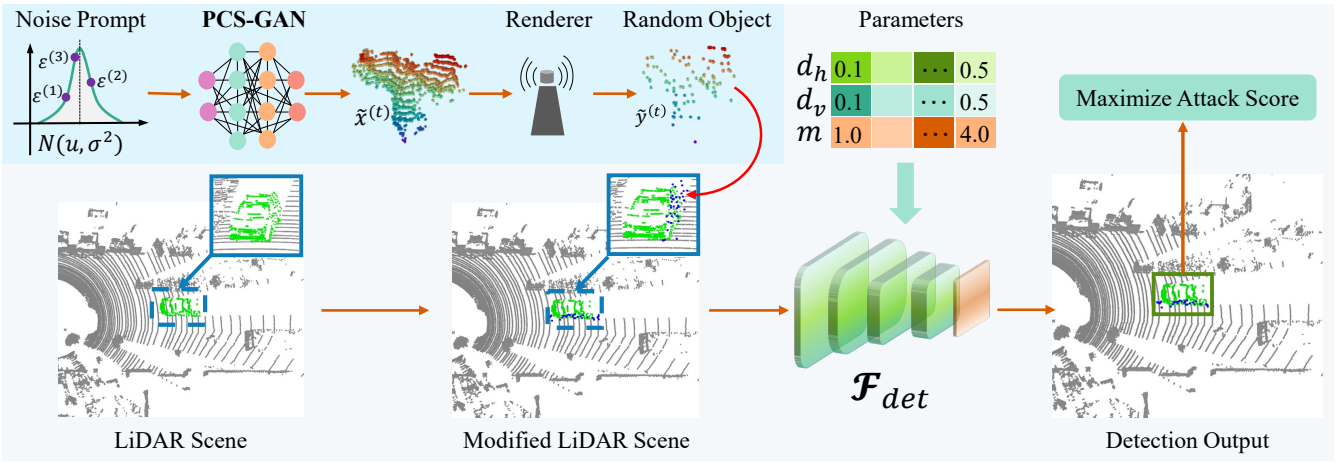


Figure 2: Our adversarial attack framework based on random object perturbations. We propose PCS-GAN to generate random object sequences ($\widetilde{P}_T = [\widetilde{x}^{(1)}, \widetilde{x}^{(2)}, \cdots, \widetilde{x}^{(K)}]$) and use a range image representation approximate renderer to simulate LiDAR scanning. We attach random objects to the target vehicle by setting different fusion modes ($m$) and fusion densities ($d_h, d_v$) to maximize the attack score of the target vehicle on LiDAR detector ($\mathcal{F}_{det}$).

# Method

## Problem Formulation

In this paper, we aim to use random objects as camouflage to conduct adversarial attacks on 3D point clouds in real scenarios. We formulate the problem of attacks based on random object perturbations into two parts, illustrated in Figure 2. The first part is the generation of random objects. We use the real random object $y^{train}$ to train the deep model $\mathcal{F}$ to obtain the optimal parameters $\theta^*$ when the noise $r^{train}$ is triggered, as follows:

$$\mathcal{F}_{\theta^*} = \arg\min_{\theta} \left( \mathcal{L}^{\text{full}} \left( \mathcal{F}_{\theta} \left( \boldsymbol{r}^{\text{train}} \right), \boldsymbol{y}^{\text{train}} \right) \right). \tag{1}$$

The second part is the generation and optimization of adversarial examples. We use the optimal random objects generated by noise $r^{val}$ under the optimal parameters of network $\mathcal{F}$ and fuse them with the target vehicle into adversarial point clouds. Subsequently, the attack score of the target vehicle $s^{val}$ is maximized on the LiDAR detector $\mathcal{F}_{det}$ by optimizing different fusion modes $m$ and fusion densities $d$ as follows:

$$\mathcal{F}_{\theta^*}^{\eta^*} = \max_{\eta} \left( \mathcal{F}_{det} \left( \mathcal{F}_{\theta^*}^{\eta} \left( \boldsymbol{r}^{\text{val}} \right), \boldsymbol{s}^{\text{val}} \right) \right), \tag{2}$$

where parameter $\eta = [m, d]$.

## Dataset of Random Objects

We present a LiDAR point cloud dataset (ROLiD) of random objects for the study of data simulation and adversarial attacks in the real world.

**Data acquisition.** We use 32-line LiDAR to collect data, including water mist data and smoke data as shown in Figure 3. The collection environment is an open factory building, and the supplementary material explains the collection equipment and detailed data acquisition methods.

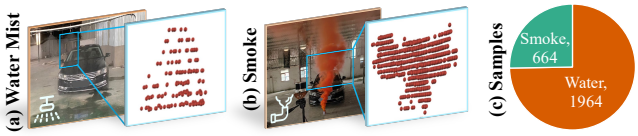


Figure 3: Random objects LiDAR Dataset (ROLiD). (a) and (b) are the data collection scheme and local point cloud visualization respectively. (c) represents the data used for the PCS-GAN network.

**Water mist data.** We spray water mist in the four directions of the vehicle's head, tail, left, and right, and use LiDAR to scan the point cloud directly against the vehicle, as shown in Figure 3(a). When collecting water mist point clouds in each direction of the vehicle, the LiDAR is fixed 10 meters away from the vehicle, and four intensities of water pressure are used, namely 0.45Mpa, 0.50Mpa, 0.55Mpa, and 0.60Mpa. Under each water pressure, we set the distance between the water mist position and the car to 2 meters, 4 meters, 6 meters, and 8 meters, respectively. The collection time of water mist point clouds at each distance under each pressure is about 3 minutes.

**Smoke data.** We released smoke near the vehicle in four directions: the front, rear, left, and right sides. LiDAR was then used to scan the point cloud directly from these positions, as shown in Figure 3(b). The smoke was emitted for approximately 2-3 minutes.

**Data characteristic.** We quantified the impact of water mist data on vehicles to demonstrate the adversarial nature of random objects. Under the same conditions for collecting water mist data, we use LiDAR to collect target vehicle point clouds without spraying water mist. To quantify the impact of the water mist on the vehicle, We counted the number of vehicle point cloud before and after it obscuration by water mist and calculated the occlusion ratio. The quantitative re-

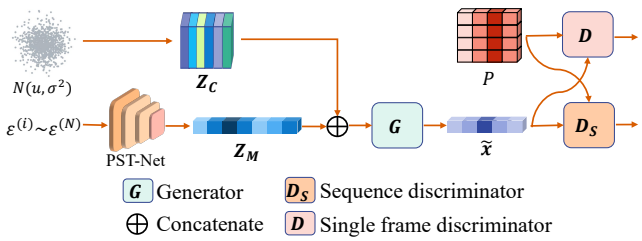


Figure 4: Generative adversarial network framework PCS-GAN for point cloud sequence generation. $P$ is real point cloud, $\widetilde{x}$ is fake point cloud , $Z_C$ is content features, $Z_M$ motion features.

sults for different distances under different pressures can be found in supplementary material.

## Sequence Generation of Random Objects

Random objects such as water mist and smoke have different distribution patterns depending on the surrounding environment. To better simulate the distribution of random objects during LiDAR scanning, we propose a novel Point Cloud Sequence generation method using a Generative Adversarial Network with a motion and content decomposition strategy (PCS-GAN).

Separating content and motion is an effective method for video generation (Tulyakov et al. 2018). Similarly, for the problem of point cloud sequence generation, the motion information between sequences can be separated from the content of the point cloud itself. Therefore, we can decompose the point cloud latent space $Z$ into content space $Z_C$ and motion space $Z_M$. For random objects, the motion space $Z_M$ is used to model the changing characteristics between point cloud sequences, and the content space $Z_C$ is used to model the invariant characteristics of the point cloud.

The proposed PCS-GAN consists of four components, namely PST-Net, generator $G$, single-frame discriminator $D$, and sequence discriminator $D_S$, as shown in Figure 4.

We implement a feature extraction network, PST-Net, based on point spatio-temporal convolution (Fan et al. 2022) to extract temporal and spatial features of point cloud sequences. The input of the network is a point cloud sequence $\varepsilon = (\varepsilon^{(1)}, \varepsilon^{(2)}, \cdots, \varepsilon^{(N)})$ obeying Gaussian distribution, and the output is the information representation $Z_M$ of the point cloud sequence.

The goal of the generator $G$ is to generate point clouds with similar statistical distributions to the real point clouds. Considering the dynamic changing characteristics of random objects, in order to better learn the complex shapes of objects, we use Warping-based Generator (Tang et al. 2022). The motion information $Z_M$ and content information $Z_C$ of the point cloud sequence are concatenated as the input of the generator, and the output is a sequence of K-frame point clouds.

The discriminators $D$ and $D_S$ are used to determine whether the point cloud generated by the generator $G$ is a real point cloud. Among them, the discriminator $D$ is used to determine whether the single-frame point cloud of the point cloud sequence generated by the generator $G$ is a real point cloud. We adopt PointNet (Qi et al. 2017) as the single-frame point cloud discriminator $D$. For generating point clouds, the input to the discriminator is a randomly sampled frame from the point cloud sequence generated by generator $G$. $D_S$ is a temporal discriminator (Li, Li, and Farimani 2021) of a point cloud sequence, which is used to determine whether the point cloud sequence generated by the generator $G$ is a real point cloud. We sample three consecutive frame point clouds from the point cloud sequence generated by generator $G$ as input to predict the temporal consistency confidence of the point cloud sequence. Similarly, for the real point cloud sequence, three consecutive frames are used as the input of $D_S$.

| | Methods | Hausdorff Distance ↓ | Chamfer Distance ↓ |
|---|---|---|---|
| water mist | PCS-GAN | 0.38 | 0.02 |
| water mist | DERT | 5.81 | 1.56 |
| smoke | PCS-GAN | 0.68 | 0.11 |

Table 1: Realistic evaluation of generated random objects.

To generate a point cloud sequence, a random vector $\varepsilon = (\varepsilon^{(1)}, \varepsilon^{(2)}, \cdots, \varepsilon^{(N)})$ of length $N$ is sampled from the Gaussian distribution. After the motion information is extracted through the PST-Net network, the sequence length will be reduced to K, recorded as $Z_M = (Z_M^1, Z_M^2, \cdots, Z_M^K)$. Therefore, a random vector of length $K$ is sampled from a Gaussian distribution and concated with $Z_M$ as the input to the generator $G$. The sequence generated by the generator $G$ is marked as $\widetilde{P}_T = [\widetilde{x}^{(1)}, \widetilde{x}^{(2)}, \cdots, \widetilde{x}^{(K)}]$, from which we randomly sample a frame of point clouds as $\widetilde{P} = [\widetilde{x}^{(i)}]$, and the point cloud sequence of three consecutive frames is marked as $\widetilde{P}_T = [\widetilde{x}^{(i-1)}, \widetilde{x}^{(i)}, \widetilde{x}^{(i+1)}]$. Similarly, a frame randomly sampled from the real point cloud is recorded as $P = [x^{(i)}]$, and a point cloud sequence of three consecutive frames is recorded as $P_T = [x^{(i-1)}, x^{(i)}, x^{(i+1)}]$.

We mark the PST-Net network as $P_M$. Therefore, according to GAN (Creswell et al. 2018) and the video generation framework MoCoGAN (Tulyakov et al. 2018), the learning problem of PCS-GAN is expressed as follows:

$$\max_{G,P_M} \min_{D,D_S} \mathcal{F}_{\mathrm{P}}\left(D, D_{\mathrm{S}}, G, P_M\right). \tag{3}$$

The objective function of $\mathcal{F}_p$ is as follows:

$$\begin{aligned} &\mathbb{E}_{\mathbf{P}}\left[-\log D\left(P\right)\right] + \mathbb{E}_{\widetilde{\mathbf{P}}}\left[-\log\left(1 - D\left(\widetilde{P}\right)\right)\right] + \\ &\mathbb{E}_{\mathbf{P}}\left[-\log D_{\mathrm{S}}\left(P_{\mathrm{T}}\right)\right] + \mathbb{E}_{\widetilde{\mathbf{P}}}\left[-\log\left(1 - D_{\mathrm{S}}\left(\widetilde{P}_{\mathrm{T}}\right)\right)\right]. \end{aligned} \tag{4}$$

During the training process, the sequence discriminator $D_S$ adopts the temporal discriminator loss (Li, Li, and Farimani 2021) of TPU-GAN. Discriminator $D$ uses improved WGAN loss (Gulrajani et al. 2017). In order to further reduce the local differences between the generated point cloud and the real point cloud, we use stitching loss (Tang et al. 2022) to enhance the generator $G$. When training PCS-GAN, we first fix the generator $G$ and PST-Net and update the discriminators $D$ and $D_S$. Then the discriminators $D$ and

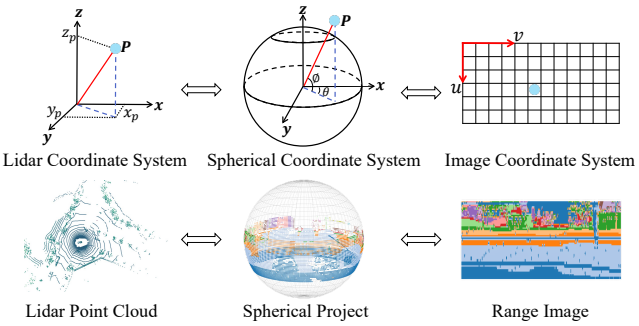


Figure 5: Obtaining a range image of the point cloud via spherical projection. The first line describes the three coordinate systems involved in the conversion process, while the second line outlines the changes in data format during this transformation.

| Dataset | Model | Attack success rate | |
|---|---|---|---|
| | | water mist | smoke |
| KITTI | TED | 87.29 | 76.02 |
| | Focals Conv - F | 95.32 | 81.47 |
| | Part-A2-Anchor | 92.05 | 90.95 |
| | Part-A2-Free | 89.63 | 87.53 |
| | PV-RCNN | 86.34 | 87.27 |
| | PointPillar | 93.26 | 86.99 |
| nuScenes | VoxelNeXt | 83.18 | 82.87 |
| | TransFusion-L | 92.61 | 93.27 |
| | CenterPoint (voxel_size=0.1) | 86.67 | 87.60 |
| | CenterPoint (voxel_size=0.075) | 85.06 | 85.73 |
| | CenterPoint-PointPillar | 86.26 | 81.43 |
| | PointPillar-MultiHead | 84.48 | 81.54 |

Table 2: Attack performance of 3D object detection models on different datasets under random object perturbations.

$D_S$ are fixed, and the generator $G$ and PST-Net are updated. Update each part alternately until the end of training.

**Realistic evaluation of generated random objects.** The similarity between the generated data and the real data is evaluated by the Hausdorff and Chamfer distance, i.e., the smaller the distance value, the higher the similarity. The evaluation results of PCS-GAN and DERT (Huang et al. 2024) are shown in Table 1. In the experiments, PCS-GAN was trained for 2000 epochs with a batch size of 4, and the initial sampling length $N$ of the point cloud sequence was set to 16.

## Adversarial Point Cloud Sequence Generation

We fuse the generated random object sequence and scene point cloud to form a new point cloud sequence. We use water mist as an example to describe the fusion process. First, align the coordinate system of the water mist with the coordinate system of the scene point cloud. Second, the water mist is attached to the body of the target vehicle under the coordinate system of the scene point cloud to complete the fusion. However, this destroys the inherent topological properties of LiDAR point clouds and also introduces occlusion issues. Therefore, we use a range image-based LiDAR simulation approach to decide which points are visible from the LiDAR perspective.

**LiDAR simulation.** The purpose of LiDAR simulation is to solve the occlusion problem caused by the fusion of random objects and point cloud objects in the scene, and to generate a fused point cloud with the real scanning characteristics of LiDAR. We implement LiDAR simulation using range image representation which provides physically accurate rendering as shown in Figure 5. Specifically, when we project the point cloud to the range image, if multiple points are projected to the same pixel at the same time, only the point closest to the LiDAR is retained. This simulates the occurrence of multiple points in the path of the laser beam, with later points being occluded by previous points. Subsequently, we back-project the range image to obtain the point cloud again. Therefore, the characteristics of a LiDAR scan are simulated during the projection process by replacing the depth values on the range image. We use this simulation approach to approximate the LiDAR renderer in this work. In addition, the range image has the problem of missing horizontal lines on KITTI. The detail of repair method can be found in supplementary material.

**Mix the water mist with car in point cloud.** After the coordinates are aligned, we select point $A$ of the water mist point cloud and move it to point $B$ of the car in the scene point cloud. Then, the water mist is adjusted according to the direction of the car to fit the body. The distance between the center of mass $O$ of the water mist point cloud and the point $A$ directly above is half the height $\Delta z$ of the point cloud. The height $\Delta z$ of the water mist point cloud is calculated as follows:

$$\Delta z = \frac{1}{2}\left(\max_{(x,y,z)\in P_{mist}} z - \min_{(x,y,z)\in P_{mist}} z\right), \quad (5)$$

where $P_{mist}$ is the water mist point cloud.

We first determine the top centers of the four bodies of the target vehicle, which are replaced by the midpoints of the four sides of the upper rectangle of the annotation box. Then take the top center of the car body facing the LiDAR as point $B$. However, vehicles do not always face the LiDAR head-on in real scenarios. Therefore, we set up four fusion modes, as follows:

- **head/tail side**: When the front or rear of the car faces the LiDAR, the water mist merges with the top center point $B$ of the front or rear of the car.
- **body side**: When one side of the body faces the LiDAR, the water mist merges with the top center point $B$ of the body.

When multiple bodies (up to two) face the LiDAR:

- **two sides**: Fuse water mist at the top midpoint of both bodies at the same time.
- **corner point**: The top point $B$ of the intersection line of the two bodies merges with the water mist

**Mix the water mist with car in range image.** We project the fused point cloud into range image. Because the points in the water mist may not appear on the laser path, we need

| Dataset | Model | Attack success rate | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | left 40° | left 20° | left 10° | left 5° | 0° | right 5° | right 10° | right 20° | right 40° |
| KITTI | TED | 25.02 | 44.14 | 54.12 | 56.33 | 55.74 | 53.72 | 53.44 | 51.94 | 31.79 |
| | Focals Conv - F | 28.42 | 41.90 | 57.39 | 62.11 | 62.03 | 61.71 | 58.33 | 46.15 | 29.01 |
| | Part-A2-Anchor | 41.95 | 53.05 | 58.77 | 60.64 | 59.31 | 57.86 | 58.20 | 56.28 | 41.99 |
| | Part-A2-Free | 33.50 | 47.81 | 53.83 | 56.50 | 55.02 | 53.71 | 52.07 | 47.44 | 32.43 |
| | PV-RCNN | 40.97 | 46.60 | 53.25 | 55.44 | 55.93 | 56.61 | 58.13 | 55.18 | 41.53 |
| | PointPillar | 56.62 | 64.40 | 70.72 | 75.37 | 74.69 | 72.31 | 70.04 | 56.97 | 49.94 |
| nuScenes | VoxelNeXt | 76.34 | 85.08 | 90.84 | 90.15 | 91.44 | 91.71 | 91.90 | 82.23 | 75.14 |
| | TransFusion-L | 86.05 | 88.59 | 95.65 | 95.83 | 95.47 | 96.20 | 96.74 | 94.02 | 85.69 |
| | CenterPoint (voxel_size=0.1) | 80.91 | 88.91 | 92.76 | 93.23 | 92.99 | 92.53 | 93.07 | 89.30 | 80.52 |
| | CenterPoint (voxel_size=0.075) | 82.43 | 88.81 | 91.17 | 92.36 | 91.73 | 93.54 | 93.93 | 90.15 | 84.79 |
| | CenterPoint-PointPillar | 77.56 | 82.69 | 89.01 | 92.31 | 92.49 | 92.67 | 91.30 | 87.00 | 78.66 |
| | PointPillar-MultiHead | 86.69 | 88.34 | 94.32 | 94.94 | 95.36 | 96.28 | 95.87 | 92.88 | 86.27 |

Table 3: Effects of different angles of spraying water mist on attack performance.

to set an error limit in both the horizontal and vertical directions to determine whether the point is visible by LiDAR. Therefore, we use these two constraints, namely density parameters $d$, to control the density of random objects in the fused point cloud. In the experiment, the density parameter in the horizontal direction is denoted as $d_h$, and the density parameter in the vertical direction is denoted as $d_v$.

# Experiments

## Experimental Setup

**Dataset.** We evaluated the performance of adversarial attacks on the KITTI (Geiger, Lenz, and Urtasun 2012) and nuScenes (Caesar et al. 2020) datasets. Since they do not provide labels for the test set, we use the training set and validation set for adversarial attack performance evaluation. For each dataset, we selected 3000 frame point clouds for the attack. Water mist perturbations were applied to each frame to create an adversarial point cloud sequence, and similarly, smoke perturbations were added to generate another adversarial sequence. In the experiments, the lengths of both the water mist and smoke sequences were set to 3.

**Attack models.** We select high-performing 3D point cloud object detection models from each dataset for the attack evaluation. The models targeted for attacks on the KITTI include TED (Wu et al. 2023), Focals Conv-F (Chen et al. 2022), Part-A2-Anchor (Shi et al. 2021), Part-A2-Free (Shi et al. 2021), PV-RCNN (Shi et al. 2020), PointPillar (Lang et al. 2019). Among them, Focals Conv-F is a multimodal model that processes both images and point clouds as inputs. The models targeted for attacks on the nuScenes include VoxelNeXt (Chen et al. 2023), TransFusion-L (Bai et al. 2022), CenterPoint(voxel_size=0.1) (Yin, Zhou, and Krahenbuhl 2021), CenterPoint (voxel_size=0.075) (Yin, Zhou, and Krahenbuhl 2021), CenterPoint-PointPillar (Yin, Zhou, and Krahenbuhl 2021), PointPillar-MultiHead.

**Evaluation metrics.** To evaluate the effectiveness of adversarial attacks involving random objects, we adopt the attack success rate as the primary metric. A successful attack is defined as a case where a vehicle initially detected by the object detection model is no longer detected after perturbations are applied. For detection to be considered successful, the vehicle confidence must exceed 0.5 and the intersection over union (IoU) between the predicted bounding box and the ground truth label must be greater than 0.7. In the experiment, the IoU threshold is set to 0.7.

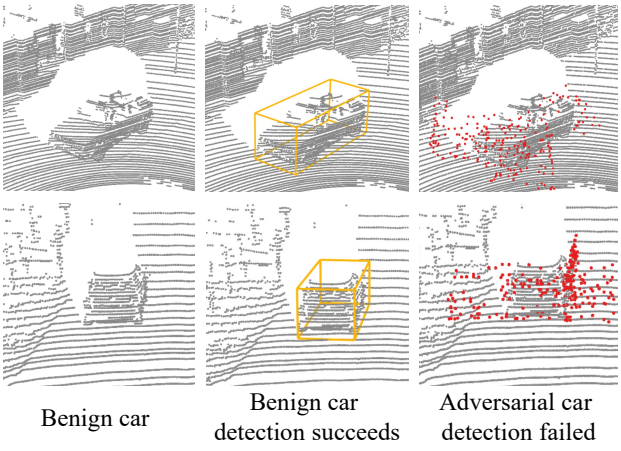


Figure 6: The performance of TED under the perturbation of water mist on KITTI when the fusion mode is two sides. Perturbed points of adversarial examples are marked in red.

## Results on KITTI and nuScenes Dataset

We evaluated various object detection models on the KITTI and nuScenes datasets, as shown in Table 2. The value ranges of the density parameters of KITTI and nuScenes are $d_v \in [0, 0.004]$, $d_h \in [0, 0.5]$ and $d_v \in [0, 0.5]$, $d_h \in [0, 0.5]$ respectively. In the experiment, the fusion mode was set to two sides. For water mist perturbation, we set $d_v = 0.004$, $d_h = 0.5$ on KITTI, and set $d_v = 0.08$, $d_h = 0.08$ on nuScenes. For smoke perturbation, we set $d_v = 0.002$, $d_h = 0.25$ on KITTI, and set $d_v = 0.05$, $d_h = 0.05$ on nuScenes. We used 3000 frames of original point clouds and 3 frames of water mist or smoke sequences to generate 9000 mixed frames, which were then used for evaluating the models. From Table 2, the proposed method achieves high attack success rates on state-of-the-art models under different den-

| Model | Car AP@0.70, 0.70, 0.70 | | | Car AP_R40@0.70, 0.70, 0.70 | | | Car AP@0.70, 0.50, 0.50 | | | Car AP_R40@0.70, 0.50, 0.50 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | easy | mod. | hard | easy | mod. | hard | easy | mod. | hard | easy | mod. | hard |
| PV-RCNN($*$) | 98.72 | 90.02 | 89.70 | 99.31 | 94.78 | 94.42 | 98.72 | 90.03 | 89.70 | 99.31 | 94.78 | 94.42 |
| PV-RCNN(#) | 98.98 | 89.99 | 89.68 | 99.40 | 94.88 | 94.15 | 98.98 | 89.99 | 89.68 | 99.40 | 94.88 | 94.15 |

Table 4: Object detection performance with water mist-augmented point clouds.

| Method | Attack success rate |
|---|---|
| two sides | 86.98 |
| body side | 61.01 |
| head/tail side | 55.30 |
| corner point | 57.22 |

Table 5: Attack performance of TED under different fusion modes.

sity parameters. For example, on the KITTI, the attack success rate ranges from a minimum of 86.34% to a maximum of 95.32% when the water mist density is high. Even when the smoke density is low, the attack success rate still reaches 93.27% on the nuScenes. Moreover, the attack success rates of Focals Conv-F on water mist and smoke are 95.32% and 81.47%, respectively, demonstrating that the proposed attack method is also effective against multimodal models. In addition, Figure 6 shows the attack performance of TED on KITTI under water mist perturbation when the fusion mode is two sides.

## Water Mist Spray Angle Simulation

We simulated the impact of different water mist spraying angles on attack performance, as shown in Table 3. We took the center of mass of the water mist point cloud as the origin and rotated 0°, 5°, 10°, 20°, and 40° to the left and right respectively. The rotated 3 frame water mist sequence and 300 frames of the original point cloud were fused into an adversarial point cloud sequence and used for model evaluation. In the experiment, when the fusion mode is body side, we set $d_v = 0.004$, $d_h = 0.5$ on KITTI, and set $d_v = 0.5$, $d_h = 0.5$ on nuScenes. Experimental results show that the attack ability of water mist can be maintained when the rotation angle is within $10^\circ$, otherwise, the attack ability of water mist will decrease as the rotation angle increases.

## Effect of Water Mist Augmentation

We evaluated the effect of water mist augmentation on object detection performance, as shown in Table 4. PV-RCNN($*$) refers to the model trained and tested on clean point clouds, while PV-RCNN(#) denotes training with both clean and water mist point clouds, with testing performed on clean point clouds. The results indicate a slight improvement in detection performance when the model is trained with water mist-augmented data, demonstrating that such augmentation effectively enhances the model's robustness and detection accuracy.

| Direction | density | Attack success rate |
|---|---|---|
| $d_h$ | 0.1 | 75.73 |
| | 0.2 | 80.92 |
| | 0.3 | 83.74 |
| | 0.4 | 85.49 |
| | 0.5 | 86.98 |
| $d_v$ | 0.001 | 73.24 |
| | 0.002 | 82.64 |
| | 0.004 | 86.98 |

Table 6: Attack performance of TED under different density parameter values when the fusion mode is two sides.

## Ablation Study

**Fusion modes.** We verified the attack performance of different fusion modes, as shown in Table 5. We select 1000 frames of the original point cloud and 3 frames of water mist sequences for fusion on KITTI. Specifically, we set the density parameters $d_v = 0.004$, and $d_h = 0.5$, and the fused point cloud was evaluated on TED. Obviously, the two sides mode has outstanding attack performance.

**Density parameters.** We study the impact of density parameters on attack performance. When the fusion mode is two sides, we use 1000 frames of original point clouds and 3 frames of water mist sequences to generate fused point clouds on KITTI, which are used to evaluate TED. When fixing $d_h = 0.5$ to verify the influence of $d_v$, and fixing $d_v = 0.04$ to verify the influence of $d_h$, the results are shown in Table 6. The results show that as the density increases, the attack performance improves.

# Conclusion

This paper introduces an adversarial attack framework that utilizes random objects to perturb targets in real scenarios. We used two types of advancing object perturbations, water mist and smoke, and verified their attack capabilities on LiDAR perception. We proposed a motion and content decomposition generative adversarial network, PCS-GAN, for point cloud sequence generation. To generate an adversarial point cloud sequence, we fuse the random object sequence produced by PCS-GAN with real point cloud data. In this process, we use a range image to simulate LiDAR scanning characteristics and effectively address the occlusion issue. The attack performance of the adversarial point cloud is verified on multiple autonomous driving datasets. Experimental results show that random objects have strong adversarial attack performance.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No. 62401225, 62471415), the Natural Science Foundation of Xiamen, China (No. 3502Z202472018), the Natural Science Foundation of Fujian Province, China (No. 2024J01115, 2023J01004), and the Jimei University Scientific Research Start-up Funding Project (No.ZQ2024034).

## References

Bai, X.; Hu, Z.; Zhu, X.; Huang, Q.; Chen, Y.; Fu, H.; and Tai, C.-L. 2022. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1090–1099.

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11621–11631.

Cao, Y.; Wang, N.; Xiao, C.; Yang, D.; Fang, J.; Yang, R.; Chen, Q. A.; Liu, M.; and Li, B. 2021. Invisible for both camera and lidar: Security of multi-sensor fusion based perception in autonomous driving under physical-world attacks. In *2021 IEEE Symposium on Security and Privacy (SP)*, 176–194.

Cao, Y.; Xiao, C.; Cyr, B.; Zhou, Y.; Park, W.; Rampazzi, S.; Chen, Q. A.; Fu, K.; and Mao, Z. M. 2019. 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.

Chen, Y.; Li, Y.; Zhang, X.; Sun, J.; and Jia, J. 2022. Focal sparse convolutional networks for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5428–5437.

Chen, Y.; Liu, J.; Zhang, X.; Qi, X.; and Jia, J. 2023. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21674–21683.

Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; and Bharath, A. A. 2018. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1): 53–65.

Dong, Y.; Zhu, J.; Gao, X.-S.; et al. 2022. Isometric 3d adversarial examples in the physical world. *Advances in Neural Information Processing Systems*, 35: 19716–19731.

Fan, H.; Yu, X.; Ding, Y.; Yang, Y.; and Kankanhalli, M. 2022. Pstnet: Point spatio-temporal convolution on point cloud sequences. *ICLR*.

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.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30.

Hu, Q.; Liu, D.; and Hu, W. 2022. Exploring the devil in graph spectral domain for 3d point cloud attacks. In *European Conference on Computer Vision*, 229–248.

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.

Huang, X.; Wu, H.; Li, X.; Fan, X.; Wen, C.; and Wang, C. 2024. Sunshine to rainstorm: Cross-weather knowledge distillation for robust 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2409–2416.

Kim, J.; Hua, B.-S.; Nguyen, T.; and Yeung, S.-K. 2021. Minimal adversarial examples for deep learning on 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 7797–7806.

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, Z.; Li, T.; and Farimani, A. B. 2021. TPU-GAN: learning temporal coherence from dynamic point cloud sequences. In *International Conference on Learning Representations*.

Ma, C.; Meng, W.; Wu, B.; Xu, S.; and Zhang, X. 2020. Efficient joint gradient based attack against sor defense for 3d point cloud classification. In *Proceedings of the 28th ACM International Conference on Multimedia*, 1819–1827.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

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, Z.; Shi, J.; Wang, X.; and Li, H. 2021. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE transactions on pattern analysis and machine intelligence*, 43(8): 2647–2664.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing properties of neural networks. *International Conference on Learning Representations*.

Tang, Y.; Qian, Y.; Zhang, Q.; Zeng, Y.; Hou, J.; and Zhe, X. 2022. WarpingGAN: Warping multiple uniform priors for adversarial 3D point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6397–6405.

Tao, Y.; Liu, D.; Zhou, P.; Xie, Y.; Du, W.; and Hu, W. 2023. 3DHacker: Spectrum-based Decision Boundary Generation for Hard-label 3D Point Cloud Attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 14340–14350.

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.

Tulyakov, S.; Liu, M.-Y.; Yang, X.; and Kautz, J. 2018. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1526–1535.

Wen, Y.; Lin, J.; Chen, K.; Chen, C. P.; and Jia, K. 2020. Geometry-aware generation of adversarial point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6): 2984–2999.

Wu, H.; Wen, C.; Li, W.; Li, X.; Yang, R.; and Wang, C. 2023. Transformation-equivariant 3D object detection for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2795–2802.

Xia, Y.; Gladkova, M.; Wang, R.; Li, Q.; Stilla, U.; Henriques, J. F.; and Cremers, D. 2023. Casspr: Cross attention single scan place recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8461–8472.

Xia, Y.; Shi, L.; Ding, Z.; Henriques, J. F.; and Cremers, D. 2024. Text2Loc: 3D Point Cloud Localization from Natural Language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Xia, Y.; Xu, Y.; Wang, C.; and Stilla, U. 2021. VPC-Net: Completion of 3D vehicles from MLS point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 174: 166–181.

Xiang, C.; Qi, C. R.; and Li, B. 2019. Generating 3d adversarial point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9136–9144.

Yin, T.; Zhou, X.; and Krahenbuhl, P. 2021. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11784–11793.

Zheng, S.; Liu, W.; Shen, S.; Zang, Y.; Wen, C.; Cheng, M.; and Wang, C. 2023. Adaptive local adversarial attacks on 3D point clouds. *Pattern Recognition*, 144: 109825.

Zheng, T.; Chen, C.; Yuan, J.; Li, B.; and Ren, K. 2019. Pointcloud saliency maps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1598–1606.

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.