remote sensing

MDPI

Article

SAGE: Saliency and Geometry Enhanced Transferable Attacks for LiDAR Point Cloud Perception in Remote Sensing

Yuheng Wu , Shiwei Lin *, Shibo Ping, Xingchao Zhai, Zhiyuan Fang, Meijuan Chen and Weiquan Liu

College of Computer Engineering, Jimei University, Xiamen 361021, China; 202421335031@jmu.edu.cn (Y.W.); 202511835018@jmu.edu.cn (S.P.); 202521335019@jmu.edu.cn (X.Z.); 202421336050@jmu.edu.cn (Z.F.); 202521331130@jmu.edu.cn (M.C.); wqliu@jmu.edu.cn (W.L.)
* Correspondence: shiwei.lin@jmu.edu.cn

Highlights

What are the main findings?

- We propose SAGE, a saliency and geometry enhanced transfer attack framework for LiDAR-based 3D perception.
- Extensive experiments on ModelNet40, ScanObjectNN, KITTI, and nuScenes show that SAGE improves black-box transferability across both point cloud classifiers and LiDAR-based 3D object detectors.

What are the implications of the main findings?

- The results reveal that transferable adversarial perturbations can affect not only isolated object classification but also full-scene LiDAR detection pipelines.
- SAGE provides a practical robustness-evaluation tool for safety-critical 3D remote sensing perception systems and suggests a general saliency–geometry direction for transferable point cloud attacks.

Abstract

LiDAR point clouds are widely used in remote sensing perception scenarios, such as autonomous driving. However, LiDAR-based perception models remain vulnerable to adversarial perturbations, which may compromise the reliability of safety-critical 3D perception systems. Among different attack paradigms, transfer-based attacks are particularly practical because they generate adversarial examples on accessible surrogate models and apply the generated examples directly to unknown target models. Nevertheless, existing transferable attacks on point clouds often perturb regions that are discriminative for the surrogate model but insufficiently stable across different architectures, leading to limited transferability and noticeable geometric distortion. To address this problem, we propose **SAGE**, a **S**aliency **A**nd **G**eometry **E**nhanced transferable attack framework for LiDAR point cloud perception in remote sensing. Specifically, SAGE unifies point-coordinate priors with source-model gradient signals to generate a saliency map, which serves as a transferable indicator of vulnerable local structures. SAGE further leverages this map through saliency-guided perturbation allocation and explicit geometric constraints to enhance transferability while preserving point-cloud geometry. To demonstrate the effectiveness of SAGE, we evaluate SAGE on point-cloud classification benchmarks and further validate it on LiDAR-based 3D object detection using KITTI and nuScenes. Experimental results show that SAGE consistently outperforms existing transferable attack methods in attack success rate while preserving favorable geometric quality of adversarial point clouds. These findings demonstrate that SAGE offers an effective and practical framework for assessing the transfer robustness of LiDAR-based remote sensing perception systems.

Check for updates

Academic Editor: Mohammad Awrangjeb

Received: 27 May 2026
Revised: 23 June 2026
Accepted: 3 July 2026
Published: 5 July 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Keywords: 3D point cloud; adversarial attack; transfer attack; remote sensing

1. Introduction

LiDAR is a remote sensing technology that acquires 3D geometric information by emitting laser pulses and measuring their returns. Owing to its accurate spatial representation and robustness to illumination variations, LiDAR point clouds have been widely used in autonomous driving, intelligent transportation, mobile mapping, urban-scale scene understanding, and other remote sensing applications. In conventional surveying and remote sensing workflows, the reliability of LiDAR data is commonly improved through sensor calibration, registration, filtering, denoising, and quality control. Nevertheless, the increasing adoption of end-to-end deep learning introduces an additional source of uncertainty: even when the acquired point cloud appears geometrically reasonable, small data anomalies may still cause unexpected changes in model predictions. Previous studies have shown that point-density variation, ambient noise, occlusion, calibration errors, and adverse weather can substantially affect point-cloud processing and recognition systems [1–3]. Therefore, evaluating the sensitivity of learning-based LiDAR perception models to abnormal point-cloud observations has become an important problem for the reliability and security of remote sensing applications.

Adversarial attacks provide a controlled means of exposing such model vulnerabilities. By introducing deliberately optimized but constrained perturbations into an input point cloud, an attacker can cause incorrect object recognition or missed detections [4–6]. As shown in Figure 1, object-level points can be extracted from a clean LiDAR scene, perturbed in the digital domain, and inserted back into the original scene, potentially changing the output of a perception model. In this work, adversarial perturbations are used as controlled stress-testing samples for identifying model-dependent vulnerabilities, quantifying robustness gaps, and supporting the development of anomaly detection, robust training, and defensive preprocessing methods for LiDAR perception systems.

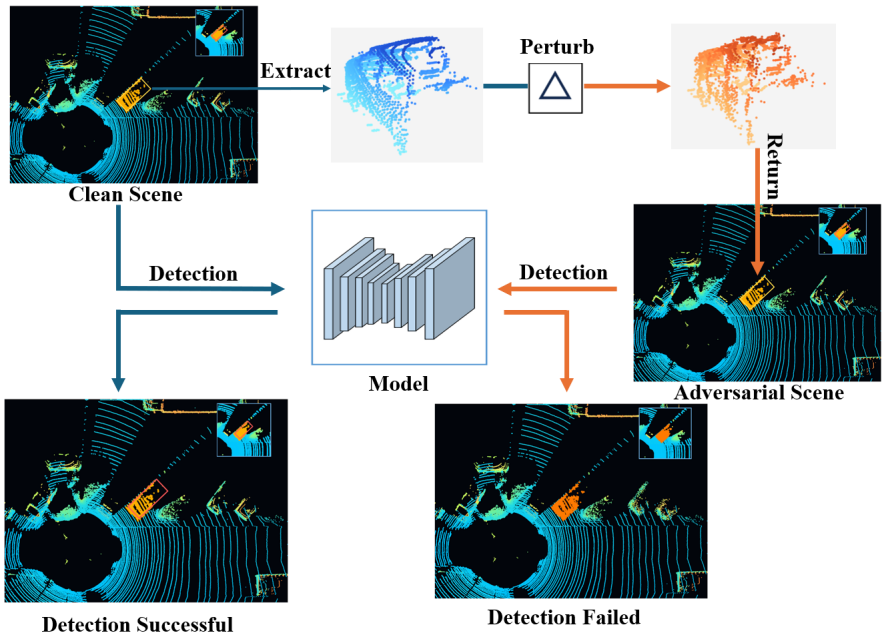


Figure 1. Illustration of adversarial attacks in LiDAR-based remote sensing perception. Object-level points are extracted from a clean scene, perturbed into adversarial points, and then inserted back into the original scene. The clean scene can be correctly perceived, whereas the adversarial scene may cause recognition failure.

Existing adversarial attacks are generally studied under white-box or black-box settings. White-box attacks assume that the attacker can access the parameters and gradients of the target model, which enables direct optimization against the victim model [7–9]. Although such attacks are effective for diagnosing model weaknesses, this assumption is often unavailable for deployed LiDAR perception systems because their internal parameters and architectures are normally inaccessible. Black-box attacks therefore provide a more challenging robustness-evaluation setting. Query-based attacks infer the behavior of the target model through repeated model queries [10], but their high computational and query costs limit their applicability when model access is restricted. Transfer-based attacks instead generate adversarial samples on an accessible surrogate model and directly evaluate them on unknown target models [11–14]. Consequently, cross-model transferability is an important indicator of whether an identified vulnerability is specific to one model or shared by multiple perception architectures.

Recent transferable attacks have mainly been developed for point-cloud classification. Existing methods improve transferability through generative modeling, perturbation decomposition, feature guidance, and hypothesis-space expansion [11–16]. However, different 3D perception models employ distinct neighborhood construction, feature aggregation, and spatial encoding mechanisms. Adversarial samples can therefore overfit the surrogate model's local gradients and fail to generalize to unseen architectures. Moreover, classification-based evaluations mainly consider isolated and normalized object point clouds, whereas LiDAR-based detection operates on full scenes affected by background clutter, occlusion, range-dependent sparsity, and detector-specific preprocessing. A transferable perturbation must remain effective after the attacked object is reinserted into the scene and processed through voxelization, feature extraction, proposal generation, and detection. These differences make transferable robustness evaluation in LiDAR detection substantially more challenging.

To address these problems, we propose **SAGE**, a **S**aliency **A**nd **G**eometry **E**nhanced transferable attack framework for evaluating the robustness of 3D point-cloud perception models. SAGE uses coordinate self-attention to capture structurally discriminative regions and tangent-plane gradient information to identify task-sensitive local structures. The two signals are fused to construct a saliency map that combines model-independent geometric evidence with task-dependent sensitivity. SAGE further introduces saliency-guided patch selection, perturbation-strength allocation, and explicit geometric constraints to reduce unnecessary deformation and alleviate overfitting to surrogate-specific gradients. Beyond controlled evaluations on ModelNet40 [17] and ScanObjectNN [18], we evaluate SAGE on KITTI [19] and nuScenes [20] to investigate whether the identified vulnerabilities remain transferable across full-scene LiDAR detection pipelines.

As illustrated in Figure 2, conventional transferable attacks often search within a relatively coarse adversarial space, whereas SAGE identifies compact regions that are simultaneously structurally important and perception-sensitive. Perturbing these regions improves cross-model attack effectiveness while limiting unnecessary changes to low-impact points. From a remote sensing perspective, this evaluation reveals which geometric structures are consistently relied upon by different learning-based models and provides challenging samples for subsequent robustness analysis and defensive model development.

The main contributions of this work are summarized as follows:

1. We investigate the transferability limitations of black-box adversarial attacks in LiDAR-based 3D remote sensing perception and identify critical-region localization as a key factor for improving cross-model attack effectiveness, providing useful insights for future studies on transferable point cloud attacks.

2. We propose a new saliency and geometry enhanced transferable attack framework, which provides an effective solution to the transferability problem in LiDAR point cloud perception for remote sensing scenarios.
3. We extend the evaluation from conventional point cloud classification benchmarks to realistic remote sensing detection scenarios. Extensive experiments demonstrate that SAGE effectively improves attack transferability in remote sensing perception while preserving the geometric imperceptibility of adversarial point clouds.

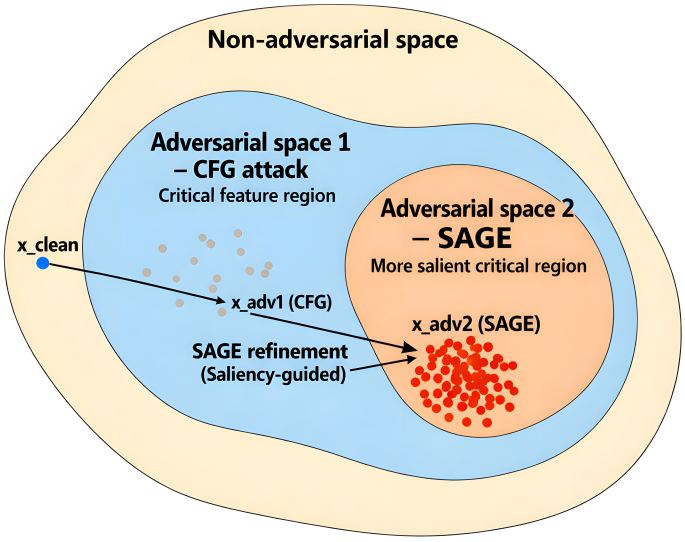


Figure 2. Existing transfer attacks may locate a coarse adversarial space, whereas SAGE searches for a more salient and transferable critical region. Concentrating perturbations in this refined region improves black-box transferability while reducing unnecessary geometric distortion.

2. Related Work

2.1. Robustness of LiDAR in Remote Sensing

The reliability of LiDAR-based remote sensing depends fundamentally on the quality of the acquired point clouds. Existing studies commonly assess LiDAR data in terms of geometric accuracy, precision, completeness, point density, and consistency across repeated observations. These properties are influenced by ranging uncertainty, scanning distance, incidence angle, pulse frequency, angular resolution, and surface reflectance. Hartmann and Alkhatib [21] modeled spatially varying uncertainty in laser-scanning measurements, while Algadhi et al. [22] demonstrated that scanning distance and incidence angle directly affect the ability of terrestrial laser scanners to capture local surface structures. Verhelst et al. [23] further showed that different pulse-frequency settings can alter point density, completeness, and the accuracy of structural measurements. These studies indicate that LiDAR point clouds are affected by nonuniform acquisition uncertainty and therefore require systematic quality assessment before subsequent interpretation.

In practical surveying and mapping applications, LiDAR data are also affected by real environmental and operational interference. Platform motion, positioning errors, object occlusion, vegetation movement, reflective surfaces, and outdoor noise may introduce outliers, missing observations, local deformation, or misalignment between consecutive scans. For example, mobile laser scanning under forest canopies is affected by both vegetation occlusion and degraded GNSS/INS positioning, which can reduce point-cloud alignment quality [24]. Outdoor LiDAR scenes also exhibit substantial range-dependent variations in point spacing and noise distribution, making fixed-parameter filtering unsuitable for all spatial regions [25]. Accordingly, calibration, registration, denoising, outlier

removal, and uncertainty compensation remain essential for recovering accurate geometric information in real surveying environments.

The above research primarily addresses errors introduced by sensors, acquisition configurations, positioning systems, and natural environments, with the objective of restoring a point cloud that more faithfully represents the physical world. However, the increasing use of end-to-end learning introduces an additional robustness concern: even geometrically bounded input variations may be amplified by a perception model and cause unstable predictions. Adversarial robustness evaluation therefore complements conventional LiDAR quality assessment by examining model sensitivity after data acquisition and preprocessing. From this perspective, SAGE provides controlled digital perturbations for identifying decision-sensitive geometric structures and evaluating whether the corresponding vulnerabilities are shared across heterogeneous point-cloud classifiers and LiDAR detectors, thereby supporting robustness diagnosis and future defensive model development.

2.2. Adversarial Attacks Against Point Cloud Classification

Point cloud classification is a fundamental task in 3D perception. Early representative models, such as PointNet [26], directly process unordered point sets using shared multilayer perceptrons and symmetric aggregation. PointNet++ [27] further introduces hierarchical local grouping to capture fine-grained geometric structures. Subsequent models improve point cloud representation from different perspectives. DGCNN [28] constructs dynamic graph neighborhoods and extracts edge features through graph convolution. PointCNN [29], PointConv [30], and CurveNet [31] design more expressive local operators to model irregular 3D geometry. Transformer-based architectures, such as PCT [32] and Point Transformer [33], further exploit long-range dependencies among points. Although these models perform well on point-cloud classification, their diverse feature extraction mechanisms rely on different decision evidence, making transferable adversarial attacks across classifiers highly challenging.

Adversarial attacks against point cloud classifiers can be broadly divided into white-box and black-box settings. In the white-box setting, the attacker has access to the parameters or gradients of the target model and can directly optimize adversarial perturbations against the target classifier. 3D-Adv [4] first demonstrated that point cloud classifiers can be misled by shifting, adding, or removing points, revealing the vulnerability of point-based networks. Subsequent studies introduced geometric constraints to improve the plausibility of adversarial point clouds. KNN [34] exploits local neighborhood structures, while GeoA^3 [5] incorporates geometry-aware regularization to preserve natural object shapes. SI-Adv [35] constrains perturbations along shape-invariant directions, and HiT-Adv [6] hides adversarial deformations in visually insensitive surface regions. These methods provide valuable insights into the geometric sensitivity of point cloud classifiers. However, their reliance on gradients limits their practicality in deployed perception systems.

Since model parameters and gradients are usually unavailable in real-world remote sensing and LiDAR perception systems, recent research has increasingly shifted toward black-box attacks. Among these attacks, transfer-based methods are an important direction because adversarial examples generated on a surrogate model can be directly applied to unknown target models without repeated queries. AdvPC [11] explores transferable adversarial point clouds through a generative strategy, showing that surrogate-based perturbations can affect unseen classifiers. PF-Attack [12] improves transferability by exploiting point-wise frequency information. CFG [14] introduces critical feature guidance to align adversarial updates with more transferable feature cues across different backbones. In a stricter setting, NoPain [16] studies no-box point cloud attacks through an optimal-transport-based formulation, while OPS [15] enhances transferability by expanding the

hypothesis space. These studies have significantly advanced black-box point cloud attacks. However, existing black-box attacks often do not sufficiently distinguish the discriminative roles of critical model features, which are essential for improving the transferability of adversarial samples.

2.3. Adversarial Attacks Against LiDAR 3D Object Detection

LiDAR-based 3D object detection is a core component of modern remote sensing perception, autonomous driving, and intelligent transportation systems. According to the point cloud representation used by the detector, existing methods can be roughly categorized into point-based, voxel-based, pillar-based, and point-voxel hybrid detectors. Point-based methods, such as PointRCNN [36], directly process raw point clouds and preserve fine-grained geometric information. Voxel-based methods, including VoxelNet [37] and SECOND [38], convert irregular point clouds into sparse voxel representations to improve computational efficiency. Pillar-based methods, such as PointPillars [39], further simplify 3D voxel structures into vertical pillars, enabling efficient pseudo-image feature extraction. Hybrid detectors, such as PartA^2 [40] and PV-RCNN [41], combine voxel-level semantic representation with point-level refinement to improve detection accuracy. These architectural differences make adversarial attacks on LiDAR detection more challenging than attacks on classification, because the perturbations must remain effective after object points are embedded into full scenes and processed by different detection pipelines.

Recent studies have begun to investigate adversarial attacks against LiDAR 3D object detectors. Wang et al. [9] studied adversarial point perturbations for 3D object detection and optimized perturbations to suppress detection confidence. SlowLiDAR [42] explored another attack objective by increasing the inference latency of LiDAR-based detection systems. Efficient Attack [43] improves the efficiency of adversarial optimization for 3D detection, while GSVA [44] focuses on attacking sparse voxel representations and addresses the difficulty of perturbing voxelized detection pipelines. In addition, recent surveys and studies [45,46] indicate that adversarial threats against LiDAR detectors have become increasingly important for safety-critical perception systems. However, many existing detection attacks are designed under white-box or detector-specific assumptions, and their transferability across heterogeneous detectors remains limited. Compared with classification attacks, transferable attacks for full-scene LiDAR detection must simultaneously consider object-level saliency, geometric fidelity, and cross-detector consistency. Motivated by these limitations, this work studies saliency- and geometry-enhanced transfer attacks across both point cloud classification and LiDAR 3D object detection, aiming to provide a more comprehensive robustness evaluation for 3D remote sensing perception systems.

3. Methodology

We present **SAGE**, a saliency and geometry enhanced transfer attack for 3D point cloud perception. As shown in Figure 3, SAGE can be applied to both object-level classification and scene-level 3D detection. Given a clean point cloud or an object point set extracted from a LiDAR scene, SAGE performs an iterative attack on a surrogate perception model. In each iteration, it identifies regions that are both task-sensitive and geometrically reliable, and then concentrates perturbations on these regions while explicitly controlling geometric distortion. The method is organized around two coupled decisions: where to perturb and how much to perturb. These decisions are optimized under a unified constrained formulation. Algorithm 1 summarizes the overall procedure.

Algorithm 1 Saliency & Geometry Enhanced Attack (SAGE)

Input: Clean point set X, task annotation $\mathcal{Y}$, surrogate perception model F_ϕ; steps T; budgets (ϵ, ρ); saliency parameters (γ, μ); gain parameters $(\alpha, g_{\max})$; patch parameters (p, k); guidance weight β.

Output: Adversarial point set $X_{\text{adv}}^{(T)}$.

Initialize $U^{(0)} \leftarrow 0$, $s^{(0)} \leftarrow 0$, $X_{\text{adv}}^{(0)} \leftarrow X$.
for $t = 0$ to $T - 1$ **do**
  Compute the unified perception loss $\mathcal{L}_{\text{per}}$ by Equation (2).
  Compute $s_{\text{attn}}^{(t)}$ by Equation (6) and $s_{\text{tan}}^{(t)}$ by Equations (7)–(9).
  Obtain the unified saliency map $s^{(t)}$ by Equations (10) and (11).
  Construct the mask $m^{(t)}$ using top-p seeds and KNN expansion; compute the gain $g^{(t)}$ by Equation (13).
  Form $\Delta^{(t)} \leftarrow g^{(t)} \odot U^{(t)}$ and $X_{\text{adv}}^{(t)} \leftarrow X + \Delta^{(t)}$.
  Update $U^{(t+1)}$ by a mask-gated gradient step on Equation (14), with $m^{(t)}$ and $g^{(t)}$ fixed.
  Form $\Delta^{(t+1)} \leftarrow g^{(t)} \odot U^{(t+1)}$.
  Align $\Delta^{(t+1)}$ by Equation (17) and enforce Equation (18).
  Set $X_{\text{adv}}^{(t+1)} \leftarrow X + \Delta^{(t+1)}$.
end for
return $X_{\text{adv}}^{(T)}$.

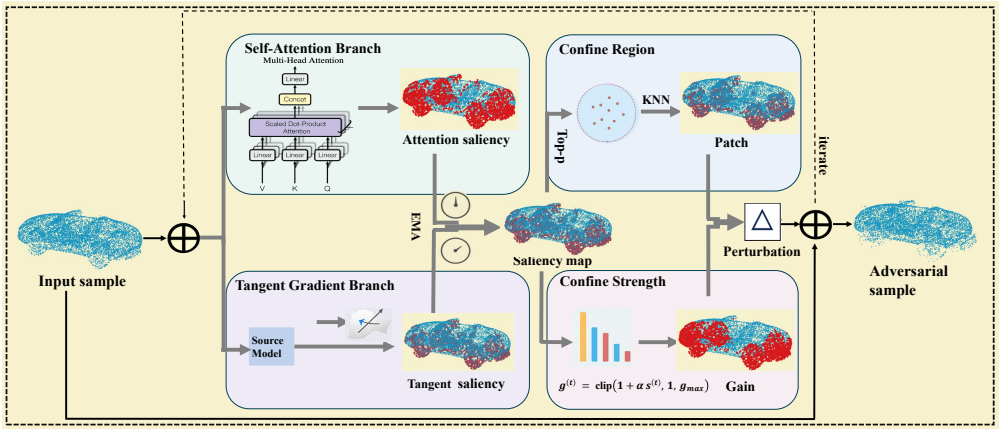


Figure 3. Overview of the SAGE pipeline. First, the Self-Attention Branch and the Tangent Gradient Branch produce complementary saliency signals, which are stabilized with EMA to form a unified saliency map. Next, the Top-p salient points are expanded via KNN into a compact patch to confine the perturbation region. Meanwhile, a gain is derived from the saliency map to confine the perturbation strength. Finally, after applying the perturbation and performing iterative updates, we obtain the adversarial sample. The red points denote the salient points.

3.1. Preliminaries

Let $X \in \mathbb{R}^{N\times 3}$ denote the point set to be attacked. For classification, X is an object-level point cloud with label y. For detection, X denotes the object points associated with a detected object in a LiDAR scene. After perturbation, the adversarial object points are pasted back into the original scene to form the adversarial scene $\mathcal{S}_{\text{adv}}$. Let F_ϕ denote a surrogate 3D perception model, which can be either a point cloud classifier or a LiDAR-based 3D detector. We generate an adversarial point cloud X_{adv} by adding a perturbation Δ to the clean point set X:

$$X_{\text{adv}} = X + \Delta, \qquad \Delta \in \mathbb{R}^{N\times 3}. \tag{1}$$

To support both classification and detection, we define a unified perception objective:

$$\mathcal{L}_{\text{per}} = \begin{cases} \mathcal{L}_{\text{cls}}\big(F_\phi(X_{\text{adv}}), y\big), & \text{classification,} \\ \mathcal{L}_{\text{det}}\big(F_\phi(\mathcal{S}_{\text{adv}}), o\big), & \text{detection,} \end{cases} \tag{2}$$

where o denotes the attacked object that is successfully detected in the clean scene.

For untargeted classification attacks, we use the negative cross entropy:

$$\mathcal{L}_{\text{cls}}\big(F_\phi(X_{\text{adv}}), y\big) = -\,\text{CE}\big(F_\phi(X_{\text{adv}}), y\big). \tag{3}$$

Minimizing Equation (3) increases the classification error of the surrogate classifier.

For untargeted detection attacks, we directly optimize the detection score of the attacked object:

$$\mathcal{L}_{\text{det}}\big(F_\phi(\mathcal{S}_{\text{adv}}), o\big) = q_\phi(\mathcal{S}_{\text{adv}}, o), \tag{4}$$

where $q_\phi(\mathcal{S}_{\text{adv}}, o)$ denotes the confidence score assigned by the detector to the attacked object o in the adversarial scene. If the attacked object is no longer detected, $q_\phi(\mathcal{S}_{\text{adv}}, o)$ becomes low.

Under this unified formulation, adversarial example generation is written as

$$\min_{X_{\text{adv}}} \mathcal{L}_{\text{per}} \quad \text{s.t.} \quad D(X_{\text{adv}}, X) \le \epsilon, \tag{5}$$

where $D(\cdot)$ measures geometric distortion and ϵ is the imperceptibility budget. SAGE follows Equation (5) and instantiates $D(\cdot)$ with explicit ℓ_2 constraints in Section 3.4. Before introducing the constrained solution, we first describe how SAGE estimates saliency and then converts saliency into two control signals: a patch mask for region selection and a gain vector for update allocation.

3.2. Unified Saliency Estimation

At iteration t, denote the current adversarial point cloud by $X_{\text{adv}}^{(t)}$. SAGE computes a saliency vector $s^{(t)} \in \mathbb{R}^N$ by combining coordinate self-attention and tangent-plane gradients. The attention branch captures structurally important regions from point relationships, while the tangent branch identifies task-sensitive regions under local geometric constraints. Their fusion favors points that are both structurally representative and prediction-sensitive, reducing overfitting to surrogate-specific gradients. An exponential moving average is further used to stabilize the saliency across iterations.

3.2.1. Coordinate Self-Attention Saliency

We map point coordinates to point-wise feature vectors through a linear projection and apply a lightweight self-attention module to obtain $H^{(t)} \in \mathbb{R}^{N\times d}$. This branch captures spatial dependencies among points and highlights regions that contribute strongly to the point-cloud structure. The attention-based saliency is computed by max-normalized feature magnitudes:

$$s_{\text{attn}}^{(t)} = \frac{\|H^{(t)}\|_2}{\max \|H^{(t)}\|_2}. \tag{6}$$

3.2.2. Tangent-Plane Gradient Saliency

Using the unified perception loss in Equation (2), we compute the coordinate gradient at the current iterate:

$$G^{(t)} = \nabla_{X_{\text{adv}}^{(t)}} \mathcal{L}_{\text{per}} \in \mathbb{R}^{N\times 3}. \tag{7}$$

Let $N_X \in \mathbb{R}^{N\times 3}$ be the estimated normal field of the clean point set X. We remove the normal component of the gradient to obtain a tangent-plane signal:

$$G_{\perp}^{(t)} = G^{(t)} - \left(G^{(t)} \cdot N_X\right) N_X. \tag{8}$$

This projection retains task-sensitive variations along the local surface and suppresses excessive off-surface responses. The tangent saliency is then defined by max-normalized tangent magnitudes:

$$s_{\text{tan}}^{(t)} = \frac{\|G_{\perp}^{(t)}\|_2}{\max \|G_{\perp}^{(t)}\|_2}. \tag{9}$$

3.2.3. Fusion and Temporal Stabilization

The two branches provide complementary information. Attention saliency emphasizes stable geometric structures but does not directly measure their influence on the prediction. Tangent saliency reflects task sensitivity but may contain surrogate-specific gradient responses. Their fusion strengthens regions supported by both structural importance and task sensitivity, making the selected regions more likely to remain effective across different models. The two saliency signals are fused as follows:

$$\tilde{s}^{(t)} = s_{\text{attn}}^{(t)} + \gamma\, s_{\text{tan}}^{(t)}, \qquad \tilde{s}^{(t)} \leftarrow \frac{\tilde{s}^{(t)}}{\max \tilde{s}^{(t)}}. \tag{10}$$

To suppress iteration-specific noise, we maintain an exponential moving average:

$$s^{(t)} = \mu\, s^{(t-1)} + (1-\mu)\, \tilde{s}^{(t)}. \tag{11}$$

The parameter γ controls the contribution of the tangent-plane signal, and μ is the EMA decay. The final saliency map at iteration t is $s^{(t)}$.

3.3. Salient Patch Mask and Gain Allocation

Given the unified saliency $s^{(t)}$, SAGE constructs two control signals: a compact patch mask that specifies where to update and a gain vector that specifies how strongly each point is updated. The mask localizes perturbations to salient regions, while the gain redistributes the perturbation strength according to point-wise saliency.

3.3.1. Salient Patch Mask

We first select the p points with the largest saliency scores as seeds and expand each seed to its k-NN neighborhood. The union of these neighborhoods forms a coherent salient patch $\mathcal{P}^{(t)}$. We encode $\mathcal{P}^{(t)}$ as a binary mask $m^{(t)} \in \{0,1\}^N$:

$$m_j^{(t)} = \begin{cases} 1, & j \in \mathcal{P}^{(t)}, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

During optimization, $m^{(t)}$ gates the gradient update, thereby concentrating perturbations on compact salient regions and avoiding unnecessary changes to less relevant points.

3.3.2. Gain Allocation

In parallel, we derive a point-wise gain vector from the saliency map:

$$g^{(t)} = \text{clip}\left(1 + \alpha\, s^{(t)}, 1, g_{\max}\right), \tag{13}$$

where α controls the amplification strength and $g_{\max}$ prevents excessive local distortion. High-saliency points receive stronger updates, while low-saliency points remain close to the baseline update strength. The clipping range $[1, g_{\max}]$ improves optimization stability and reduces the influence of transient saliency peaks.

3.4. Generation of Adversarial Examples

We combine the above controls to generate X_{adv}. The optimization uses a composite objective, while explicit ℓ_2 budgets are enforced throughout the iterative process. Mask gating and gain modulation are applied at every iteration but serve different roles. The mask determines where gradients are applied, and the gain determines the relative update strength of each point.

3.4.1. Objective

We optimize the following composite loss:

$$\mathcal{L} = \mathcal{L}_{\text{per}} + \mathcal{L}_{\text{cd}} + \mathcal{L}_{\text{SAGE}}, \tag{14}$$

where $\mathcal{L}_{\text{per}}$ is the task-specific perception loss defined in Equation (2), and $\mathcal{L}_{\text{cd}}(X_{\text{adv}}, X) = \text{CD}(X_{\text{adv}}, X)$ preserves the global shape. The SAGE guidance term is defined as

$$\mathcal{L}_{\text{SAGE}}(X_{\text{adv}}; X) = \beta \parallel W(X) \odot \Phi_\ell(X_{\text{adv}}) \parallel_1. \tag{15}$$

The feature tensor $\Phi_\ell(\cdot)$ denotes the last shared intermediate feature tensor before the task-specific prediction head. For classification, we use the point-wise feature before global aggregation; for detection, we use the backbone feature before the classification and regression heads. The importance weight $W(X)$ is obtained from the absolute gradient of $\mathcal{L}_{\text{per}}$ with respect to $\Phi_\ell(X)$, followed by channel-wise aggregation and max normalization. This choice balances low-level geometric information and task-specific semantics. The sensitivity to β is analyzed in Section 4.5.

3.4.2. Gain-Modulated Perturbation and Mask-Gated Update

We maintain a base offset field $U^{(t)} \in \mathbb{R}^{N\times 3}$ and form the perturbation through gain modulation:

$$\Delta^{(t)} = g^{(t)} \odot U^{(t)}, \qquad X_{\text{adv}}^{(t)} = X + \Delta^{(t)}. \tag{16}$$

At each iteration, we update $U^{(t)}$ with a first-order optimizer on Equation (14). The update is gated by $m^{(t)}$, thereby restricting the gradient update primarily to the salient patch. The gain $g^{(t)}$ reallocates the effective perturbation magnitude across points in the selected region. Specifically, $g^{(t)}$ is computed from $s^{(t)}$ at the beginning of iteration t and remains fixed when producing $U^{(t+1)}$.

3.4.3. Geometry-Aware Alignment

To preserve local surface plausibility, we align the perturbation with the clean point surface:

$$\Delta^{(t)} \leftarrow \left(\Delta^{(t)} \cdot N_X\right) N_X. \tag{17}$$

This alignment limits off-surface drift and keeps the update consistent with local surface orientation.

3.4.4. Explicit ℓ_2 Constraints

Finally, we control imperceptibility by enforcing two ℓ_2 budgets on $\Delta^{(t)}$:

$$\|\Delta^{(t)}\|_{2,\infty} \leq \epsilon, \qquad \frac{1}{N}\sum_{j=1}^{N}\|\Delta_j^{(t)}\|_2 \leq \rho\,\epsilon, \tag{18}$$

where $\|\Delta^{(t)}\|_{2,\infty} = \max_{1\leq j\leq N}\|\Delta_j^{(t)}\|_2$. The per-point constraint prevents prominent outliers, while the mean constraint limits the overall geometric distortion. Together, these constraints limit both local outliers and overall displacement, encouraging geometrically plausible adversarial examples for object-level classification and scene-level detection.

4. Experiments

4.1. Experimental Setup

4.1.1. Classification Setup

Datasets. We evaluated transfer attacks on ModelNet40 and ScanObjectNN. ModelNet40 is a synthetic CAD benchmark containing 12,311 objects from 40 categories [17]. Following the settings of CFG [14] and PF-Attack [12], we used a subset of 250 samples from 10 categories for evaluation. ScanObjectNN is a real-world scanned dataset collected from RGB-D scenes and contains approximately 15,000 objects from 15 categories [18,47]. Compared with ModelNet40, ScanObjectNN includes clutter, partial observations, and sensor noise, making transfer attacks more challenging.

Models and baselines. We evaluated the proposed method on representative classification models, including PointNet, PointNet++ (MSG), PointNet++ (SSG), and DGCNN, as well as advanced architectures such as PointConv, PointCNN, CurveNet, PCT, PT, and Point-PN. We further evaluated robustness under multiple defense models, including SRS, SOR, CCN, and Point-AT. To demonstrate the effectiveness of the proposed method, we compared it with several strong attack baselines, including 3D-Adv, KNN, *GeoA*^{3}, AdvPC, PF-Attack, and CFG.

Evaluation metrics. We used the attack success rate (ASR) and the transferability score (TS) to measure attack strength. TS is defined as the average black-box ASR over all surrogate-target pairs, where PointNet, PointNet++ (SSG/MSG), and DGCNN are alternately used as surrogate models. We reported Chamfer Distance (CD), Hausdorff Distance (HD), and F-score to evaluate geometric quality.

Implementation details. Following previous works, we used perturbation budgets of $\epsilon \in \{0.18, 0.45\}$ and fixed β at 0.40. The parameter ρ was set to 0.57 for $\epsilon = 0.18$ and 0.25 for $\epsilon = 0.45$. The rationale for these values is discussed in the ablation study. All experiments were conducted on an NVIDIA GeForce RTX 5090 GPU.

4.1.2. Detection Setup

Datasets. We conducted detection experiments on KITTI [19] and nuScenes [20]. KITTI is a widely used autonomous driving benchmark with 3712 training samples, 3769 validation samples, and 7518 test samples. nuScenes is a large-scale multimodal autonomous driving dataset with more than 1000 driving scenes, including approximately 28,000, 6000, and 6000 annotated samples for training, validation, and testing, respectively.

Models and baselines. We evaluated the proposed method on four representative LiDAR-based 3D object detectors, including PartA^2, PointPillars, PV-RCNN, and SECOND. To verify the effectiveness of the proposed method, we compared it with two strong attack baselines, AdvPC and HiT-Adv.

Implementation details. We extracted only the points inside the successfully detected 3D boxes and applied coordinate perturbations. The 3D IoU thresholds for Car, Pedestrian,

and Cyclist were set to 0.7, 0.5, and 0.5, respectively, ensuring reliable object matching before and after the attack. Furthermore, $\epsilon = 0.18$ and $\epsilon = 0.45$ correspond to maximum per-point displacements of 0.18 m and 0.45 m. The smaller budget represents mild local coordinate variation, whereas $\epsilon = 0.45$ is used as a strong stress-test setting rather than a typical real-world LiDAR noise level. These settings preserve the overall object structure and provide controlled perturbation budgets for LiDAR robustness evaluation. For the main experiments, results are averaged over three runs with random seeds 0, 42, and 2026.

4.2. Classification Task

4.2.1. Transferability on ModelNet40

Tables 1 and 2 show that SAGE consistently surpasses the compared transfer-based attack baselines on ModelNet40. The improvement is not uniform across architectures: we observe larger gains on targets with relatively concentrated decision evidence, where the salient patch discovered by SAGE is more likely to remain transferable across models. For example, on PointCNN under $\epsilon = 0.18$, SAGE improves black-box ASR from 32.7% to 40.0%. A similar trend can also be observed on PointNet++(SSG), which indicates that saliency-guided compact perturbation is particularly beneficial when the downstream architecture relies on stable local geometric cues.

Table 1. Attack transferability across classical point cloud models on ModelNet40. Attack success rate (%).

Source	Attack	$\epsilon = 0.18$				$\epsilon = 0.45$			
		PointNet	PointNet++ (MSG)	PointNet++ (SSG)	DGCNN	PointNet	PointNet++ (MSG)	PointNet++ (SSG)	DGCNN
PointNet	3D-Adv [4]	100.0	8.4	10.4	6.8	100.0	8.8	9.6	8.0
	KNN [34]	100.0	9.6	10.8	6.0	100.0	9.6	8.4	6.4
	GeoA^3 [5]	100.0	20.0	19.6	7.2	100.0	23.6	20.8	7.2
	AdvPC [11]	98.8	20.4	27.6	22.4	98.8	18.0	26.8	20.4
	PF-Attack [12]	100.0	49.6	61.5	24.8	100.0	59.3	64.8	28.8
	CFG [14]	100.0	72.6	80.6	39.3	100.0	84.7	88.0	49.3
	SAGE (Ours)	**100.0**	**78.4**	**85.6**	**43.6**	**100.0**	**88.4**	**91.6**	**56.2**
PointNet++ (MSG)	3D-Adv [4]	6.8	100.0	28.4	11.2	7.2	100.0	29.2	11.2
	KNN [34]	6.4	100.0	22.0	8.8	6.4	100.0	23.2	7.6
	GeoA^3 [5]	4.4	100.0	14.4	6.4	4.4	100.0	13.6	6.0
	AdvPC [11]	13.2	97.2	54.8	**39.6**	18.4	98.0	58.0	**39.2**
	PF-Attack [12]	17.2	100.0	67.0	24.3	19.2	100.0	75.0	27.5
	CFG [14]	17.6	100.0	72.9	25.3	36.6	100.0	83.5	34.7
	SAGE (Ours)	**20.5**	**100.0**	**75.9**	26.1	**42.6**	**100.0**	**90.8**	35.8
PointNet++ (SSG)	3D-Adv [4]	7.6	9.6	100.0	6.0	7.2	10.4	100.0	7.2
	KNN [34]	6.4	9.2	100.0	6.4	6.8	7.6	100.0	6.0
	GeoA^3 [5]	5.2	10.4	100.0	2.2	4.8	9.2	100.0	4.0
	AdvPC [11]	12.0	27.2	99.2	22.8	14.0	30.8	99.2	27.6
	PF-Attack [12]	13.9	47.4	100.0	19.6	15.7	56.9	100.0	23.0
	CFG [14]	18.1	**84.7**	100.0	38.3	37.7	86.4	100.0	45.9
	SAGE (Ours)	**18.9**	81.9	**100.0**	**39.8**	**37.8**	**88.8**	**100.0**	**47.4**
DGCNN	3D-Adv [4]	9.2	11.2	31.2	100.0	9.6	12.8	30.4	100.0
	KNN [34]	7.2	9.6	14.0	99.6	6.8	10.0	11.2	99.6
	GeoA^3 [5]	4.4	27.2	27.6	100.0	4.4	26.8	25.6	100.0
	AdvPC [11]	19.6	46.0	64.4	94.8	32.8	48.8	64.4	97.2
	PF-Attack [12]	21.3	60.9	74.8	100.0	26.1	79.7	85.8	100.0
	CFG [14]	29.3	70.1	88.9	100.0	39.0	86.0	91.8	100.0
	SAGE (Ours)	**29.5**	**72.5**	**89.8**	**100.0**	**43.5**	**86.4**	**92.8**	**100.0**

Table 2. Attack transferability across state-of-the-art point cloud models on ModelNet40. Attack success rate (%). Source model: PointNet.

Attack	$\epsilon = 0.18$						$\epsilon = 0.45$					
	Point Conv	Point CNN	Curve Net	PCT	PT	Point -PN	Point Conv	Point CNN	Curve Net	PCT	PT	Point -PN
3D-Adv [4]	0.8	1.0	1.8	0.0	0.2	1.2	0.8	0.1	1.8	0.0	1.0	0.7
GeoA^3 [5]	10.3	2.1	5.5	3.3	18.3	5.5	14.9	5.5	6.7	7.5	19.6	10.2
PF-Attack [12]	50.3	29.5	22.7	23.4	49.9	26.1	53.2	33.4	24.8	28.7	52.3	40.8
CFG [14]	68.1	32.7	29.2	30.4	65.6	42.5	81.3	43.4	47.5	48.6	84.7	61.5
SAGE (Ours)	**74.8**	**40.0**	**31.2**	**30.4**	**75.2**	**45.6**	**83.5**	**47.4**	**56.2**	**52.6**	**88.0**	**71.1**

4.2.2. Global Transferability Score

We further evaluated the overall black-box transferability on ModelNet40 using TS. As shown in Table 3, SAGE achieves the highest TS under both perturbation budgets. Its improvement over the compared methods is more pronounced under the larger budget, indicating that saliency-guided perturbation allocation makes more effective use of the available budget without sacrificing cross-architecture transferability.

Table 3. Transferability score TS (%) on ModelNet40 under different perturbation budgets.

ϵ	3D-Adv	KNN	*GeoA*^{3}	AdvPC	PF-Attack	CFG	SAGE (Ours)
0.18	12.2	9.7	12.4	30.8	40.2	53.1	**55.2**
0.45	12.6	9.2	12.5	33.3	46.8	63.6	**66.8**

4.2.3. Transferability on ScanObjectNN

We further evaluated our method on ScanObjectNN, which is a more realistic benchmark because of clutter, incompleteness, and sensor noise. As shown in Tables 4 and 5, SAGE consistently outperforms previous transfer-based attacks on both classical and state-of-the-art targets. Notably, the performance trend on ScanObjectNN remains largely consistent with that on ModelNet40, which suggests that the proposed saliency–geometry coupling is not restricted to CAD data and remains effective on real scanned objects.

Table 4. Transfer-based attacks on classical models on ScanObjectNN. Attack success rate (%). Source model: PointNet.

Attack	$\epsilon = 0.18$				$\epsilon = 0.45$			
	PointNet	PointNet++ (MSG)	PointNet++ (SSG)	DGCNN	PointNet	PointNet++ (MSG)	PointNet++ (SSG)	DGCNN
3D-Adv [4]	100.0	7.5	7.5	9.0	100.0	7.3	7.5	9.0
GeoA^3 [5]	100.0	18.3	20.1	18.2	100.0	17.8	19.5	17.8
PF-Attack [12]	100.0	50.4	54.6	40.3	100.0	51.4	54.2	40.8
CFG [14]	100.0	66.5	68.0	58.3	100.0	69.1	70.3	58.4
SAGE (Ours)	**100.0**	**74.4**	**75.2**	**58.8**	**100.0**	**75.8**	**76.0**	**65.6**

Table 5. Transfer-based attack performance on ScanObjectNN. Attack success rate (%). Source model: PointNet.

Attack	$\epsilon = 0.18$						$\epsilon = 0.45$					
	Point Conv	Point CNN	Curve Net	PCT	PT	Point -PN	Point Conv	Point CNN	Curve Net	PCT	PT	Point -PN
3D-Adv [4]	9.3	10.6	10.0	8.6	12.8	27.3	9.2	2.1	10.0	9.0	13.2	27.2
GeoA^3 [5]	21.7	20.6	18.8	17.3	27.3	40.4	21.4	20.8	18.3	16.7	27.0	41.9
PF-Attack [12]	50.1	45.0	47.1	45.6	51.9	68.8	47.9	45.5	46.4	47.1	51.8	69.5
CFG [14]	61.7	50.9	64.5	**63.6**	67.0	80.0	63.8	53.7	66.5	66.8	68.9	81.8
SAGE (Ours)	**75.6**	**67.6**	**65.2**	60.8	**71.6**	**84.0**	**84.8**	**74.8**	**72.0**	**80.4**	**79.6**	**88.4**

4.2.4. Attacking Defended Classifiers

To evaluate the robustness of the proposed attack beyond standard undefended classifiers, we additionally tested SAGE against several representative defense strategies. As shown in Tables 6 and 7, SAGE consistently achieves the highest or nearly the highest black-box ASR on defended models.

The evaluated defenses include Simple Random Sampling (SRS), Statistical Outlier Removal (SOR), the Convex Combination Network (CCN) [48], and an adversarially trained PointNet (AT) [49]. These results indicate that the perturbation regions identified by SAGE remain effective even after standard purification and robustness-enhancement procedures.

Table 6. Black-box transfer attacks against defenses on ModelNet40. Attack success rate (%). Source model: PointNet.

Attack	$\epsilon = 0.18$				$\epsilon = 0.45$			
	SRS	**SOR**	**CCN**	**AT**	**SRS**	**SOR**	**CCN**	**AT**
3D-Adv [4]	26.4	17.2	0.0	5.0	1.0	17.2	0.0	1.0
GeoA^3 [5]	58.3	43.8	11.3	3.8	53.8	40.8	11.7	3.8
PF-Attack [12]	55.7	42.4	30.0	10.4	61.1	33.8	36.5	14.4
CFG [14]	79.2	62.9	50.6	**22.6**	87.9	73.0	66.8	30.8
SAGE (Ours)	**82.0**	**66.4**	**58.4**	22.0	**88.0**	**73.5**	**73.9**	**30.9**

Table 7. Black-box transfer attacks against defenses on ScanObjectNN. Attack success rate (%). Source model: PointNet.

Attack	$\epsilon = 0.18$				$\epsilon = 0.45$			
	SRS	**SOR**	**CCN**	**AT**	**SRS**	**SOR**	**CCN**	**AT**
3D-Adv [4]	67.4	67.8	8.3	23.5	48.7	48.7	8.3	23.5
GeoA^3 [5]	77.6	59.4	23.4	27.5	77.2	59.0	22.8	26.9
PF-Attack [12]	60.1	50.4	52.3	41.6	56.8	50.4	53.4	41.0
CFG [14]	96.6	82.2	63.4	56.0	95.7	81.1	64.8	60.7
SAGE (Ours)	**97.0**	**83.0**	**68.0**	**57.2**	**95.8**	**82.0**	**71.2**	**71.6**

4.2.5. Running Time Comparison

We also compared the average adversarial generation time under the same experimental setting. As shown in Table 8, although SAGE is slightly slower than the simplest white-box attacks, it remains more efficient than the stronger transfer baselines PF-Attack and CFG while achieving better transfer performance. This result indicates that the additional saliency estimation and gain allocation in SAGE introduce only moderate computational overhead.

Table 8. Time comparison. The source model is PointNet. Running time is the average time (in minutes) required to generate one adversarial point cloud.

Attack Method	**3D-Adv**	***GeoA*^{3}**	**PF-Attack**	**CFG**	**SAGE (Ours)**
Running Time	**0.04**	0.05	0.10	0.07	0.06

4.2.6. Imperceptibility on ModelNet40

Table 9 reports the geometric discrepancy between clean and adversarial point clouds. Under both budgets, SAGE consistently achieves better imperceptibility than CFG [14], which further verifies that the proposed saliency-guided perturbation allocation improves transferability without introducing excessive geometric distortion.

Table 9. Imperceptibility on ModelNet40 with PointNet as surrogate. The arrows indicate the preferred direction: ↓ means lower is better, and ↑ means higher is better.

$\epsilon = 0.18$				
Method	CD(avg)↓	HD(avg)↓	F@0.010↑	F@0.020↑
CFG [14]	0.006850	0.031494	0.1004	0.2239
SAGE (Ours)	**0.006245**	**0.029601**	**0.1029**	**0.2291**
$\epsilon = 0.45$				
Method	CD(avg)↓	HD(avg)↓	F@0.010↑	F@0.020↑
CFG [14]	0.010866	0.166553	0.0949	0.2120
SAGE (Ours)	**0.008241**	**0.151758**	**0.1101**	**0.2363**

4.3. Detection Task

We next evaluated whether the transferability improvements achieved by SAGE on classification benchmarks extend to realistic LiDAR detection pipelines. Unlike point cloud

classification, object detection requires the adversarial object to remain effective after being inserted back into the full scene and transferred across different detector architectures. Therefore, this setting is more challenging and better reflects practical black-box threats.

4.3.1. Transferability on KITTI

As shown in Tables 10 and 11, we summarized the transfer results on KITTI. Following the standard KITTI evaluation protocol, we report results for three object categories, namely Car, Pedestrian, and Cyclist. The results show that SAGE consistently outperforms AdvPC and HiT-Adv under $\epsilon = 0.18$ in most source–target transfer settings, demonstrating its strong transferability under a relatively small perturbation budget. However, the improvement is not uniform across all cases. For example, under $\epsilon = 0.18$, SAGE is slightly lower than HiT-Adv in the PointPillars-to-SECOND transfer on Cyclist, indicating that sparse categories and certain detector pairs remain challenging. When the perturbation budget is increased to $\epsilon = 0.45$, the transfer ASR of SAGE exceeds 90% in several settings, further confirming the effectiveness of the proposed method.

Table 10. Transfer attack success rate (%) on KITTI under $\epsilon = 0.18$. For each class, the reported ASR is averaged over the Easy, Moderate, and Hard difficulty levels.

Source	Attack	PartA^2			PointPillars			PV-RCNN			SECOND		
		Car	Pedestrian	Cyclist	Car	Pedestrian	Cyclist	Car	Pedestrian	Cyclist	Car	Pedestrian	Cyclist
PartA^2	AdvPC [11]	14.31	27.22	11.74	9.25	28.52	6.90	9.66	31.11	13.85	10.05	26.54	11.56
	HiT-Adv [6]	7.15	20.53	7.30	6.59	25.95	9.35	6.10	21.63	8.08	6.67	21.00	7.52
	SAGE (Ours)	**24.71**	**71.54**	**19.39**	**10.49**	**28.71**	**10.29**	**20.93**	**68.55**	**29.84**	**15.09**	**53.84**	**13.58**
PointPillars	AdvPC [11]	25.95	25.13	14.41	43.55	31.28	24.67	23.71	25.58	14.39	28.31	25.16	14.68
	HiT-Adv [6]	34.54	51.34	36.30	35.24	51.38	33.81	31.21	42.49	37.16	32.54	42.57	**26.51**
	SAGE (Ours)	**45.38**	**53.39**	**37.99**	**57.52**	**52.99**	**36.26**	**40.07**	**44.46**	**40.56**	**45.71**	**46.10**	25.14
PV-RCNN	AdvPC [11]	12.56	26.85	11.21	9.55	30.06	6.08	8.99	29.12	13.13	9.74	25.68	11.19
	HiT-Adv [6]	7.04	21.28	4.63	6.50	26.98	5.98	5.85	22.90	5.57	6.38	23.83	4.40
	SAGE (Ours)	**27.92**	**67.68**	**21.53**	**9.60**	**30.57**	**7.66**	**23.70**	**63.29**	**26.58**	**17.55**	**48.07**	**15.23**
SECOND	AdvPC [11]	13.24	28.30	12.28	9.64	30.25	**12.90**	9.45	30.22	14.39	10.47	27.52	11.19
	HiT-Adv [6]	6.74	19.61	6.58	6.20	25.05	6.17	5.42	20.52	6.28	5.78	21.81	6.61
	SAGE (Ours)	**24.74**	**70.47**	**19.75**	**10.58**	**30.64**	10.17	**20.30**	**65.06**	**29.66**	**14.61**	**49.34**	**11.38**

Table 11. Transfer attack success rate (%) on KITTI under $\epsilon = 0.45$. For each class, the reported ASR is averaged over the Easy, Moderate, and Hard difficulty levels.

Source	Attack	PartA^2			PointPillars			PV-RCNN			SECOND		
		Car	Pedestrian	Cyclist	Car	Pedestrian	Cyclist	Car	Pedestrian	Cyclist	Car	Pedestrian	Cyclist
PartA^2	AdvPC [11]	14.15	28.19	11.39	9.35	28.58	12.71	9.60	29.48	13.64	10.23	26.49	11.74
	HiT-Adv [6]	11.57	51.50	13.88	10.26	52.92	13.83	10.49	52.45	15.86	10.83	51.41	14.13
	SAGE (Ours)	**89.33**	**98.93**	**86.65**	**48.11**	**94.16**	**65.42**	**87.42**	**98.16**	**90.05**	**81.59**	**96.31**	**83.30**
PointPillars	AdvPC [11]	25.69	24.76	14.41	43.46	31.09	24.30	23.73	25.23	16.52	28.23	25.62	14.31
	HiT-Adv [6]	95.53	89.76	85.23	76.68	86.71	75.33	92.82	88.87	84.09	92.72	85.92	77.80
	SAGE (Ours)	**95.60**	**94.21**	**88.08**	**90.06**	**94.67**	**88.41**	**94.30**	**94.26**	**88.83**	**93.77**	**94.35**	**86.42**
PV-RCNN	AdvPC [11]	12.30	25.62	11.92	9.45	29.74	12.34	9.03	28.71	14.36	9.74	26.77	10.83
	HiT-Adv [6]	11.66	45.34	16.01	10.63	49.58	16.07	10.84	46.99	15.14	10.83	46.51	14.13
	SAGE (Ours)	**91.32**	**99.04**	**87.90**	**50.86**	**94.48**	**65.05**	**89.60**	**98.33**	**90.96**	**84.88**	**95.73**	**83.49**
SECOND	AdvPC [11]	13.21	27.92	12.63	9.63	30.25	12.90	9.72	31.19	13.90	10.44	27.47	11.19
	HiT-Adv [6]	9.93	44.91	14.23	9.23	46.05	13.08	9.03	44.24	13.15	8.87	45.70	10.83
	SAGE (Ours)	**92.65**	**98.98**	**90.93**	**51.64**	**96.53**	**68.22**	**91.05**	**98.83**	**93.67**	**85.40**	**97.06**	**85.50**

To further evaluate the scene-level detection degradation, we use 3D mAP as a complementary metric, which is computed following the KITTI AP_{R40} protocol. As shown in Table 12, the adversarial examples generated by SAGE clearly reduce the mAP of different target detectors, further demonstrating the effectiveness of the proposed transferable attack in full-scene LiDAR detection.

In addition, we visualized adversarial examples generated on four detectors in Figures 4 and 5. The visual results show that SAGE achieves strong attack performance

while preserving visually plausible and geometry-consistent object structures. To further evaluate the scene-level visual effect of SAGE in realistic LiDAR-based remote sensing perception, we pasted the generated adversarial objects back into their original full-frame scenes. As shown in Figure 6, the adversarial samples remain visually consistent with the surrounding scene after being reinserted into the full LiDAR frame, demonstrating the applicability of SAGE to realistic scene-level perception scenarios.

Table 12. Transfer results of 3D mAP (%) on KITTI. The downward arrows indicate the reduction in mAP after the attack.

Source	PartA^2		PointPillars		PV-RCNN		SECOND	
	Clean	Adv. (↓)	Clean	Adv. (↓)	Clean	Adv. (↓)	Clean	Adv. (↓)
PartA^2	88.80	39.01 (↓49.79)	78.61	61.01 (↓17.60)	87.01	37.19 (↓49.82)	83.68	45.81 (↓37.88)
PointPillars	88.80	78.63 (↓10.17)	78.61	68.56 (↓10.05)	87.01	76.91 (↓10.10)	83.68	73.06 (↓10.62)
PV-RCNN	88.80	38.49 (↓50.31)	78.61	60.60 (↓18.01)	87.01	38.17 (↓48.84)	83.68	46.77 (↓36.91)
SECOND	88.80	38.75 (↓50.06)	78.61	61.61 (↓17.00)	87.01	38.44 (↓48.57)	83.68	47.46 (↓36.22)

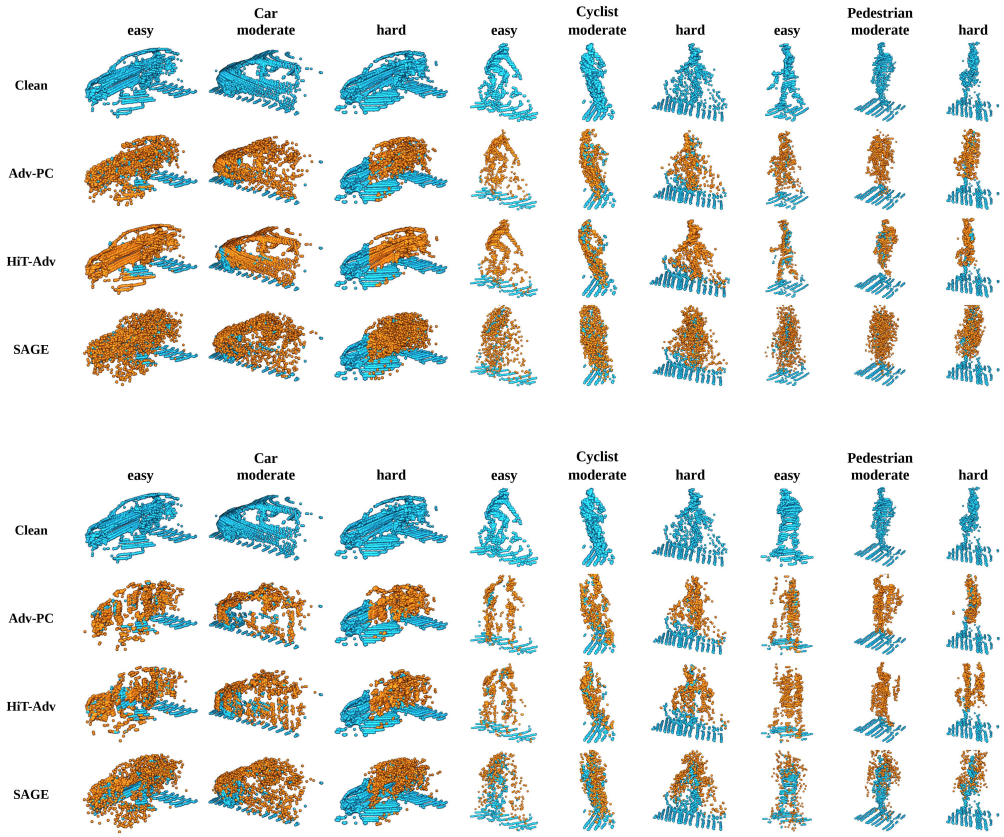


Figure 4. Visualization of adversarial objects on KITTI. Blue points denote clean objects and yellow points denote adversarial objects. The two rows correspond to adversarial samples generated from PartA^2 and PointPillars, respectively.

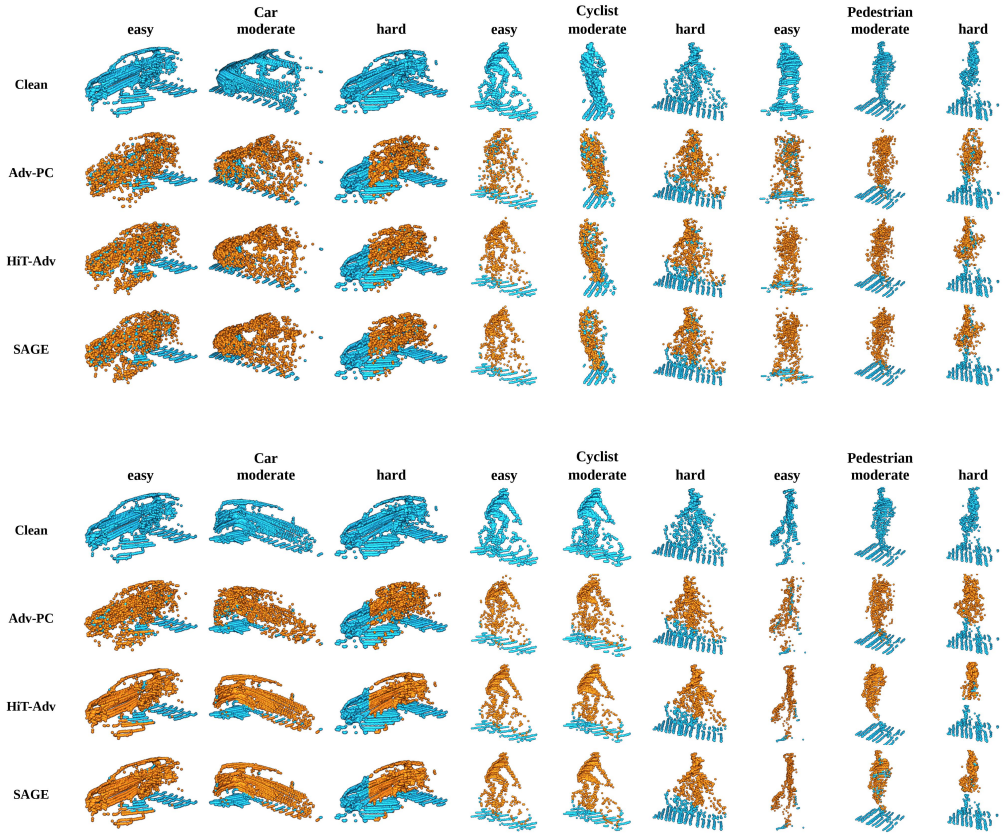


Figure 5. Visualization of adversarial objects on KITTI. Blue points denote clean objects and yellow points denote adversarial objects. The two rows correspond to adversarial samples generated from PV-RCNN and SECOND, respectively.

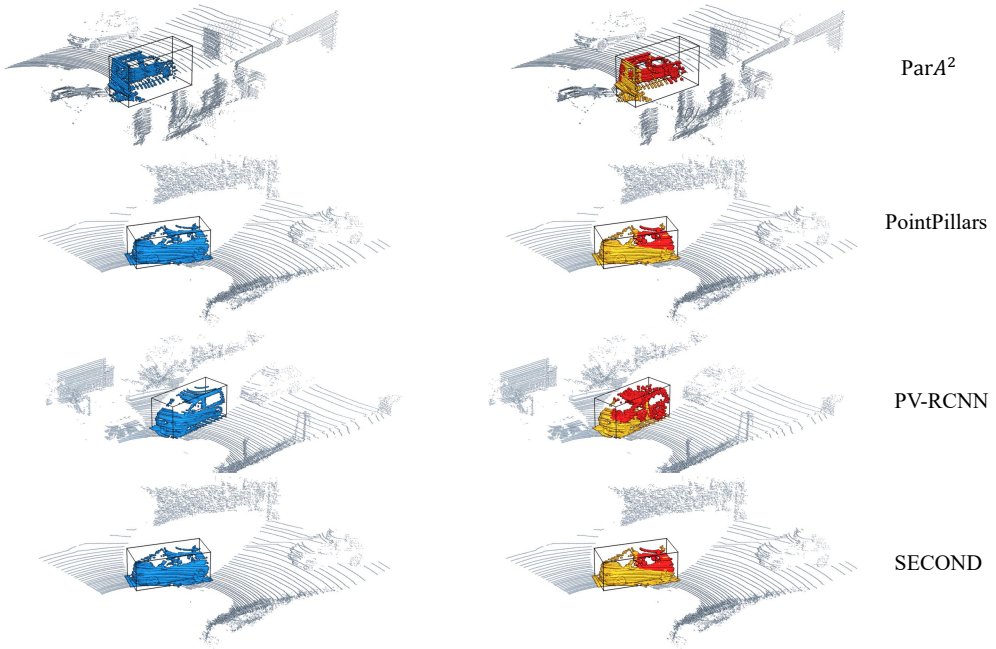


Figure 6. Full-scene visualization on KITTI. The left column shows original LiDAR scenes with clean vehicle points in blue, while the right column shows adversarial scenes with original points in yellow and perturbed points in red.

4.3.2. Transferability on nuScenes

To further demonstrate the generalization ability of the proposed method, we evaluated SAGE on nuScenes. Tables 13 and 14 report the transfer results on the nuScenes dataset. Compared with KITTI, nuScenes contains more diverse object categories and a wider range of scene layouts, making cross-detector transfer more challenging and leading to lower overall transfer success rates. Nevertheless, SAGE still achieves strong transferable attack performance, and in several source–target settings, its transfer results are substantially higher than those of the baseline methods. However, the gains are not uniform across all categories and detector pairs. For example, under $\epsilon = 0.45$, SAGE is less effective than HiT-Adv on Barrier in the PointPillars-to-PartA^2 transfer, suggesting that some scene-dependent categories remain challenging.

Table 13. Transfer attack success rate (%) on nuScenes under $\epsilon = 0.18$. Reported results are shown for Car, Pedestrian, Barrier, and Truck.

Source	Attack	PartA^2				PointPillars				PV-RCNN				SECOND			
		Car	Pedestrian	Barrier	Truck	Car	Pedestrian	Barrier	Truck	Car	Pedestrian	Barrier	Truck	Car	Pedestrian	Barrier	Truck
PartA^2	AdvPC [11]	11.90	31.97	45.62	10.95	2.77	19.17	31.65	1.86	8.64	33.43	29.75	8.30	3.61	32.28	**54.36**	3.70
	HiT-Adv [6]	11.32	**32.69**	34.25	11.19	3.25	**28.91**	33.15	1.24	9.51	**40.78**	37.40	8.36	4.35	30.63	34.33	3.40
	SAGE (Ours)	**13.13**	32.17	**46.09**	**17.08**	**3.70**	25.02	**33.88**	**2.48**	**10.21**	36.42	**38.50**	**11.74**	**4.84**	**37.00**	49.13	**6.48**
PointPillars	AdvPC [11]	13.26	42.05	32.45	15.55	11.48	46.10	61.06	13.62	11.25	43.24	27.91	12.11	7.62	40.99	55.84	8.33
	HiT-Adv [6]	13.61	44.64	44.23	9.03	6.86	43.58	57.38	4.02	11.15	45.11	36.46	10.18	6.73	40.25	54.36	4.63
	SAGE (Ours)	**18.21**	**49.32**	**45.70**	**16.95**	**17.61**	**54.57**	**71.61**	**19.50**	**13.82**	**49.00**	**36.83**	**14.59**	**11.18**	**47.72**	**70.89**	**11.11**
PV-RCNN	AdvPC [11]	11.43	31.90	44.94	12.01	2.70	29.24	33.93	2.48	9.16	44.58	29.24	10.03	3.50	42.76	54.50	3.40
	HiT-Adv [6]	11.32	33.65	33.30	7.22	2.81	27.96	30.97	1.24	8.79	42.59	**35.00**	9.82	4.13	32.79	33.11	4.01
	SAGE (Ours)	**13.05**	**34.28**	**49.36**	**14.59**	**3.48**	**31.47**	**36.92**	**3.55**	**11.05**	**45.85**	30.35	**15.30**	**5.32**	**46.25**	**54.73**	**5.86**
SECOND	AdvPC [11]	12.28	41.82	**54.06**	13.43	2.91	40.04	**43.56**	1.86	9.12	46.02	34.36	7.27	3.36	40.65	53.69	4.32
	HiT-Adv [6]	11.83	34.17	32.74	8.30	2.88	30.00	36.07	3.41	9.43	43.06	34.58	10.18	3.91	34.42	29.02	4.01
	SAGE (Ours)	**14.98**	**43.91**	50.49	**17.08**	**3.97**	**42.93**	43.04	**4.02**	**12.21**	**47.61**	**37.55**	**11.39**	**4.34**	**41.41**	**58.08**	**5.86**

Table 14. Transfer attack success rate (%) on nuScenes under $\epsilon = 0.45$. Reported results are shown for Car, Pedestrian, Barrier, and Truck.

Source	Attack	PartA^2				PointPillars				PV-RCNN				SECOND			
		Car	Pedestrian	Barrier	Truck	Car	Pedestrian	Barrier	Truck	Car	Pedestrian	Barrier	Truck	Car	Pedestrian	Barrier	Truck
PartA^2	AdvPC [11]	12.71	42.53	55.13	9.16	2.76	38.66	52.82	1.86	8.09	43.52	30.37	7.07	3.81	42.81	53.92	4.00
	HiT-Adv [6]	14.77	42.91	49.22	10.29	5.21	**41.24**	50.45	4.02	11.50	46.03	47.36	8.51	5.50	43.54	51.96	4.39
	SAGE (Ours)	**37.12**	**72.22**	**86.00**	**38.71**	**10.38**	38.58	**72.23**	**12.69**	**29.11**	**63.28**	**70.74**	**28.82**	**15.70**	**50.03**	**78.88**	**19.69**
PointPillars	AdvPC [11]	13.96	44.90	32.89	9.89	11.71	45.41	61.75	14.55	10.34	46.10	25.66	12.72	8.09	41.08	56.08	7.08
	HiT-Adv [6]	26.13	55.32	**84.31**	17.28	23.09	**85.43**	**89.33**	13.93	23.13	59.53	**78.80**	18.44	20.12	54.78	**84.84**	13.79
	SAGE (Ours)	**49.36**	**63.85**	43.44	**35.48**	**32.54**	83.67	87.84	**29.10**	**45.56**	**63.00**	43.94	**30.21**	**32.09**	**66.49**	61.04	**25.23**
PV-RCNN	AdvPC [11]	13.21	41.72	52.64	9.52	2.62	38.15	53.37	2.48	7.89	44.48	30.47	9.19	4.28	42.54	53.92	4.62
	HiT-Adv [6]	14.68	42.02	49.81	8.09	4.92	**38.22**	43.22	3.10	10.83	46.22	49.53	10.28	5.28	42.60	48.99	5.96
	SAGE (Ours)	**37.58**	**52.15**	**75.97**	**37.28**	**10.63**	32.14	**63.88**	**11.76**	**34.19**	**65.56**	**73.82**	**30.90**	**17.05**	**43.42**	**66.48**	**17.85**
SECOND	AdvPC [11]	12.58	44.45	55.70	11.36	2.95	39.03	52.28	1.86	9.05	46.67	33.84	6.71	3.99	41.01	54.32	3.08
	HiT-Adv [6]	13.71	42.02	44.25	6.62	4.62	36.08	45.57	4.64	10.99	47.48	44.16	9.93	5.94	39.72	46.28	4.08
	SAGE (Ours)	**39.28**	**66.25**	**64.19**	**36.92**	**13.40**	**51.54**	**69.69**	**14.86**	**35.12**	**63.76**	**56.37**	**33.33**	**22.03**	**66.22**	**78.75**	**20.31**

In addition, to further show that SAGE can generate adversarial objects with better geometric quality, we visualized the adversarial point clouds produced by different attacks. The visualization results are shown in Figure 7. Furthermore, to evaluate the scene-level visual effect of SAGE in realistic LiDAR-based remote sensing perception, we reinserted the generated adversarial objects into their original full-frame scenes on the nuScenes dataset. The results are shown in Figure 8.

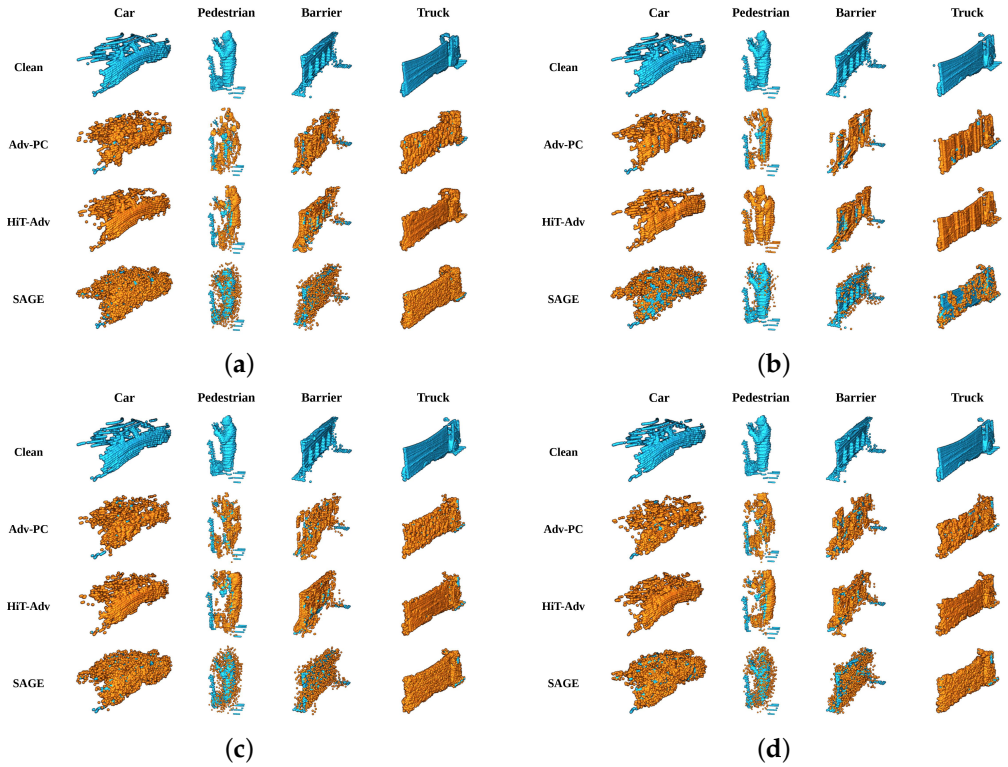


Figure 7. Visualization of adversarial objects on nuScenes. Blue points denote clean objects, and yellow points denote adversarial objects. The four images are arranged in the order of PartA^2, PointPillars, PV-RCNN, and SECOND. (**a**) PartA^2. (**b**) PointPillars. (**c**) PV-RCNN. (**d**) SECOND.

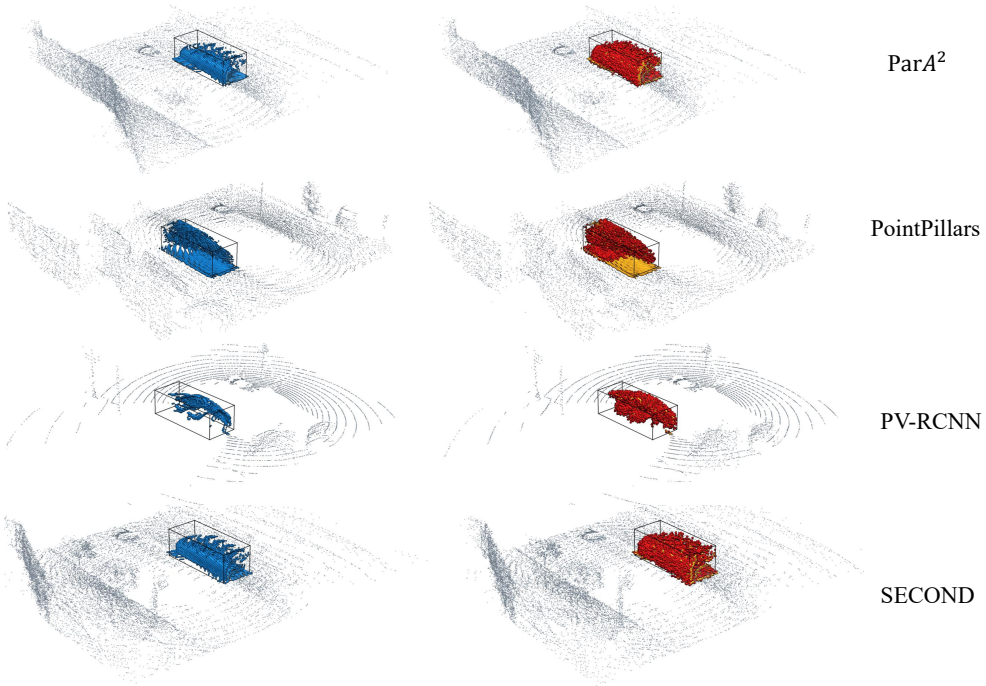


Figure 8. Full-scene visualization on nuScenes. The left column shows original LiDAR scenes with clean vehicle points in blue, while the right column shows adversarial scenes with original points in yellow and perturbed points in red.

4.4. Failure Case Analysis

Although SAGE achieves strong overall transferability, several failure cases remain. In classification, its advantage becomes smaller when the source and target models employ substantially different feature aggregation mechanisms. For example, SAGE is less

effective than AdvPC for the PointNet++(MSG) to DGCNN transfer in Table 1, and slightly underperforms CFG on PCT in Table 5. In detection, lower transferability is mainly observed for sparse and small objects, such as Cyclist and Truck, especially under the smaller perturbation budget, as shown in Tables 10 and 13. This is because limited object points make saliency estimation less stable. When only a few points are available, the local neighborhood and normal estimation become less reliable, and many points may receive similarly high saliency scores, making it harder to identify a compact and truly transferable critical region. Detector-specific voxelization and scene context may further weaken the perturbations after scene reinsertion. These results indicate that SAGE remains sensitive to large architectural differences and extremely sparse object observations.

4.5. Ablation Study

4.5.1. Ablation on Saliency and Confine Modules.

Figure 9 shows that, in the saliency module, removing either saliency signal decreases black-box ASR, and removing both signals causes a more pronounced drop. Furthermore, in the confine module, we observe an interesting phenomenon: when the gain is removed, using only the patch leads to worse performance than the setting without both patch and gain. The best results are achieved when both patch masking and gain allocation are enabled. We conjecture that, without gain, confining the perturbation region tends to produce updates that are simultaneously weak and spatially restricted, which can make the attack less effective.

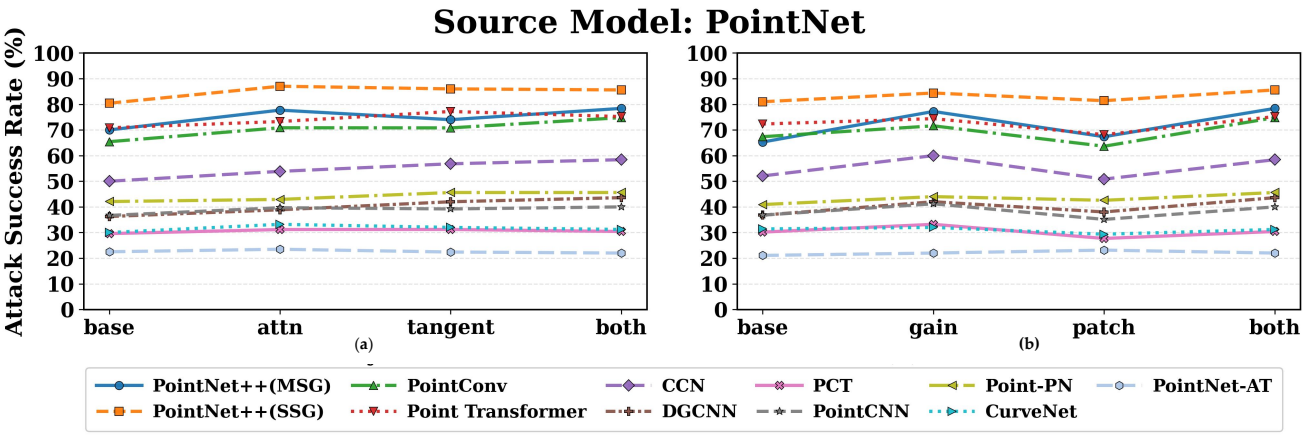


Figure 9. Ablation study of the two modules ($\epsilon = 0.18$) on ModelNet40 with PointNet as the surrogate. (**a**) Saliency module. (**b**) Control module.

To further examine the contribution of each component in LiDAR-based 3D object detection, we conducted an additional ablation study on KITTI under $\epsilon = 0.45$. PointPillars and SECOND were selected as representative source detectors, and each reported ASR is averaged over the four target detectors. As shown in Table 15, removing the saliency-estimation or perturbation-control components generally weakens the averaged transfer performance, indicating that these components are beneficial for detection transferability. Meanwhile, this ablation is not intended to exhaustively cover all source detectors; rather, it provides representative component-level evidence from two source detectors and shows that the proposed modules remain effective in the detection transfer setting.

Table 15. Detection-task ablation results on KITTI under $\epsilon = 0.45$.

Configuration	Source: PointPillars			Source: SECOND		
	Car	Pedestrian	Cyclist	Car	Pedestrian	Cyclist
w/o attention and tangent	85.81	73.15	64.78	66.26	76.84	63.12
w/o tangent	85.49	76.48	68.65	73.73	79.02	67.21
w/o attention	89.51	77.41	69.18	72.14	81.06	68.93
w/o patch and gain	87.57	72.60	62.98	68.19	78.13	62.45
w/o patch	88.56	76.15	65.45	73.24	83.71	66.08
w/o gain	87.58	75.89	68.46	71.25	80.43	69.07
SAGE	**93.43**	**94.37**	**87.94**	**80.19**	**97.85**	**84.58**

4.5.2. Parameter Selection.

We analyzed β and ρ on ModelNet40 with $\epsilon = 0.18$. We first varied β with fixed $\rho = 0.50$, using 0.4, 0.5, 0.6, 0.7, and 0.8. As shown in Figure 10a, the smallest value performs best in our setting. As β increases, the guidance term drives the update too strongly toward surrogate-specific evidence, and transferability on unseen architectures gradually weakens. With $\beta = 0.40$, the guidance term acts as a light bias rather than a hard constraint, so the perturbation still follows the classification objective while being nudged toward more reusable features.

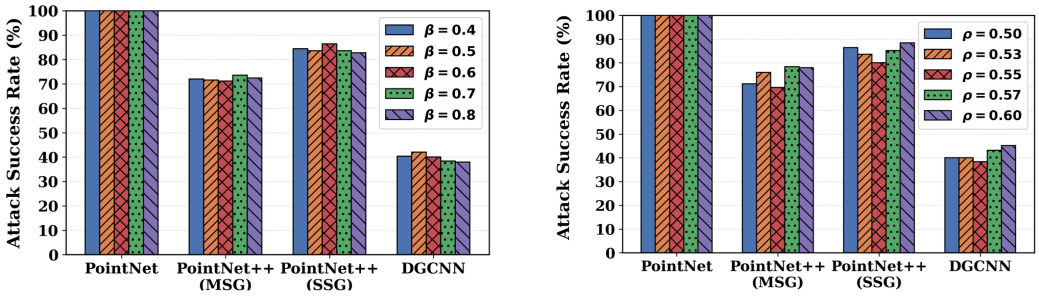


Figure 10. Parameter sensitivity analysis on ModelNet40 with PointNet as the source model under $\epsilon = 0.18$. (**a**) Sensitivity analysis of β with $\rho = 0.50$. (**b**) Sensitivity analysis of ρ with $\beta = 0.60$.

We then varied ρ with fixed $\beta = 0.60$, using 0.50, 0.53, 0.55, 0.57, and 0.60. As shown in Figure 10b, smaller ρ makes the mean constraint overly tight, so the salient patch cannot accumulate sufficient displacement under the same per-point bound. In contrast, larger ρ relaxes the mean constraint and makes the perturbation more likely to spread beyond the salient patch, thereby introducing unnecessary drift in low-impact regions. We chose $\rho = 0.57$ at $\epsilon = 0.18$ because it keeps the update concentrated while preserving sufficient patch strength. For $\epsilon = 0.45$, we decreased ρ to 0.25 to regulate average displacement and confine the increased per-point budget to localized regions.

4.5.3. EMA and Mean-ℓ_2 Constraint Design

We further analyzed two design choices, namely the global mean-ℓ_2 constraint and EMA smoothing. First, removing the global mean-ℓ_2 constraint while retaining the per-point ℓ_2 bound leads to increased geometric distortion: the average CD rises from 0.006547 to 0.007490. This observation is in line with our expectation and further verifies that the proposed global constraint is effective in enhancing geometric fidelity.

Second, EMA mainly contributes to transfer stability. As shown in Figure 11, removing EMA consistently reduces the black-box ASR on most target models under both perturbation budgets. Under $\epsilon = 0.18$, the variant with EMA achieves higher attack success rates on most targets, with clear gains on PointNet++(MSG), PointNet++(SSG), PointConv, Point Transformer, CCN, DGCNN, PCT, PointCNN, CurveNet, and PointNet-AT.

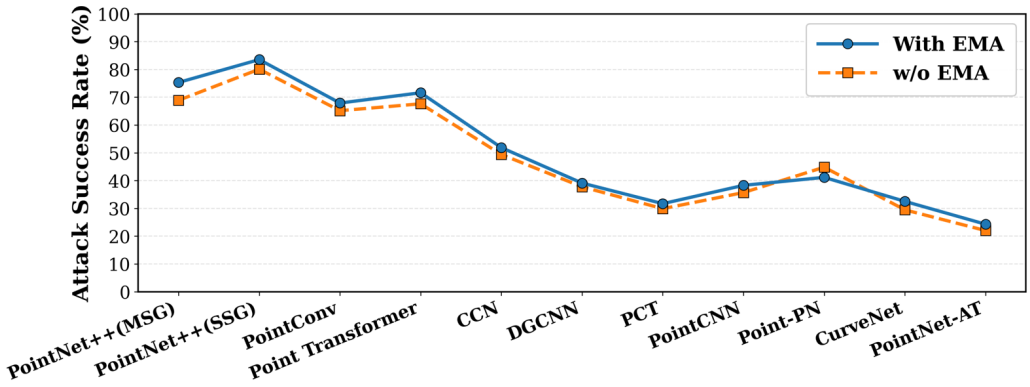


Figure 11. Effect of EMA on black-box transfer attack performance with PointNet as the source model under $\epsilon = 0.18$.

As shown in Figure 12, a similar trend is observed under $\epsilon = 0.45$, where EMA again improves performance on the majority of targets, especially on PointNet++(MSG), PointNet++(SSG), PointConv, Point Transformer, CCN, Point-PN, and CurveNet. These results indicate that EMA effectively suppresses saliency fluctuation across iterations and helps maintain a more coherent and transferable perturbation focus.

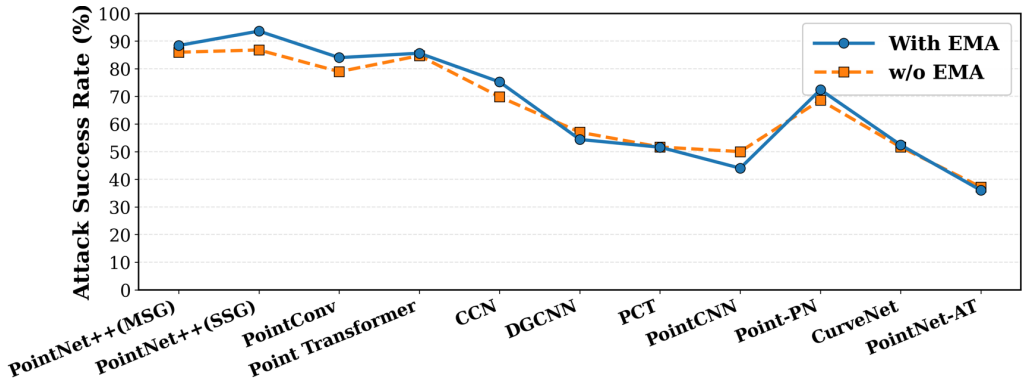


Figure 12. Effect of EMA on black-box transfer attack performance with PointNet as the source model under $\epsilon = 0.45$.

5. Discussion

The results show that the transferability of adversarial point clouds is closely related to whether the perturbed regions represent stable geometric evidence shared by different perception architectures. By combining structural saliency with task-sensitive gradients, SAGE concentrates perturbations on compact critical regions and avoids unnecessary modifications to less influential points. This explains why the proposed method achieves stronger cross-model transferability while maintaining favorable geometric quality. The identified regions also provide useful clues for understanding the common decision dependencies of point-cloud classifiers and LiDAR detectors. However, the effectiveness of SAGE still depends on the surrogate model to some extent. When the source and target models use substantially different feature-aggregation mechanisms, the saliency estimated from the surrogate model may contain source-specific cues, which can reduce the transfer gain in some architecture pairs.

The current study focuses on digital-domain robustness evaluation and modifies only the three-dimensional coordinates of existing points. This setting allows the influence of geometric perturbations to be examined independently across different point-cloud perception models. In real LiDAR acquisition, however, point generation is also affected by scanning patterns, angular resolution, viewing distance, occlusion, surface reflectance, and sensor noise. Therefore, the results obtained in this study mainly reflect vulnerabilities

in the downstream digital perception pipeline rather than directly demonstrating physical attack realizability.

Extending digital perturbations to physical environments requires additional consistency constraints, including laser-ray geometry, visibility relationships, range-dependent point density, reflection intensity, and cross-modal alignment. These factors will be incorporated in future work through ray-consistent and occlusion-aware perturbation generation, intensity-aware optimization, sensor simulation, and controlled physical evaluation. In addition, the transferable critical regions and adversarial samples generated by SAGE can support future studies on adversarial training, anomaly detection, and robustness enhancement for LiDAR-based perception systems.

6. Conclusions

In this paper, we proposed **SAGE**, a saliency and geometry enhanced transferable attack framework for LiDAR point cloud perception in remote sensing. By integrating saliency estimation with geometry-aware perturbation control, SAGE identifies transferable critical regions and generates adversarial point clouds that are both highly effective and geometrically plausible. Extensive experiments on classification benchmarks and realistic remote sensing perception scenarios demonstrate that SAGE substantially improves transfer attack success rates over existing baselines while preserving geometric imperceptibility. These results suggest that SAGE provides an effective solution for robustness evaluation in real-world remote sensing perception systems. In future work, we will extend SAGE to more realistic physical and temporal settings and further investigate defense strategies against transferable point cloud attacks.

Author Contributions: Conceptualization, Y.W., S.L. and W.L.; methodology, Y.W., S.P. and W.L.; software, X.Z. and Z.F.; validation, Y.W., M.C. and Z.F.; formal analysis, Y.W., S.P. and X.Z.; investigation, Y.W., S.L. and W.L.; resources, S.L. and W.L.; data curation, Y.W., M.C. and Z.F.; writing—original draft preparation, Y.W. and S.P.; writing—review and editing, Y.W., S.L. and W.L.; visualization, Y.W. and X.Z.; supervision, S.L. and W.L.; project administration, S.L. and W.L.; funding acquisition, S.L. and W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under grant number 62401225; the Fujian Provincial Natural Science Foundation of China under grant numbers 2024J01723, 2025J09046, 2024J01115; the Natural Science Foundation of Xiamen of China under grant number 3502Z202472018; the Jimei University Scientific Research Start-Up Funding Project under grant number ZQ2024034, and the Open Competition for Innovative Projects of Xiamen under grant number 3502Z20251012.

Data Availability Statement: The data used in this study are from open and publicly available datasets. Further inquiries can be directed to the corresponding author.

Acknowledgments: The authors would like to thank the providers of the public datasets and open-source implementations used in this work. During the preparation of this manuscript/study, the authors used ChatGPT (OpenAI's GPT-5.5) for the purposes of translation and language polishing of the manuscript. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Balado, J.; Arias, P.; Lorenzo, H.; Meijide-Rodríguez, A. Disturbance analysis in the classification of objects obtained from urban LiDAR point clouds with convolutional neural networks. *Remote Sens.* **2021**, *13*, 2135. [CrossRef]
2. Kettelgerdes, M.; Sarmiento, N.; Erdogan, H.; Wunderle, B.; Elger, G. Precise adverse weather characterization by deep-learning-based noise processing in automotive lidar sensors. *Remote Sens.* **2024**, *16*, 2407.

3. Zhang, Y.; Ding, M.; Yang, H.; Niu, Y.; Ge, M.; Ohtani, K.; Zhang, C.; Takeda, K. Lidar point cloud augmentation for adverse conditions using conditional generative model. *Remote Sens.* **2024**, *16*, 2247. [CrossRef]
4. Xiang, C.; Qi, C.R.; Li, B. Generating 3d adversarial point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2019; pp. 9136–9144.
5. Wen, Y.; Lin, J.; Chen, K.; Chen, C.P.; Jia, K. Geometry-aware generation of adversarial point clouds. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *44*, 2984–2999. [CrossRef]
6. Lou, T.; Jia, X.; Gu, J.; Liu, L.; Liang, S.; He, B.; Cao, X. Hide in thicket: Generating imperceptible and rational adversarial perturbations on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2024; pp. 24326–24335.
7. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572.
8. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
9. Wang, X.; Cai, M.; Sohel, F.; Sang, N.; Chang, Z. Adversarial point cloud perturbations against 3D object detection in autonomous driving systems. *Neurocomputing* **2021**, *466*, 27–36. [CrossRef]
10. Liu, D.; Tao, Y.; Zhou, P.; Hu, W. Hard-label black-box attacks on 3d point clouds. *arXiv* **2024**, arXiv:2412.00404.
11. Hamdi, A.; Rojas, S.; Thabet, A.; Ghanem, B. Advpc: Transferable adversarial perturbations on 3d point clouds. In *Proceedings of the European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 241–257.
12. He, B.; Liu, J.; Li, Y.; Liang, S.; Li, J.; Jia, X.; Cao, X. Generating transferable 3d adversarial point cloud via random perturbation factorization. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*; Association for the Advancement of Artificial Intelligence: Washington, DC, USA, 2023; Volume 37, pp. 764–772.
13. Chen, H.; Zhao, S.; Yang, X.; Yan, H.; He, Y.; Xue, H.; Qian, F.; Su, H. ANF: Crafting transferable adversarial point clouds via adversarial noise factorization. *IEEE Trans. Big Data* **2024**, *11*, 835–847. [CrossRef]
14. Pang, S.; Chen, Z.; Zhang, S.; Lu, L.; Liang, S.; Du, A.; Zhou, Y. Towards a 3D Transfer-based Black-box Attack via Critical Feature Guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: Piscataway, NJ, USA, 2025; pp. 26912–26922.
15. Guo, Y.; Liu, W.; Xu, Q.; Zheng, S.; Huang, S.; Zang, Y.; Shen, S.; Wen, C.; Wang, C. Boosting Adversarial Transferability through Augmentation in Hypothesis Space. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*; IEEE: Piscataway, NJ, USA, 2025; pp. 19175–19185.
16. Li, Z.; Du, X.; Lei, N.; Chen, L.; Wang, W. Nopain: No-box point cloud attack via optimal transport singular boundary. In *Proceedings of the Computer Vision and Pattern Recognition Conference*; IEEE: Piscataway, NJ, USA, 2025; pp. 3492–3502.
17. Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; Xiao, J. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2015; pp. 1912–1920.
18. Uy, M.A.; Pham, Q.H.; Hua, B.S.; Nguyen, T.; Yeung, S.K. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2019; pp. 1588–1597.
19. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2012; pp. 3354–3361.
20. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2020; pp. 11621–11631.
21. Hartmann, J.; Alkhatib, H. Uncertainty modelling of laser scanning point clouds using machine-learning methods. *Remote Sens.* **2023**, *15*, 2349. [CrossRef]
22. Algadhi, A.; Psimoulis, P.; Grizi, A.; Neves, L. Assessment of the accuracy of terrestrial laser scanners in detecting local surface anomaly. *Remote Sens.* **2024**, *16*, 4647. [CrossRef]
23. Verhelst, T.E.; Calders, K.; Burt, A.; Demol, M.; D'hont, B.; Nightingale, J.; Terryn, L.; Verbeeck, H. Implications of pulse frequency in terrestrial laser scanning on forest point cloud quality and individual tree structural metrics. *Remote Sens.* **2024**, *16*, 4560. [CrossRef]
24. Čeňava, J.; Tuček, J.; Chudá, J.; Koreň, M. Mobile laser scanning data collected under a forest canopy with GNSS/INS-positioned systems: Possibilities of processability improvements. *Remote Sens.* **2024**, *16*, 1734.
25. Wang, L.; Chen, Y.; Xu, H. Point cloud denoising in outdoor real-world scenes based on measurable segmentation. *Remote Sens.* **2024**, *16*, 2347. [CrossRef]
26. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2017; pp. 652–660.

27. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
28. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic graph cnn for learning on point clouds. *ACM Trans. Graph. (TOG)* **2019**, *38*, 146. [CrossRef]
29. Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; Chen, B. Pointcnn: Convolution on x-transformed points. *Adv. Neural Inf. Process. Syst.* **2018**, *31*. [PubMed]
30. Wu, W.; Qi, Z.; Fuxin, L. Pointconv: Deep convolutional networks on 3D point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2019; pp. 9621–9630.
31. Xiang, T.; Zhang, C.; Song, Y.; Yu, J.; Cai, W. Walk in the cloud: Learning curves for point clouds shape analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: Piscataway, NJ, USA, 2021; pp. 915–924.
32. Guo, M.H.; Cai, J.X.; Liu, Z.N.; Mu, T.J.; Martin, R.R.; Hu, S.M. Pct: Point cloud transformer. *Comput. Vis. Media* **2021**, *7*, 187–199. [CrossRef]
33. Zhao, H.; Jiang, L.; Jia, J.; Torr, P.H.; Koltun, V. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: Piscataway, NJ, USA, 2021; pp. 16259–16268.
34. Wicker, M.; Kwiatkowska, M. Robustness of 3d deep learning in an adversarial setting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2019; pp. 11767–11775.
35. Huang, Q.; Dong, X.; Chen, D.; Zhou, H.; Zhang, W.; Yu, N. Shape-invariant 3D adversarial point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2022; pp. 15335–15344.
36. Shi, S.; Wang, X.; Li, H. Pointrcnn: 3D object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2019; pp. 770–779.
37. Zhou, Y.; Tuzel, O. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2018; pp. 4490–4499.
38. Yan, Y.; Mao, Y.; Li, B. Second: Sparsely embedded convolutional detection. *Sensors* **2018**, *18*, 3337. [CrossRef] [PubMed]
39. Lang, A.H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2019; pp. 12697–12705.
40. Shi, S.; Wang, Z.; Shi, J.; Wang, X.; Li, H. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 2647–2664.
41. Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; Li, H. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2020; pp. 10529–10538.
42. Liu, H.; Wu, Y.; Yu, Z.; Vorobeychik, Y.; Zhang, N. Slowlidar: Increasing the latency of lidar-based detection using adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2023; pp. 5146–5155.
43. Chen, H.; Yan, H.; Yang, X.; Su, H.; Zhao, S.; Qian, F. Efficient adversarial attack strategy against 3d object detection in autonomous driving systems. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 16118–16132. [CrossRef]
44. Wu, J.; Yao, W.; Jia, S.; Jiang, T.; Zhou, W.; Ma, C.; Chen, X. Gradient-based sparse voxel attacks on point cloud object detection. *Pattern Recognit.* **2025**, *160*, 111156.
45. Zhang, Y.; Hou, J.; Yuan, Y. A comprehensive study of the robustness for lidar-based 3d object detectors against adversarial attacks. *Int. J. Comput. Vis.* **2024**, *132*, 1592–1624.
46. Zheng, S.; Liu, W.; Guo, Y.; Zang, Y.; Shen, S.; Wen, C.; Cheng, M.; Zhong, P.; Wang, C. Sr-adv: Salient region adversarial attacks on 3d point clouds for autonomous driving. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 14019–14030. [CrossRef]
47. Dai, A.; Chang, A.X.; Savva, M.; Halber, M.; Funkhouser, T.; Nießner, M. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2017; pp. 5828–5839.
48. Li, G.; Xu, G.; Qiu, H.; He, R.; Li, J.; Zhang, T. Improving adversarial robustness of 3D point cloud classification models. In *Proceedings of the European Conference on Computer Vision (ECCV)*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 672–689.
49. Zhang, H.; Yu, Y.; Jiao, J.; Xing, E.; El Ghaoui, L.; Jordan, M. Theoretically principled trade-off between robustness and accuracy. In *Proceedings of the International Conference on Machine Learning (ICML)*; PMLR: Cambridge, MA, USA 2019; pp. 7472–7482.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.