Explainability-Aware Frustum Attack: Exposing Structural Vulnerabilities in LiDAR-Based 3D Object Detectors

Chengzeng You, Binbin Xu*, and Soteris Demetriou

Imperial College London, UK
{chengzeng.you19, b.xu17, s.demetriou}@imperial.ac.uk

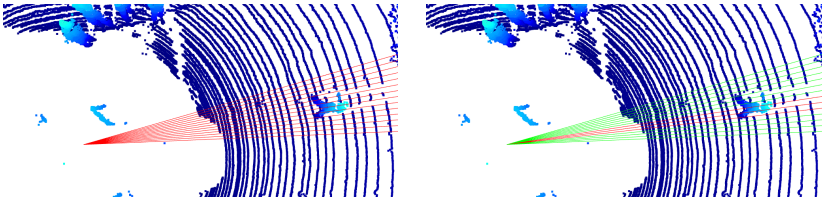
Abstract. The structural vulnerabilities of point cloud–based 3D object detectors remain poorly understood. Prior work has studied adversarial robustness primarily on isolated 3D object models, while recent LiDAR spoofing attacks target richer and more realistic driving scenes but focus mainly on physical realizability rather than understanding detector behavior or attack efficiency. In this work, we investigate how LiDAR-based detectors rely on spatial evidence in complex scenes and whether these reliance patterns can be exploited to induce failures more efficiently. To this end, we propose an explainability-guided adversarial analysis methodology. We introduce the Saliency-LiDAR (*SALL*) method, which aggregates Integrated Gradient attributions across scenes to produce universal saliency maps for LiDAR-based 3D object detectors. Guided by these maps, we design the Explainability-aware Frustum Attack (*EFA*), which selectively perturbs only the most influential frustums rather than uniformly attacking entire object regions. Experiments on KITTI and nuScenes, across detectors such as PointPillars and SECOND, show that EFA reduces detection recall by more than 15 percentage points while requiring 25–50% fewer perturbed frustums than the SOTA non–saliency-aware baseline. These findings reveal that modern 3D detectors concentrate discriminative evidence in a small subset of spatial regions, exposing a structural robustness vulnerability in current LiDAR perception systems. Our code is released at `https://github.com/SecMindLab/Saliency_LiDAR`.

Keywords: 3D Object Detection · Point Cloud Saliency · Security

1 Introduction

Understanding the vulnerabilities of modern 3D object detectors is increasingly important as these models are deployed in real-world perception systems. While significant progress has been made in accuracy, far less is known about the structural dependencies that govern their decisions and how attribution patterns affect robustness. Prior explainability-driven work has begun probing this question. Tan *et al.* [22] showed that, for isolated 3D object classifiers operating on

* Now at Google.



(a) Previous attacks [3, 16, 17] on pedestrian. **(b)** Our attack on pedestrian.

Fig. 1: Saliency-guided hiding visualization. Red rays denote perturbed frustums while green rays denote unperturbed frustums. Prior approaches remove all frustums associated with the object, whereas our method perturbs only a few critical frustums.

normalized CAD models [24], a sparse set of critical points can strongly influence predictions. However, such studies are limited to digital classification settings and do not address full-scene 3D object detection pipelines, where localization, proposal generation, non-maximum suppression, clutter, occlusion, and range-dependent sparsity fundamentally alter the decision structure.

Autonomous driving presents both a more critical and more challenging testbed. LiDAR-based 3D detectors are central to safety-critical systems increasingly deployed in urban environments. At the same time, driving scenes contain multiple interacting objects, background clutter, sensor sparsity, and geometric constraints absent from simplified benchmarks [24]. Understanding structural failure modes in this setting is therefore both timely and methodologically more challenging than simplified digital benchmarks. Prior work in this domain has primarily studied LiDAR spoofing and frustum-level attacks [3, 16, 17]. These methods demonstrate that manipulating or removing entire frustums can hide objects, but typically require perturbing large spatial regions. They do not explicitly analyze whether a detector's reliance is concentrated in smaller, structurally critical regions, nor whether such regions are consistent across scenes and models.

In this work, we bridge explainability and LiDAR attack research by introducing **SALL**, a framework that aggregates instance-level Integrated Gradients into *universal, class-level* saliency maps for LiDAR-based 3D object detection. These maps reveal stable spatial importance distributions across rich real-world 3D scenes, from different datasets, and when derived from distinct detector architectures. We then use these saliency priors to construct a structured perturbation strategy, *EFA*, operating under geometry-consistent constraints inspired by LiDAR sensing. Fig. 1 illustrates the key distinction. Whereas prior attacks [3,16,17] perturb all frustums associated with an object, our method enables targeting only the most salient regions as identified by *SALL*. Our evaluation shows that universal saliency maps are transferable across datasets collected in real-world driving scenarios and key detector architectures. Under geometry-consistent perturbation constraints, targeting saliency-identified regions, our *EFA* achieves over 95 % degradation rates, improving effectiveness by 20 % to 30 % on large objects and approximately 23 % on small objects compared to prior frustum-based approaches [3, 16, 17], while affecting 25 % to 50 % fewer regions.

In summary, our main contributions are as follows:

- We propose **SALL**, a framework that aggregates instance-level attributions into universal, class-level saliency maps for LiDAR-based 3D object detection.
- We demonstrate that these maps reveal consistent discriminative regions which transfer across detector architectures and datasets in realistic autonomous driving scenarios.
- We introduce a geometry-consistent saliency-guided frustum selection and perturbation strategy (*EFA*) and show that it substantially outperforms prior frustum-based methods [3, 16, 17] while requiring significantly smaller perturbation budgets.

2 Related Work

2.1 LiDAR Spoofing Attacks

LiDAR measurements can be spoofed by replaying LiDAR pulses to create fake points in the sensed environment. Such an attack is challenging for the LiDAR system to recognize as it doesn't require any physical contact with the LiDAR sensor or interference with the processing of sensor measurements. To perform realistic attacks, researchers have been improving the hardware and software of LiDAR spoofers [3, 10, 16–19]. A common attack strategy is to capture LiDAR signals from the victim LiDAR, then add a time delay and fire fake laser beams back to the victim LiDAR. Fake points are shown to be reliably injected to fool 3D object detectors to output erroneous predictions. As a result, real objects can be hidden [3,9,10,13,16,17,23,28] while ghost objects can be injected [6,10,16,19]. Real object hiding is regarded as a more dangerous type of attack than ghost object injection, as it is more likely to cause fatal collisions. Based on digital effects, object hiding attacks can be categorized into 2 types of attacks: point-level hiding attacks and frustum-level hiding attacks. For point-level hiding attacks, the adversary leveraged spoofers, attack algorithms or 3D-printed objects [13, 23] to shift [9, 28] or introduce a few points [10] in the view of LiDAR-based perception. Frustum-level hiding attacks [3, 16, 17], by taking advantage of recent spoofing hardware improvements, have been shown to be more powerful than point-level hiding attacks. Specifically, the adversary is equipped with the capability to perturb the whole frustum space to successfully hide objects. Once the frustum is targeted, all points inside the frustum are perturbed.

This creates a clear trade-off: frustum-level attacks are powerful but highly inefficient as they require a large spoofing area that can lead to hardware overheating [17]; while point-level attacks are efficient but less powerful. Our work bridges this gap by proposing a saliency-guided frustum selection and perturbation strategy to achieve the effectiveness of a full frustum attack with the operational efficiency of a point-level one.

2.2 Critical Points and Saliency Map

Identifying *which* regions to attack is a central challenge in our work. Prior research has shown that critical points [15] contribute to features of max-pooling

layers and summarize skeleton shapes of input objects [22]. Based on critical points, researchers further studied the model robustness by perturbing or dropping critical point set identified through monitoring the max-pooling layer or accumulating losses of gradients [11, 26, 29]. However, capturing the output of the max-pooling layer struggled to identify discrepancies between key points, and simultaneously, saliency maps based on raw gradients have been proven to be defective [1, 20]. To overcome these issues, Tan *et al.* [22] introduced Integrated Gradient (IG) [21] which is oriented on generating saliency maps of inputs by calculating gradients during propagation, to investigate the sensitivity of model robustness to the critical point sets and successfully fooling target classifiers with very few point perturbations. However, this study identified critical points specific to object point clouds [24] without further summarizing richer 3D scenes [2, 5]. As a result, for every instance of an object, the attacker needs to run a separate iterative optimization process. Zhu *et al.* [30] proposed an attack framework based on which the attacker could identify a few adversarial locations in the 3D LiDAR scene. They further studied the characteristics of adversarial locations by visualizing them using region maps. However, the region map generation using digital point injection attacks failed to deal with point occlusion problems which disobeyed the physics of LiDAR. This results in reduced reliability of the region map. Additionally, the region map is not considered representative due to limited samples (100 samples), one single dataset and one type of object (car object). Our work, *SALL*, builds on these concepts but differs in three fundamental ways:

1. **Task:** Inspired by the high-level idea of gradient attribution in the task of single-object classification, we design *SALL* for the more complex domain of *3D object detection* in large-scale autonomous driving scenes.
2. **Universality:** Unlike methods requiring per-instance optimization [22], *SALL* aggregates saliency maps across thousands of complex, real-world LiDAR scenes and objects to derive a *universal saliency map* for an entire object class, despite immense instance variability (direction, size, occlusion, *etc.*).
3. **Realism & Robustness:** Unlike region maps built on unrealistic physics [30], our maps are generated using physically-aware perturbations and are validated across multiple large-scale datasets and object types.

3 Methodology

3.1 Threat Model

We consider a *geometry-constrained adversary* ($\mathcal{A}$) that modifies LiDAR point clouds under constraints consistent with LiDAR sensing. The adversary's capabilities are defined to align with behaviors demonstrated in prior LiDAR spoofing and signal manipulation works [3, 4, 9, 14, 16–19]. Specifically, $\mathcal{A}$ may remove, shift, or displace a subset of points along their original LiDAR ray directions. Such perturbations correspond to physically plausible manipulations of return timing, where points may appear either nearer [18] or further [4, 9] relative to the sensor.

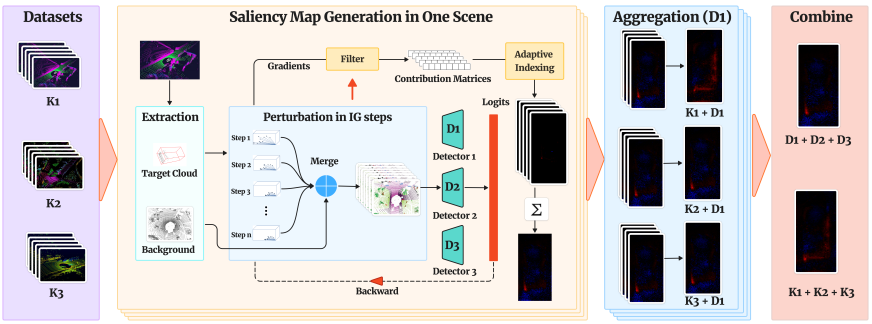


Fig. 2: Overview of universal saliency map generation for LiDAR objects with *SALL*. In *Filter* operation, only gradients of the best predictions are saved by referring to the perturbation region. In *Adaptive Indexing*, point-level saliency maps are downsampled to 2D matrices. Σ denotes the summation of 2D matrices.

We assume these operations can be applied to multiple rays and maintained over time as the platform moves [4, 16, 19].

$\mathcal{A}$ is restricted to localized spatial regions (e.g., angular frustums) under a limited perturbation budget and cannot arbitrarily rewrite the point cloud, introduce globally inconsistent structures, or manipulate other sensing modalities. These constraints preserve ray consistency and local geometric structure while enabling structured adversarial evaluation.

Following Hau *et al.* [9], we assume $\mathcal{A}$ has query access to victim detector outputs (*e.g.*, confidence scores) but no access to model architecture or parameters. The attack therefore operates in a black-box setting with respect to model internals. We also evaluate a version of $\mathcal{A}$ without knowledge of which detector the victim had deployed. The objective of $\mathcal{A}$ is to reduce detection confidence and induce missed detections through geometry-consistent perturbations. Our design goal is not to emulate a specific spoofer, but to analyze 3D detector sensitivity under structured perturbations grounded in prior empirical demonstrations.

3.2 Saliency-LiDAR and Saliency Maps

To identify critical regions for 3D object detection, we propose Saliency-LiDAR (SALL). Inspired by Tan et al [22], SALL adapts Integrated Gradients to autonomous-driving object detection and aggregates instance-level attributions across scenes into universal saliency maps for each object class. Fig. 2 summarizes the pipeline.

Preprocessing. *SALL* takes a raw 3D scene S as input. Before IG computation, it preprocesses the scene through an extraction module ($\mathcal{E}$). After extracting target object clouds T and background points G, the target object clouds T are fed into the IG component to compute point-level contributions.

Integrated Gradient Computation. For each IG step, points in a T^i are perturbed by a perturbation module ($\mathcal{P}$) and outputs $T^{i'}$. Given a base point

$b_0 = (x_0, y_0)$, each point $p_j = (x_j, y_j)$ in T^i is first converted to a vector $\boldsymbol{z_j} = (x_j - x_0, y_j - j_0)$. The base point is customized. For example, $\mathcal{A}$ can select the LiDAR unit as the base point to obey the physics of LiDAR. $\mathcal{A}$ can also select the corner point of the focus region R to only perturb points inside the region and make better use of each IG step.

Let the overall IG steps be M, at the N-th IG step ($N \leq M$), perturbed point ${p_j}'$ in ${T^i}'$ can be calculated in Eq. (1).

$$ {p_j}' = b_0 + \frac{N}{M} * \boldsymbol{z_j} \tag{1} $$

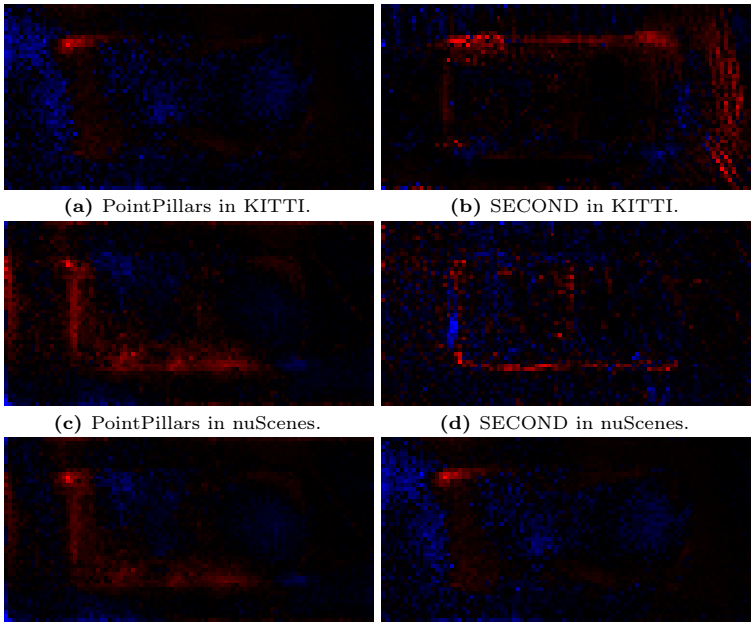
All ${T^i}'$ in perturbed object clouds T' are then merged with the background points (G) to produce the perturbed 3D scene (S'): $S' = T' \bigoplus G$. S' is used for the gradient computation. It passes through a 3D object detector ($\mathcal{D}$) which outputs a set of logits O^i for each target object i. To focus on target objects instead of the whole LiDAR scene, Intersection of Unions (IoUs) between the focus regions R and predicted bounding boxes are first computed to identify the best predictions. Gradients of the best predictions are saved while other gradients are filtered out in the filter module. After that, a point-level contribution map C_i^p is generated per target object. Finally, an integrator function integrates all C_i^p across all IG steps, to produce a point-level saliency or contribution map C^p for objects in a single scene.

Adaptive Indexing. Since R regions have different dimensions and rotations in different LiDAR scenes, to generate a universal saliency map, point-level saliency maps need to be downsampled to pixel-level with the same size. To achieve that, each extracted target point cloud T^i is first converted from LiDAR coordinates to bounding box coordinates. Then, given the target size of the universal saliency map (2D-pixel image), each target object's voxel size is adaptively computed based on R's dimension. After that, indices of each point in T^i can be computed. According to the point-level saliency map C^p, the contribution of each voxel is summed up to generate a 2D-pixel matrix C^v in which each element indicates the contribution of each voxel.

Aggregation Across Scenes. For each scene S, we generate a contribution matrix C^v. C^vs are then aggregated across all k scenes by simple matrix additions to generate the universal saliency map C^{uv} for the target object type.

Combining Different Detectors and Datasets. On the benefit of *SALL*'s aggregation nature, C^{uv}s learned from different datasets (*e.g.*, KITTI [5] or nuScenes [2]) and different detectors (*e.g.*, PointPillars [12] or SECOND [25]) can also be combined to generate a joint universal saliency map which is more representative than the individual saliency map.

Saliency Map Visualization. Fig. 3 shows the saliency map for *Car* objects at 5 - 8m. Most positive pixels fall in edges with some less critical and negative pixels falling in the center of the bounding box. This is true for LiDAR objects where most points appear on the surfaces. Specifically, PointPillars-based saliency maps tend to highlight the left corner as the most critical area while SECOND-based saliency maps prefer to indicate a *L Shape* area that aligns with



(a) PointPillars in KITTI. **(b)** SECOND in KITTI.

(c) PointPillars in nuScenes. **(d)** SECOND in nuScenes.

(e) PointPillars in KITTI + nuScenes. **(f)** PointPillars + SECOND in KITTI.

Fig. 3: Saliency map visualization of *Cars* in different datasets and detectors. Red pixels denote positive contribution values while blue pixels denote negative contributions.

bounding box edges. It is also obvious that the joint saliency maps across different datasets (Fig. 3e) and different detectors (Fig. 3f) become more representative by strengthening common areas and weakening uncommon areas.

3.3 Explainability-Aware Frustum Attack

Explainability-aware Frustum Attack (*EFA*) consists of six phases, taking in raw LiDAR point cloud (S) of the scene, performing perturbation of target objects, and producing the adversarial point cloud (S') as its output.

Phase 1: Extraction. *EFA* detects objects from S. Then it separates S into background points G ($G \in \mathbb{R}^{n_g \times d}$) and a set of target point clouds $T = \{T^1, T^2, ..., T^m\}$. Each target point cloud T^i is extracted from a region of interest R. R can be defined by the ground truth bounding box to solely focus on the exact object cloud or by the expanded dimension bounding box to include surrounding points of the detected object.

Phase 2: 2D Indexing. Each target point cloud T^i is further discretized. Inspired by Lang *et al.* [12], we compute the corresponding indices of each point in pillar format. 2D indexing is more efficient than voxelisation methods, because it does not need to convert points to voxels. Also, the corresponding voxel size is customized and can be set to near point level where each voxel only contains a few points or even one point. Based on the indices, $\mathcal{A}$ can define a virtual patch

as shown in [28] or get critical values according to the coordinates of the saliency map generated by our *SALL* method (Sec. 3.2).

Phase 3: Frustum Simulation. Given LiDAR's origin $P^0 = (x_0, y_0, z_0)$, each point $P_j = (x_j, y_j, z_j)$ in T^i is converted into a 2D degree D_j in Eq. (2).

$$D_j = \arctan \frac{(y_j - y_0)}{(x_j - x_0)} * \frac{180^\circ}{\pi} \tag{2}$$

R's corners are also taken into account for comparison to find a minimum degree D_{min}. Assume the frustum degree is D_{step}, the relative frustum index f_{id} can be calculated in Eq. (3)

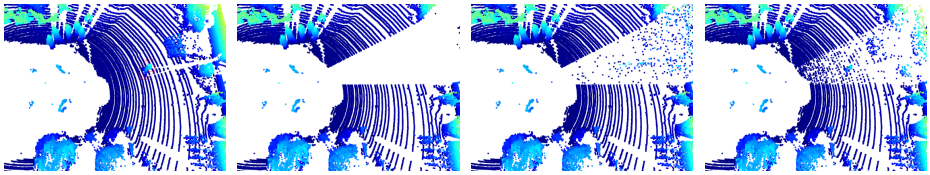
$$f_{id} = \left\lfloor \frac{D_j - D_{min}}{D_{step}} \right\rfloor \tag{3}$$

Phase 4: Frustum Selection. Based on frustum indices, there are 2 frustum selection strategies: *Random Frustum Selection* and *Critical Frustum First*. With *Random Frustum Selection*, given a frustum-perturbation budget, $\mathcal{A}$ randomly selects frustums among frustum candidates. With *Critical Frustum First*, $\mathcal{A}$ selects the most critical frustums according to the *SALL*-generated universal saliency map of the target object. The criticality of each frustum is determined by the criticality of all points inside the frustum. Because of *EFA*'s modular architecture, other novel selection strategies can also be easily incorporated.

Phase 5: Frustum Perturbation. After selecting frustums, different perturbation strategies can be applied to perturb points inside each frustum. For example, *EFA* supports the *Remove Perturbation* strategy similarly to PRA [3] which simply removes all points within a frustum. *EFA* also supports *Shift Perturbation* strategy in [28] which shifts points in accordance with the rays that the LiDAR points fall on. The result is a perturbed point cloud V'.

Phase 6: Merge. All V's are then merged with G to output the final adversarial 3D LiDAR scene $S' = G \bigoplus V'$. S' is in the same format as the original LiDAR scene S, and can be fed into any LiDAR-based detectors for evaluations.

Instantiation of Prior Frustum-Level Perturbations. To enable controlled comparisons, we digitally instantiate the perturbation strategies proposed in prior frustum-level LiDAR attacks, including PRA [3], HFR [16], and A-HFR [17]. For alignment with prior works, perturbations are applied to the central region of the target object. HFR is instantiated using *Shift Perturbation*, PRA using *Remove Perturbation*, and A-HFR using *Remove Perturbation* with point removal rates reported in [17]. All methods operate over a 30° frustum span to ensure comparable perturbation budgets. As illustrated in Fig. 4, these instantiations reproduce the characteristic digital effects reported in [3, 16, 17]. In subsequent experiments, we use these digitally instantiated baselines to compare against our saliency-guided strategy (*EFA*) under identical frustum constraints.



(a) Benign scene. **(b)** PRA [3] **(c)** HFR [16] **(d)** A-HFR [17]

Fig. 4: Instantiation of prior frustum-level LiDAR attacks. Our simulations of attacking a 30° frustum area can achieve the same digital attack effects as [3, 16, 17].

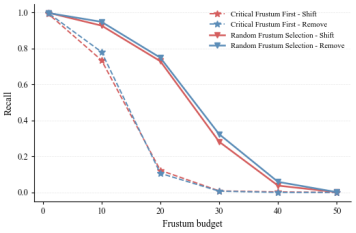
4 EFA Evaluation

4.1 Experimental Setup

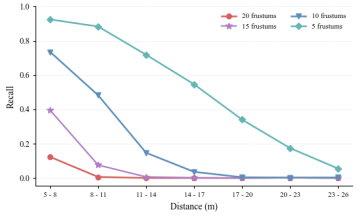
Datasets. We use the KITTI [5] trainval dataset (7481 scenes) and the nuScenes [2] trainval dataset (34 149 samples) for saliency map generation and evaluation.
Target Objects. We focus on *Car* objects within 5 m to 8 m in front of the ego-vehicle. Due to the lower frequency of *Pedestrian* objects, we expanded their focus region to 0 m to 20 m. For each object, we define the Region of Interest (R) by expanding its ground-truth bounding box by a scale factor of 1.5 (as described in Sec. 3.3) to understand the contribution of surrounding points.
Target Models. We evaluate our attack against two popular and representative 3D object detectors: PointPillars [12] and SECOND [25].
Attack Parameters. We define our *EFA* using a frustum degree $D_{step} = 1°$, matching the average resolution of LiDARs attacked in prior work [16]. For *SALL*, we set the number of Integrated Gradient steps to $M = 25$ and use the nearest corner of R as the IG base point (see Sec. 3.2). Our adaptive indexing module normalizes all saliency maps to a fixed size of 64×32 voxels, resulting in an average voxel size of approximately $0.1\,\text{m} \times 0.1\,\text{m}$.
Evaluation Metrics. We use the recall of the detector to study the adversarial robustness of object detection (Sec. 4.2). To accurately evaluate the attack performance, we also define Attack Success Rate (ASR) as the ratio of the number of hidden objects out of all targeted objects.

4.2 Adversarial Robustness of Object Detection

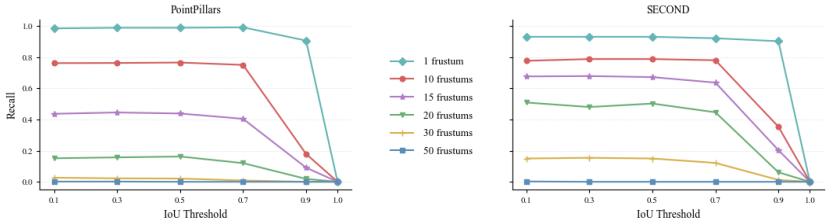
Robustness under Attack Strategies. We first evaluated the detection robustness under different *EFA* attack selection and perturbation strategies. We defined four types of attacks to test *Car* objects predicted by PointPillars in KITTI dataset. As shown in Fig. 5a, recalls of all strategies decreased with increasing frustum budgets. The *Critical Frustum First* selection strategies (guided by *SALL*) decreased the recall to 0.0 using a frustum budget of 30, while *Random Frustum Selection* required 50 frustums (the entire ROI) to achieve the same result. We find that the **selection strategy** is the dominant factor. Once the critical frustums are identified, the specific **perturbation strategy** (*Shift* or *Remove*) has a negligible impact on the outcome. This is a key finding,



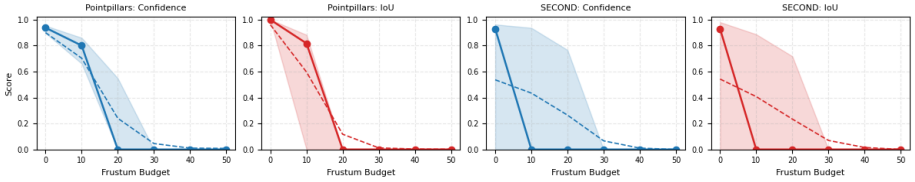
(a) Robustness under attack strategies.



(b) Robustness at different distances.



(c) Detectors with different IoU thresholds.



(d) Robustness of detectors' internal parameters.

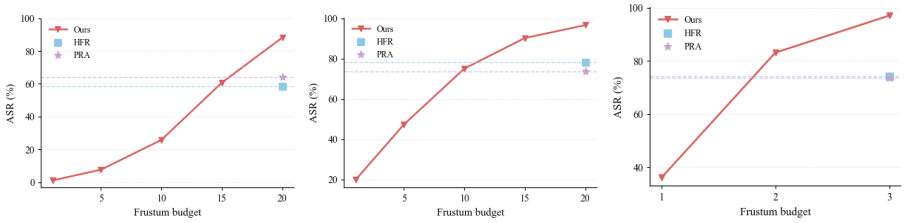
Fig. 5: Adversarial robustness of 3D object detection under the EFA attack.

as it simplifies the adversary's task to one of simply identifying and perturbing the most salient regions.

Robustness at Different Distances. We tested the adversarial robustness of *Car* objects detection from 5 m to 26 m using different frustum budgets as shown in Fig. 5b. It is noticeable that further objects are generally easier to attack. Using 5 frustums successfully hid 94.5 % *Car* objects at a distance of around 25 m. This is because when ranges become longer and objects' 3D representation sparser, the detectors' performance usually decreases significantly and therefore adversarially perturbed measurements have a more noticeable effect.

Robustness with IoU Thresholds. We further studied the adversarial robustness of *Car* object detection under our *EFA* (*Critical Frustum First − Shift*) with different IoU thresholds. From Fig. 5c, we observe that when increasing the frustum budget, the recall falls. For *Cars* at $IoU \geq 0.7$, we observe a significant decline in recall for both detectors, with recall falling below 0.6 for most of the attacks showing that our *EFA* attack is very effective in degrading the performance of object detection.

Robustness of Internal Parameters. We further look into detector's internal parameters (confidence score and IoU values) under our EFA attacks. Both metrics quantify how EFA affects the detector's internal prediction quality beyond the



(a) Attack KITTI dataset. **(b)** Attack nuScenes dataset. **(c)** Attack small objects.

Fig. 6: Saliency-guided attack performance on different datasets and object types. With the guidance of saliency maps, our attack consistently outperforms the baselines.

final attack outcome. For both PointPillars and SECOND we report the median across attacked objects, with interquartile ranges shown as shaded regions in Fig. 5d. Our evaluations show that increasing the frustum budget consistently drops both confidence and geometric overlap, confirming EFA's effectiveness from the detector's internal prediction perspective.

4.3 SALL-Guided Attack Effectiveness

Attacking Different Datasets. To validate our primary hypothesis, we deployed *SALL*-guided attack (*Critical Frustum First* selection with *Shift Perturbation*) against *Car* objects on both datasets. We compare our method against two baselines that mimic state-of-the-art (SOTA) brute-force frustum attacks: HFR [16] and PRA [3]. As shown in Fig. 6a and Fig. 6b, our ASR scales with the number of frustums perturbed. With the guidance of *SALL*-generated saliency maps, our *SALL*-guided attack consistently outperforms the baselines. Specifically, with a 20-frustum budget, our attack achieves a 96.55 % ASR on nuScenes, an improvement of over 20 percentage points compared to HFR (78.29 %) and PRA (73.47 %). We observe a similar 30-point improvement on the KITTI dataset.

Crucially, our method achieves the same ASR as the baselines while requiring substantially fewer perturbed frustums: only 10 on nuScenes (a **50 % reduction**) and 15 on KITTI (a **25 % reduction**). This shows that detector vulnerability is concentrated in a small subset of spatial regions, allowing the same attack success with substantially smaller perturbation budgets and potentially mitigating hardware overheating in physical LiDAR spoofing attacks [17].

Attacking Small Objects. We evaluate our attack on small, difficult-to-detect objects, using *Pedestrian* objects in KITTI as a test case. We generated a universal *SALL* map for pedestrians (Fig. 7) and applied our *SALL*-guided attack. The results in Fig. 6c are stark. Our method achieves a 97.00 % ASR with a tiny **3-frustum budget**. This is ≈ 23 percentage points higher than the HFR (74.25 %) and PRA (73.61 %) using the same budget. This indicates that brute-force frustum attacks [16, 17] are particularly inefficient on small objects, where only a few salient frustums are sufficient.

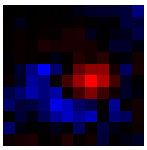
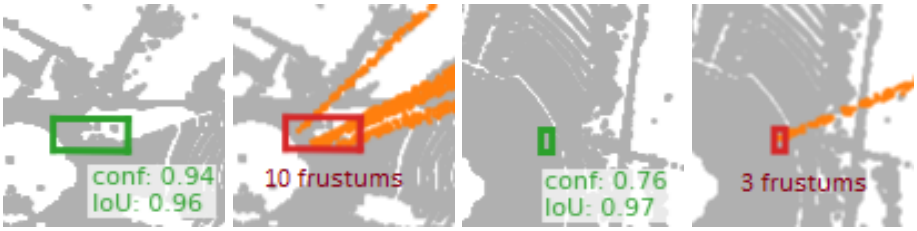


Fig. 7: Universal saliency map of pedestrians.



(a) Car (benign). **(b)** Car (attack). **(c)** Pedestrian (benign). **(d)** Pedestrian (attack).

Fig. 8: EFA attack visualization. Orange points are perturbed, and gray points are unperturbed. Green box shows the target benign object, while red box marks the same region after attack, where the object is hidden.

Attack Visualization. Fig. 8 provides qualitative attack results on *Car* and *Pedestrian* objects, illustrating precise point manipulation and resulting prediction removal. It is also clear that our method perturbs only a few critical frustums to attack *Car* and *Pedestrian* objects.

4.4 Attack Transferability and Generalization

Cross-Model Transferability. To evaluate the transferability of our *SALL* maps, we performed a cross-model attack on KITTI using a 30-frustum budget. We generated *SALL* maps for PointPillars (Fig. 3a) and SECOND (Fig. 3b) individually, as well as a joint map (Fig. 3f). As shown in Tab. 1a, the *SALL* maps show strong transferability. Notably, a map generated from *PointPillars* is effective at attacking *SECOND*, improving ASR by ≈ 7 percentage points over the native *SECOND* map (94.84 % vs. 88.12 %). This suggests *SALL* identifies shared object vulnerabilities rather than being dependent on model-specific artifacts. Furthermore, a joint map created by combining *SALL* maps from both detectors improves performance on both models, achieving 99.44 % on PointPillars. This confirms *SALL*'s aggregation capability and suggests that maps become more universal as more detector architectures are included. Although runtime issues might arise due to multi-modal detectors which can be further avoided by running *SALL* for each detector simultaneously and performing offline combination.
Cross-Dataset Transferability. We performed a similar cross-dataset transferability test using a 20-frustum budget, attacking PointPillars on both KITTI and

Table 1: Saliency map transferability across detectors(left) and datasets (right).

(a) Model dependence (ASR %).

Map Src.	Target Model	
	SECOND	PointPillars
SECOND	88.12	97.63
PointPillars	94.84	98.88
Joint	95.29	99.44

(b) Dataset dependence (ASR %).

Map Src.	Target Data	
	KITTI	nuScenes
KITTI	88.42	96.64
nuScenes	83.82	96.04
Joint	88.15	96.90

nuScenes. As shown in Tab. 1b, the maps are highly transferable. A map learned from KITTI is highly effective on nuScenes, achieving a 96.64 % ASR. Similar to our model-dependence findings, a joint saliency map aggregated from both KITTI and nuScenes further enhanced attack capability (achieving 96.90 % ASR on nuScenes). This demonstrates *SALL*'s robustness and its ability to generate highly representative saliency maps from diverse data sources.

Extending SALL to Point-Level Attacks. In addition to frustum-level attacks, we further explore *point-level targeted attacks* based on saliency maps and compare against ORA [9]. Following a trend similar to Fig. 6 , our *SALL*-guided point attack consistently outperforms the baseline as the perturbation budget increases. With a 1200-point budget, it achieves 91.07 % ASR on KITTI, improving over ORA by more than 37 %, and shows a similar 20 % ASR improvement on nuScenes. Moreover, it reaches the same ASR as the baseline with substantially fewer points: 600 on KITTI (**50** % fewer) and 300 on nuScenes (**40** % fewer). This shows that our *SALL*-guided point-level attack significantly reduces the required spoofing signal, thereby lowering both attack complexity and potential hardware overheating issues.

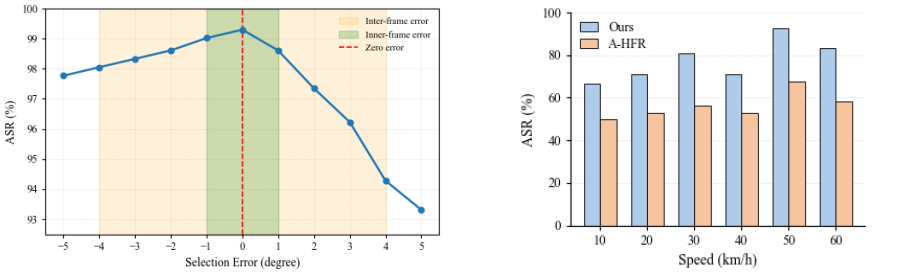
4.5 Runtime Overhead

Universal saliency maps are generated once offline and reused during attack execution. Offline generation consists of *Preprocessing* (3.8 ms), *Adaptive Indexing* (14 ms), *Aggregation* (3.5 ms), and *IG Computation* (51.79 s), averaged over 10 scenes, on a 12 GB NVIDIA TITAN Xp GPU, with the latter dominating the one-time cost due to point density. Online execution requires only saliency lookup using compact maps (5.8 KB for *Cars* and 0.3 KB for *Pedestrians*), introducing negligible overhead over existing LiDAR spoofing attacks.

4.6 Real-World Feasibility

Realistic LiDAR Spoofing Errors. SOTA synchronized LiDAR spoofing hardware is, by design, intended to precisely control the position of each spoofed point. This assumption has also been experimentally validated by HFR [16], which reported an inner-frame error of $\approx 10\,\mathrm{cm}$ and an inter-frame error of $\approx 35\,\mathrm{cm}$. For front-near objects at 5 m to 8 m, these correspond to 0.7°–1.1° and 2.5°–4.0° frustum selection error, respectively. We therefore evaluate EFA under selection errors from $-5°$ to $5°$. As shown in Fig. 9a, the ASR degrades only slightly: even within realistic inner-frame and inter-frame error ranges, the attack remains highly effective (over 98% and 94% ASR, respectively). This demonstrates EFA's strong robustness to realistic frustum selection errors.

Speed-Dependent Perturbation Constraints. We further evaluate our method under perturbation budgets derived from the point removal rates reported in A-HFR [17], which vary with vehicle speed. In A-HFR, higher driving speeds reduce the fraction of points that can be reliably manipulated. We simulate these speed-dependent constraints by applying the corresponding removal rates at speeds ranging from $10\,\mathrm{km\,h^{-1}}$ to $60\,\mathrm{km\,h^{-1}}$.



(a) Attack with LiDAR spoofing errors. **(b)** Speed-dependent perturbation constraints.

Fig. 9: Robustness of EFA under realistic physical-world constraints.

We compare the original A-HFR strategy against a *SALL*-guided variant using a 3-frustum budget for *Pedestrian* objects. As shown in Fig. 9b, incorporating *SALL*-based critical-region selection consistently improves attack effectiveness across all speeds. The improvement is most pronounced at higher speeds ($50\,\mathrm{km\,h^{-1}}$ to $60\,\mathrm{km\,h^{-1}}$), where perturbation budgets are more constrained. For example, at $50\,\mathrm{km\,h^{-1}}$, *SALL*-guided A-HFR achieves 92.70 % success compared to 67.38 % for the original method, an improvement of 25 %.

These results indicate that selecting structurally critical regions becomes increasingly important as perturbation budgets shrink, highlighting the efficiency of universal saliency priors under constrained conditions.

5 Defense Analysis

Existing Defenses. Existing defenses such as *CARLO* [19], *Shadow-Catcher* [8] and *3D-TC2* [27] primarily target injection-style anomalies, and their assumptions do not directly carry over to the object-hiding setting. Object-hiding perturbations aim to suppress the detector's output for a real object. This setting differs fundamentally from injection attacks in two respects: *(i)* no bounding box may be produced for the target object, eliminating the anchor region required for frustum-based analysis; and *(ii)* even under an oracle bounding box assumption, density-based checks become less discriminative when perturbations are sparse and localized, as in our saliency-guided setting. Adapting injection-focused defenses to this scenario would therefore require reasoning about missing objects or absence patterns, which goes beyond their original design assumptions.

Obstacle Detection [7] and *FSD* [3] provided good insights of using 3D shadows for detecting object removal attacks, but they provided neither a quantitative analysis nor publicly available resources. We therefore implemented *Obstacle Detection* to the best of our understanding and evaluated it on 7,481 KITTI scenes. We find that even in benign scenarios, the method produces false alarms in 4,561 scenes, yielding a false alarm rate of 60.96%. After manual inspection, we observed that naturally sparse background regions are frequently misinterpreted as shadow artifacts. This highlights the intrinsic difficulty of distinguishing structured removal from normal LiDAR sparsity, suggesting that current shadow-based heuristics face substantial challenges in this setting.

Table 2: Random frustum dropping. To significantly reduce the ASR, more than 10 out of 30 frustums must be removed.

Dropped frustums	0	2	4	6	8	10
ASR (%)	99.16	98.19	94.00	85.63	72.38	59.14

Random Frustum Dropping. We further evaluate a mitigation strategy based on random frustum dropping. After selecting target frustums using *EFA*, we randomly discard a subset of frustums and measure the resulting ASRs. As shown in Tab. 2, the *SALL*-guided perturbation achieves an ASR of 99.16 % without dropping. To reduce the ASR by approximately 40 %, more than 10 out of 30 frustums must be removed. Such substantial random removal indiscriminately suppresses both informative and non-informative regions.

This behavior reflects a structural trade-off: modern 3D detectors concentrate discriminative evidence in a limited set of spatial regions. While this concentration improves detection efficiency, it also implies that perturbations targeting these regions are disproportionately effective. Conversely, defenses that indiscriminately discard large portions of the input risk degrading nominal detection performance unless the model is explicitly trained to distribute reliance more redundantly.

Overall, our findings suggest that input randomization alone is insufficient to mitigate saliency-aware perturbations. Model-level regularization or adversarial training could be potential alternatives for improving robustness.

6 Conclusion

We propose **SALL**, an explainability-driven method for probing structural vulnerabilities in LiDAR-based 3D object detectors. By aggregating instance-level attributions into universal class-level saliency maps, we uncover stable spatial attribution patterns that persist across scenes, datasets, and detectors. Unlike prior classification-based studies on isolated 3D objects, we evaluate full-scene autonomous driving perception under realistic sensing conditions. Under geometry-consistent LiDAR perturbation constraints, targeting universally salient regions degrades detection performance with substantially smaller perturbation budgets than uniform strategies. Across multiple detectors and datasets, saliency-guided perturbations achieve over 95 % degradation rates, improving effectiveness by 20 % to 30 % on large objects and approximately 23 % on small objects compared to prior approaches [3, 16, 17], while affecting 25 % to 50 % fewer regions.

Our results show that 3D detectors concentrate discriminative evidence in spatially structured regions, creating a robustness–efficiency trade-off. More broadly, they demonstrate that attribution-driven adversarial analysis is a principled diagnostic tool for understanding and improving 3D detector robustness.

Future work could explore training strategies that encourage spatial redundancy or reduce saliency concentration, and extend attribution-driven analysis to more diverse benchmarks, additional sensors and multi-sensor fusion systems.

References

1. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B.: Sanity checks for saliency maps. Advances in neural information processing systems **31** (2018)
2. 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. pp. 11621–11631 (2020)
3. Cao, Y., Bhupathiraju, S.H., Naghavi, P., Sugawara, T., Mao, Z.M., Rampazzi, S.: You can't see me: Physical removal attacks on {LiDAR-based} autonomous vehicles driving frameworks. In: 32nd USENIX Security Symposium (USENIX Security 23). pp. 2993–3010 (2023)
4. Cao, Y., Xiao, C., Cyr, B., Zhou, Y., Park, W., Rampazzi, S., Chen, Q.A., Fu, K., Mao, Z.M.: Adversarial sensor attack on lidar-based perception in autonomous driving. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 2267–2281 (2019)
5. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013)
6. Hallyburton, R.S., Liu, Y., Cao, Y., Mao, Z.M., Pajic, M.: Security analysis of camera-lidar fusion against black-box attacks on autonomous vehicles. In: 31st USENIX Security Symposium (USENIX Security 22). pp. 1903–1920 (2022)
7. Hau, Z., Demetriou, S., Lupu, E.C.: Using 3d shadows to detect object hiding attacks on autonomous vehicle perception. In: 2022 IEEE Security and Privacy Workshops (SPW). pp. 229–235. IEEE (2022)
8. Hau, Z., Demetriou, S., Muñoz-González, L., Lupu, E.C.: Shadow-catcher: Looking into shadows to detect ghost objects in autonomous vehicle 3d sensing. In: Computer Security–ESORICS 2021: 26th European Symposium on Research in Computer Security, Darmstadt, Germany, October 4–8, 2021, Proceedings, Part I 26. pp. 691–711. Springer (2021)
9. Hau, Z., Kenneth, T., Demetriou, S., Lupu, E.C.: Object removal attacks on lidar-based 3d object detectors. In: Workshop on Automotive and Autonomous Vehicle Security (AutoSec). vol. 2021, p. 25 (2021)
10. Jin, Z., Ji, X., Cheng, Y., Yang, B., Yan, C., Xu, W.: Pla-lidar: Physical laser attacks against lidar-based 3d object detection in autonomous vehicle. In: 2023 IEEE Symposium on Security and Privacy (SP). pp. 1822–1839. IEEE (2023)
11. Kim, J., Hua, B.S., Nguyen, T., Yeung, S.K.: Minimal adversarial examples for deep learning on 3d point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7797–7806 (2021)
12. 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 Conference on Computer Vision and Pattern Recognition. pp. 12697–12705 (2019)
13. Modas, A., Sanchez-Matilla, R., Frossard, P., Cavallaro, A.: Toward robust sensing for autonomous vehicles: An adversarial perspective. IEEE Signal Processing Magazine **37**(4), 14–23 (2020)
14. Petit, J., Stottelaar, B., Feiri, M., Kargl, F.: Remote attacks on automated vehicles sensors: Experiments on camera and lidar. Black Hat Europe **11**, 2015 (2015)
15. 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. pp. 652–660 (2017)

16. Sato, T., Hayakawa, Y., Suzuki, R., Shiiki, Y., Yoshioka, K., Chen, Q.A.: LiDAR Spoofing Meets the New-Gen: Capability Improvements, Broken Assumptions, and New Attack Strategies. In: Network and Distributed System Security Symposium (NDSS) (2024)
17. Sato, T., Suzuki, R., Hayakawa, Y., Ikeda, K., Sako, O., Nagata, R., Yoshida, R., Chen, Q.A., Yoshioka, K.: On the realism of lidar spoofing attacks against autonomous driving vehicle at high speed and long distance. In: Proceedings of the Network and Distributed System Security Symposium (NDSS) (2025)
18. Shin, H., Kim, D., Kwon, Y., Kim, Y.: Illusion and dazzle: Adversarial optical channel exploits against lidars for automotive applications. In: International Conference on Cryptographic Hardware and Embedded Systems. pp. 445–467. Springer (2017)
19. Sun, J., Cao, Y., Chen, Q.A., Mao, Z.M.: Towards robust {LiDAR-based} perception in autonomous driving: General black-box adversarial sensor attack and countermeasures. In: 29th USENIX Security Symposium (USENIX Security 20). pp. 877–894 (2020)
20. Sundararajan, M., Taly, A., Yan, Q.: Gradients of counterfactuals. arXiv preprint arXiv:1611.02639 (2016)
21. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: International conference on machine learning. pp. 3319–3328. PMLR (2017)
22. Tan, H., Kotthaus, H.: Explainability-aware one point attack for point cloud neural networks. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 4581–4590 (2023)
23. Tu, J., Ren, M., Manivasagam, S., Liang, M., Yang, B., Du, R., Cheng, F., Urtasun, R.: Physically realizable adversarial examples for lidar object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13716–13725 (2020)
24. 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 conference on computer vision and pattern recognition. pp. 1912–1920 (2015)
25. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10), 3337 (2018)
26. Yang, J., Zhang, Q., Fang, R., Ni, B., Liu, J., Tian, Q.: Adversarial attack and defense on point sets. arXiv preprint arXiv:1902.10899 (2019)
27. You, C., Hau, Z., Demetriou, S.: Temporal consistency checks to detect lidar spoofing attacks on autonomous vehicle perception. In: Proceedings of the 1st Workshop on Security and Privacy for Mobile AI. pp. 13–18 (2021)
28. You, C., Hau, Z., Xu, B., Demetriou, S.: Adversarial 3d virtual patches using integrated gradients. In: 2024 IEEE Security and Privacy Workshops (SPW). pp. 289–295. IEEE (2024)
29. Zheng, T., Chen, C., Yuan, J., Li, B., Ren, K.: Pointcloud saliency maps. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1598–1606 (2019)
30. Zhu, Y., Miao, C., Zheng, T., Hajiaghajani, F., Su, L., Qiao, C.: 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. pp. 1945–1960 (2021)