Shape-invariant 3D Adversarial Point Clouds

Qidong Huang[1]   Xiaoyi Dong[1]   Dongdong Chen[2]   Hang Zhou[3]

Weiming Zhang[1,*]   Nenghai Yu[1]

[1]University of Science and Technology of China   [2]Microsoft Cloud AI   [3]Simon Fraser University

{hqd0037@mail., dlight@mail., zhangwm@, ynh@}ustc.edu.cn

{cddlyf@, zhouhang2991@}gmail.com

Abstract

Adversary and invisibility are two fundamental but conflict characters of adversarial perturbations. Previous adversarial attacks on 3D point cloud recognition have often been criticized for their noticeable point outliers, since they just involve an "implicit constrain" like global distance loss in the time-consuming optimization to limit the generated noise. While point cloud is a highly structured data format, it is hard to constrain its perturbation with a simple loss or metric properly. In this paper, we propose a novel Point-Cloud Sensitivity Map to boost both the efficiency and imperceptibility of point perturbations. This map reveals the vulnerability of point cloud recognition models when encountering shape-invariant adversarial noises. These noises are designed along the shape surface with an "explicit constrain" instead of extra distance loss. Specifically, we first apply a reversible coordinate transformation on each point of the point cloud input, to reduce one degree of point freedom and limit its movement on the tangent plane. Then we calculate the best attacking direction with the gradients of the transformed point cloud obtained on the white-box model. Finally we assign each point with a non-negative score to construct the sensitivity map, which benefits both white-box adversarial invisibility and black-box query-efficiency extended in our work. Extensive evaluations prove that our method can achieve the superior performance on various point cloud recognition models, with its satisfying adversarial imperceptibility and strong resistance to different point cloud defense settings. Our code is available at: https://github.com/shikiw/SI-Adv.

1. Introduction

Deep Neural Networks (DNNs) have contributed a lot to various computer vision tasks with their highly praised performances. However, recent works prove that DNNs are
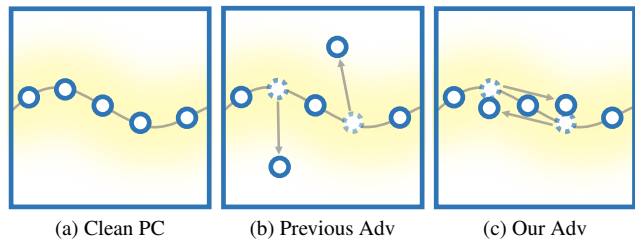
---
*Corresponding author.

Figure 1. The point-cloud shape surface of clean **(left)**, previous adversarial perturbations **(middle)** and our perturbation **(right)**.

vulnerable to adversarial examples [11, 25], which are generated by adding imperceptible perturbations on clean inputs to mislead the prediction of deep learning models. Numerous algorithms have been proposed to implement different kinds of adversarial attacks [4, 8, 11, 13, 25]. In addition to 2D images, the research scope of adversarial attacks has been gradually extended to 3D point clouds. As one of the most popular data representations captured by 3D sensors like LiDAR, point clouds are widely used in a variety of 3D computer vision applications, such as autopilot [42] and autonomous driving [20]. Similar with image recognition models, recent works [3, 22, 24, 28, 30, 38, 40] show that point-cloud models are also susceptible to adversarial perturbations. But unfortunately, existing 3D adversarial attacks either generate the noticeable point perturbation or focus on the white-box case, leading to the poor performance on well-defended models or unseen black-box models.

The motivation for delving into black-box attacks on point cloud recognition is quite clear. On the one hand, both industry and academia have attached increasing importance on evaluating the robustness of point cloud models, especially for some security-critical applications like autonomous driving. But usually, they can not release the details of the model or algorithms to robustness testers due to the protection agreement of trade secrets. Therefore, testers can only conduct black-box attacks to reach the demand of robustness evaluation, which requires the more efficient

Insensitive ▬▬▬▬▬▬▬▬ Sensitive

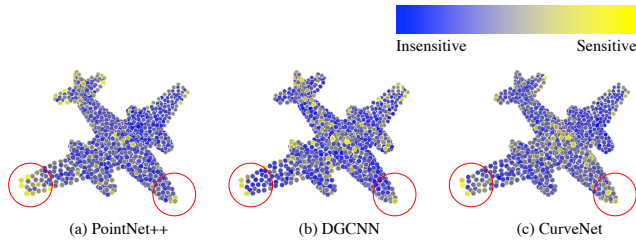(a) PointNet++   (b) DGCNN   (c) CurveNet

Figure 2. Visualization results of the point-cloud sensitivity maps obtained from different 3D recognition models.

black-box attack method. On the other hand, black-box attacks are obviously more practical but challenging compared with white-box attacks, since the attacker can hardly access to the target model in many realistic scenarios. Considering these points, in this paper we adopt a novel shape-invariant perspective to rethink the adversarial imperceptibility and black-box efficiency of adversarial point clouds, based on the following two crucial observations.

First, we mathematically analyze the vulnerability of a white-box point cloud recognition model by defining the novel point-cloud sensitivity map. This map is constructed by a non-negative score set, where each score measures the variance of the recognition confidence when the corresponding point encounters the imperceptible perturbation. To realize such imperceptibility, we intuitively design the point perturbation along the certified optimal direction on the the tangent plane, assuming this slight point shift does not change the shape surface of the point cloud. Each point cloud input can obtain its unique sensitivity map on the white-box model, where all points can be ranked in a descending order according to their scores. Through iteratively perturbing the top ranked 30% points, we find that the white-box recognition accuracy can be degraded over 40% on different models, which demonstrates the existing of shape-invariant white-box attacks.

Second, we discover that these top ranked points obtained from the same point cloud input are similar across different recognition models. As visualized in Figure 2, the positions of top ranked points almost overlap with each other to a large extent when scored by different 3D classifiers. To support this finding more concretely, we look into the 1024-point data samples on the commonly used point-cloud dataset ModelNet40 [31] and compare the top ranked points for different recognition models, *i.e.*, PointNet [5], PointNet++ [21], DGCNN [29] and CurveNet [34]. It shows that the top ranked 30% points of each point cloud have 75.8% overlapped between every two models on average and 40.3% overlapped among every three of them, which are much more than 30% and 9% obtained by random sampling respectively. Inspired by this internal commonality, we can estimate the top ranked points for the un-

seen black-box model with a white-box surrogate model to improve the black-box attack ability.

Motivated by the above two observations, our paper further proposes the first query-based black-box attack algorithm on point cloud recognition. Through inheriting the ranked sensitivity map prior from white-box surrogate model $\mathcal{H}_w$ to black-box target model $\mathcal{H}_b$, we limit the point movement on the shape surface as shown in Figure 1. To boost the query efficiency, we combine each point with its certified best direction as a point-wise basis. Only one basis is perturbed during each query step according to the sensitivity ranking, in the light of SimBA [14] and LeBA [36]. Experiments on ModelNet40 [31] and ShapeNetPart [37] show that our method achieve the superior performance on various 3D recognition models, which highlights the meaning of the proposed point-cloud sensitivity map.

To summarize, our contributions are three-fold as below.

- We propose the point-cloud sensitivity map, for evaluating the variance of recognition confidence when each point encounters the shape-invariant perturbation.

- Guided by the sensitivity map, we propose the strong shape-invariant white-box attack and the first query-based black-box attack on point cloud recognition.

- Experiments show the efficiency of our attack on various recognition models with the high imperceptibility and the low query cost under black-box settings.

2. Related Work

2.1. Point Cloud Recognition

A point cloud is a sparse and unordered point set sampled from object surfaces by 3D sensors. Thanks to the development of deep learning, various methods [5,12,21,29,34,35] have been proposed to process 3D point clouds. One of the pioneering works is PointNet [5], which directly applies multi-layer perceptron to learn point features and aggregates them with a max-pooling module in an effective way. And its variant, called PointNet++ [21], is a improved hierarchical neural network for better abstracting local features, including single-scale (SSG) and multi-scale (MSG) designs. By exploiting point structures with neighborhood graphs, DGCNN [29] is another typical work using convolutional networks. In this paper, we mainly evaluate our black-box attack on six representative recognition models.

2.2. Existing 3D Adversarial Attacks and Defenses

Current 3D adversarial attack methods can be roughly divided into three categories: point-shifting based [15,28,30, 32,40], point-adding based [33] and point-dropping based [38]. Following previous works w.r.t. adversarial attacks on 2D images, a variety of point-shifting based methods either

utilize the gradient descent or solve an optimization problem like C&W [4] to perturb point sets and deceive target classifiers. As for point-adding and point-dropping based approaches, they depend on changing the number of input points to realize attacks. For instance, Zheng *et al.* propose saliency map [39] to drop out critical points of the input and get model fooled successfully. Corresponding to attacks, defense methods based on point-cloud preprocessing [41], adversarial training [17, 23], gather-vector [7] or provable defense [18] have been seriously under-researched.

Since there is still lack of the study on 3D query-based black-box adversarial attack, we delve into this topic based on the proposed sensitivity map in this paper.

2.3. Query-based Black-box Adversarial Attacks

Traditional query-based black-box attacks on 2D images can be divided into two categories: decision-based [2,9] and score-based [1, 6, 14, 27, 36]. Both of them mainly rely on the limited times of queries to the unseen black-box model to achieve attacks, without any inner model information or gradient guidance. The different things is that, score-based methods utilize the output confidence score, while decision-based methods need the output label from the query feedback. In our work, we extend the score-based black-box attack to 3D point-cloud recognition, which is a seriously under-researched field for adversarial attack.

3. Preliminaries

3.1. Point Cloud Sensitivity Maps

For a point cloud $\mathcal{P}$ that consists of total N points, we can define it as a set of vectors $\{\boldsymbol{p}_i\}_{i=1}^{N}$ where $\boldsymbol{p}_i \in \mathbb{R}^3$ denotes the 3D coordinates of each point. Also, there exists k nearest neighbors $\mathcal{N}_{\boldsymbol{p}_i}$ for each point $\boldsymbol{p}_i = (p_{i1}, p_{i2}, p_{i3})$, which can help us simulate its normal vector by calculating the 3×3 positive semi-definite matrix [16] formulated as

$$\mathcal{C}_{\boldsymbol{p}_i} = \sum_{\boldsymbol{q} \in \mathcal{N}_{\boldsymbol{p}_i}} (\boldsymbol{q} - \boldsymbol{p}_i) \otimes (\boldsymbol{q} - \boldsymbol{p}_i), \qquad (1)$$

where $\otimes$ denotes the outer product operation. Note that the normal vector $\boldsymbol{n}_i = (n_{i1}, n_{i2}, n_{i3})$ of $\boldsymbol{p}_i$ can be acquired by solving the eigen-decomposition on matrix $\mathcal{C}_{\boldsymbol{p}_i}$ and choosing the last unit eigen-vector.

Once both the shape surface point $\boldsymbol{p}_i$ and its normal vector $\boldsymbol{n}_i$ are determined in the original coordinate system, the corresponding plane that is tangent to the shape surface is also uniquely determined, so called tangent plane Ω_i. Since the normal vector $\boldsymbol{n}_i$ is perpendicular to Ω_i, the coordinates of any position $\boldsymbol{m} = (m_1, m_2, m_3)$ on the tangent plane can be easily represented with the following equation:

$$(\boldsymbol{m} - \boldsymbol{p}_i) \cdot \boldsymbol{n}_i = \sum_{j=1}^{3} (m_j - p_{ij})n_{ij} = 0 \qquad (2)$$

where $\cdot$ denotes the inner product operation. As illustrated in Figure 3(a), we assume that this tangent plane Ω_i intersects with the coordinate axes (*i.e.*, x axis, y axis and z axis) of the original coordinate system at position $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$. Combined with Eq.(2), we obtain these three positions as

$$\boldsymbol{A}(\frac{\boldsymbol{p}_i \cdot \boldsymbol{n}_i}{n_{i1}}, 0, 0), \boldsymbol{B}(0, \frac{\boldsymbol{p}_i \cdot \boldsymbol{n}_i}{n_{i2}}, 0), \boldsymbol{C}(0, 0, \frac{\boldsymbol{p}_i \cdot \boldsymbol{n}_i}{n_{i3}}). \quad (3)$$

Obviously, $\boldsymbol{O}$-$\boldsymbol{ABC}$ forms a right angle triangular pyramid since the three axes x, y and z are orthogonal with each other, where $\boldsymbol{O}$ denotes the coordinate origin. As clarified in Figure 3(b), we suppose the projection of origin $\boldsymbol{O}$ on the tangent plane Ω_i as $\boldsymbol{O'}$. Note that $\boldsymbol{OO'} \perp \Omega_i$ and $\boldsymbol{BA} \in \Omega_i$, thus we can get $\boldsymbol{OO'} \perp \boldsymbol{BA}$. Likewise, we can also get $\boldsymbol{OC} \perp \boldsymbol{BA}$. Based on these two vertical relationship, we can obtain $\boldsymbol{BA} \perp \boldsymbol{COO'}$ where $\boldsymbol{COO'}$ means the plane determined by $\boldsymbol{C}$, $\boldsymbol{O}$ and $\boldsymbol{O'}$. Since $\boldsymbol{CO'} \in \boldsymbol{COO'}$, we finally obtain $\boldsymbol{CO'} \perp \boldsymbol{BA}$, which is the fundamental point for us to introduce the following coordinate transformation.

To limit the movement of $\boldsymbol{p}_i$ on the tangent plane Ω_i with only two variables, we reduce its one degree of freedom (DoF). Intuitively, we consider to leverage a transformed coordinate system to represent this shape surface point, based on the orthogonality among $\boldsymbol{OO'}$, $\boldsymbol{CO'}$ and $\boldsymbol{BA}$. Specifically, we assume the direction denoted by $\overrightarrow{\boldsymbol{BA}}$ as the transformed axis x', the direction denoted by $\overrightarrow{\boldsymbol{CO'}}$ as the transformed axis y' and the direction denoted by $\overrightarrow{\boldsymbol{O'O}}$ as the transformed axis z'. As shown in Figure 3(c), the origin of the newly constructed rectangular coordinate system becomes $\boldsymbol{O'}$, with the transformed coordinates of $\boldsymbol{p}_i$ as $\boldsymbol{p}_i' = (p_{i1}', p_{i2}', p_{i3}')$. In this way, the surface point can be constrained on the tangent plane when we assign the coordinate value on axis z' as zero. Therefore, the problem is simplified as formulating the transformation applied for converting the original coordinate system $\boldsymbol{O}$-xyz to the transformed coordinate system $\boldsymbol{O'}$-$x'y'z'$.

Theorem 1. *Assume $f_i : \mathbb{R}^3 \mapsto \mathbb{R}^3$ is the transformation for converting the original coordinate system $\boldsymbol{O}$-xyz to the new coordinate system $\boldsymbol{O'}$-$x'y'z'$. Then f_i is reversibly composited by rotation transformation f_{ir} and translation transformation f_{it}, and new 3×1 coordinates $\boldsymbol{p}_i'$ can be converted with original 3×1 coordinates $\boldsymbol{p}_i$ as*

$$\boldsymbol{p}_i' = \boldsymbol{R}_i(\boldsymbol{p}_i + \boldsymbol{T}_i), \quad \boldsymbol{p}_i = \boldsymbol{R}_i^\top \boldsymbol{p}_i' - \boldsymbol{T}_i, \qquad (4)$$

where $\boldsymbol{R}_i$ and $\boldsymbol{T}_i$ is the rotation transformation matrix and the translation transformation matrix respectively, i.e.,

$$\boldsymbol{R}_i = \begin{pmatrix} \frac{n_{i2}}{\sqrt{1-n_{i3}^2}} & \frac{-n_{i1}}{\sqrt{1-n_{i3}^2}} & 0 \\ \frac{n_{i1}n_{i3}}{\sqrt{1-n_{i3}^2}} & \frac{n_{i2}n_{i3}}{\sqrt{1-n_{i3}^2}} & -\sqrt{1-n_{i3}^2} \\ n_{i1} & n_{i2} & n_{i3} \end{pmatrix}, \qquad (5)$$
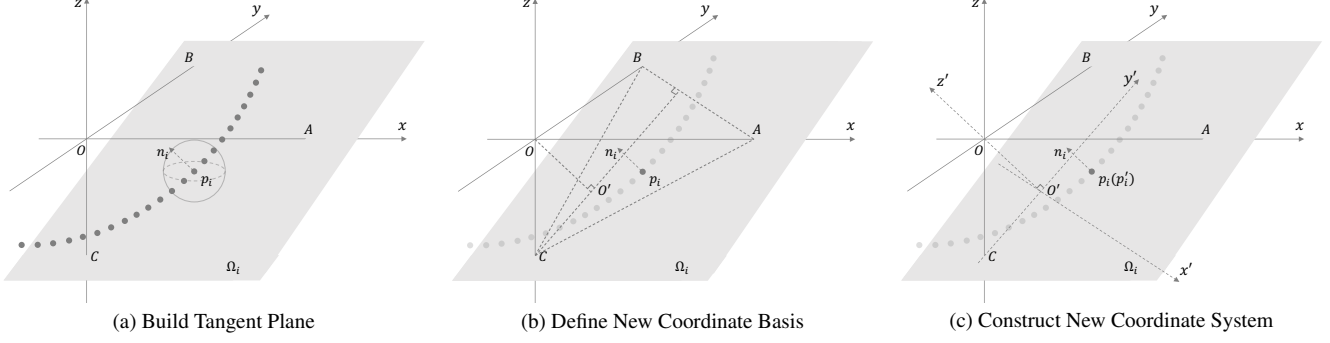
Figure 3. Schematic diagram of the newly constructed coordinate system for our reversible coordinate transformation.

$$\boldsymbol{T}_i = (\boldsymbol{p}_i \cdot \boldsymbol{n}_i)\boldsymbol{n}_i, \tag{6}$$

including a boundary case when $|n_{i3}| \to 1$ that is

$$\lim_{|n_{i3}|\to 1} \boldsymbol{R}_i = \begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ n_{i3}/\sqrt{2} & n_{i3}/\sqrt{2} & 0 \\ 0 & 0 & n_{i3} \end{pmatrix}. \tag{7}$$

Proof. The proof is given in supplementary materials. $\square$

The above transformation facilitates the convenience of the sensitivity map computation. Given a white-box point cloud recognition model $\mathcal{H}_w$ and the point cloud input $\mathcal{P}$, we first apply the reversible transformation elaborated in Eq.(4) on original coordinates of each point, *i.e.*, $\forall \boldsymbol{p}_i \in \mathcal{P}$,

$$\mathcal{P}' = \{\boldsymbol{R}_i(\boldsymbol{p}_i + \boldsymbol{T}_i)\}_{i=1}^N, \quad \mathcal{P} = \{\boldsymbol{R}_i^\top \boldsymbol{p}_i' - \boldsymbol{T}_i\}_{i=1}^N, \tag{8}$$

then we implement the model inference for $\mathcal{P}$ and compute the max-margin logit loss proposed in C&W [4], *i.e.*,

$$\mathcal{L}(\mathcal{P}, t; \theta_w) = \max\left([\mathcal{H}_w(\mathcal{P})]_t - \max_{j\neq t}[\mathcal{H}_w(\mathcal{P})]_j, 0\right), \tag{9}$$

where t is the ground-truth label of point cloud $\mathcal{P}$ and θ_w is the model parameter of white-box model $\mathcal{H}_w$. And $[\mathcal{H}_w(\mathcal{P})]_j$ is the probability logit score on class j. Since the transformation in Eq.(8) is differentiable, we can easily obtain the gradient map of $\mathcal{P}'$ by back propagation,

$$\mathcal{G} = \{\boldsymbol{g}_i\}_{i=1}^N = \nabla_{\mathcal{P}'}\mathcal{L} = \frac{\partial \mathcal{L}(\mathcal{P}, t; \theta_w)}{\partial \mathcal{P}'}, \tag{10}$$

where $\boldsymbol{g}_i = (g_{i1}, g_{i2}, g_{i3}) = (\frac{\partial \mathcal{L}}{\partial x'}, \frac{\partial \mathcal{L}}{\partial y'}, \frac{\partial \mathcal{L}}{\partial z'})$ denotes the gradients of loss $\mathcal{L}$ to transformed point coordinates $\boldsymbol{p}_i'$.

With the guidance of gradient map for the transformed point clouds, the shape-invariant point cloud sensitivity can be defined as: *For each point $\boldsymbol{p}_i \in \mathcal{P}$, we define its sensitivity on white-box model $\mathcal{H}_w$ as the maximal variation rate of the logit loss $\mathcal{L}$ when perturbing $\boldsymbol{p}_i$ along any direction on the tangent plane Ω_i.* Through assigning the last element p_{i3}' of $\boldsymbol{p}_i'$ as a constant zero and ignoring g_{i3} in the gradients

$\boldsymbol{g}_i$, we can limit the point on its tangent plane and measure the aforementioned variation with $\boldsymbol{g}_i = (g_{i1}, g_{i2}, 0)$. So the remaining problem is how to find the direction that achieves the maximal variation on the logit loss $\mathcal{L}$.

Theorem 2. *Given the function $\mathcal{L}$ and the freedom-reduced ternary variable $\boldsymbol{p}' = (x', y', 0)$ initialized as $(p_{i1}', p_{i2}', 0)$, the variation rate of $\mathcal{L}$ is upper bounded at $\sqrt{g_{i1}^2 + g_{i2}^2}$.*

Proof. We ignore the constant zero item and shift the variable in the spherical coordinate system where the point is presented as $(x', y', 0) = (p_{i1}' + r\cos\theta, p_{i2}' + r\sin\theta, 0)$, where θ is the angle between the perturbing direction and the positive half axis of x'. Hence r can be presented as $r = \sqrt{(x' - p_{i1}')^2 + (y' - p_{i2}')^2}$. Based on this, the variation of $\mathcal{L}$ can be regarded as $\nabla_r \mathcal{L}$, upper bounded by

$$\frac{\partial \mathcal{L}}{\partial r} = \frac{\partial \mathcal{L}}{\partial x'}\frac{\partial x'}{\partial r} + \frac{\partial \mathcal{L}}{\partial y'}\frac{\partial y'}{\partial r} = g_{i1}\cos\theta + g_{i2}\sin\theta \tag{11}$$

$$\leq \sqrt{(g_{i1}^2 + g_{i2}^2)(\cos^2\theta + \sin^2\theta)} = \sqrt{g_{i1}^2 + g_{i2}^2}, \tag{12}$$

where we leverage *Cauchy Inequality* and the equal condition of maximum is when angle $\theta = \arctan(g_{i2}/g_{i1})$. $\square$

Therefore, when we use θ to represent the perturbing direction on the tangent plane, the sensitivity score of each point can be determined by the maximal variation rate solved above. Hereby, the sensitivity map of $\mathcal{P}$ is defined as

$$\mathcal{S} = \{\boldsymbol{s}_i\}_{i=1}^N, \quad \text{subject to: } \boldsymbol{s}_i = \sqrt{g_{i1}^2 + g_{i2}^2} \tag{13}$$

3.2. Shape-invariant White-box Attack

With the same definition on point cloud $\mathcal{P} \in \mathbb{R}^{N\times 3}$, white-box model $\mathcal{H}_w$ and gradient map $\mathcal{G} = \{(g_{i1}, g_{i2}, 0)\}_{i=1}^N$, the shape-invariant coordinate transformation denoted in Eq.(8) are applied in our white-box attack. Specifically, we first perform gradient-based optimization on the transformed point cloud $\mathcal{P}'$ with white-box model $\mathcal{H}_w$ to generate the transformed adversarial point cloud $\tilde{\mathcal{P}}'$. Then we apply the inverse transform mentioned

in Eq.(8) to get adversarial point cloud $\tilde{\mathcal{P}}$. Inspired by I-FGSM [13], we implement the adversarial example generation in an iterative style, where rotation matrix gather $\mathcal{R} = \{\boldsymbol{R}_i\}_{i=1}^N$ and translation matrix gather $\mathcal{T} = \{\boldsymbol{T}_i\}_{i=1}^N$ are iteratively updated in each step t to inherit the shape invariance. To simplify the formulation, we first define the following mappings:

$$F_t(\mathcal{P}) = \mathcal{R}_t(\mathcal{P} + \mathcal{T}_t), \quad F_t'(\mathcal{P}') = \mathcal{R}_t^\top \mathcal{P}' - \mathcal{T}_t, \quad (14)$$

where $\mathcal{R}^\top = \{\boldsymbol{R}_i^\top\}_{i=1}^N$. Then in each step t, our untargeted adversarial example generation can be formulated as

$$\tilde{\mathcal{P}}_{t+1}' = \tilde{\mathcal{P}}_t' - \beta \cdot \frac{\nabla_{\tilde{\mathcal{P}}'}\mathcal{L}(F_t'(\tilde{\mathcal{P}}_t'), t; \theta_w)}{\|\nabla_{\tilde{\mathcal{P}}'}\mathcal{L}(F_t'(\tilde{\mathcal{P}}_t'), t; \theta_w)\|}, \quad (15)$$

$$\tilde{\mathcal{P}}_{t+1} = Clip(F_t'(\tilde{\mathcal{P}}_{t+1}')), \quad (16)$$

$$\tilde{\mathcal{P}}_{t+1}' = F_{t+1}(\tilde{\mathcal{P}}_{t+1}), \quad (17)$$

where β means the step size and $Clip$ means the l_∞-norm constraint. When the generation is completed, the adversarial point cloud $\tilde{\mathcal{P}}$ can fool $\mathcal{H}_w$ with satisfying visual quality.

Algorithm 1: Shape-invariant Query-based Attack

Input: point-cloud input $(\mathcal{P}, l)$, black-box model $\mathcal{H}_b$, surrogate model $\mathcal{H}_w$ and step size ϵ.

Output: adversarial point cloud $\tilde{\mathcal{P}}$

1   Initialize the perturbation $\delta = 0$
2   Initialize the prediction pool $\mathbf{p}_c = [\mathcal{H}_b(\mathcal{P})]_c$
3   Transform $\mathcal{P}$ to $\mathcal{P}'$ ◁ Eq.(8)
4   Compute gradient map $\mathcal{G}$ of $\mathcal{P}'$ on $\mathcal{H}_w$ ◁ Eq.(10)
5   Compute and rank sensitivity map $\mathcal{S}$ ◁ Eq.(13)
6   **while** $l = \arg\max_c \mathbf{p}_c$ **and** $\mathcal{S} \neq \varnothing$ **do**
7   Pick top ranked $q \in \mathcal{S}$ and $\mathcal{S} = \mathcal{S} \backslash \{q\}$
8   Get its direction $\theta = \arctan(g_{i2}/g_{i1})$
9   Compute basis $q = q \cdot (\cos\theta, \sin\theta, 0)$
10   **for** $\alpha \in \{\epsilon, -\epsilon\}$ **do**
11   Reverse $\mathcal{P}' + \delta + \alpha q$ to $\mathcal{P}_{inp}$ ◁ Eq.(8)
12   Get prediction $\mathbf{p}_c' = [\mathcal{H}_b(\mathcal{P}_{inp})]_c$
13   **if** $\mathbf{p}_l' < \mathbf{p}_l$ **then**
14   Update $\delta = \delta + \alpha q$
15   New prediction pool $\mathbf{p}_c = \mathbf{p}_c'$
16   **end**
17   **end**
18   **end**
19   **return** $\tilde{\mathcal{P}} = \mathcal{P}_{inp}$

3.3. Shape-invariant Black-box Attack

Given a clean point cloud $\mathcal{P}$ and its ground-truth label l, the goal of query-based black-box attack is to compute the constrained adversarial perturbation δ through fully utilizing the limited times of queries to black-box model $\mathcal{H}_b$.

The queries provide the output logits, *i.e.*, probability scores predicted on each class. In our method, we assume the attacker can adopt a white-box surrogate model $\mathcal{H}_w$ to compute the sensitivity map $\mathcal{S}$ of $\mathcal{P}$. The non-negative scores on $\mathcal{S}$ are ranked in a descending order, so that the most sensitive points could be ranked first. For a black-box model $\mathcal{H}_b$, we give priority to attacking these top ranked points through regarding each point as a basic basis, as done by SimBA [14]. Specifically, in each query step, we only perturb one point along or back along the corresponding direction θ obtained in sensitivity map calculation, since we do not know whether the gradient value in black-box model $\mathcal{H}_b$ is positive or negative. The algorithm is detailed in Algo 1.

Theorem 3. *For our query-based black-box adversarial attack with maximal steps T and fixed step size ϵ, the l_2 norm of perturbation on $\mathcal{P}$ is upper bounded at $\sqrt{T}\epsilon$.*

Proof. Since the perturbation searching is to add or subtract on $\mathcal{P}' + \delta$, each basis q will be discarded if both $+\epsilon q$ and $-\epsilon q$ fail to reduce the confidence $\mathbf{P}_l$. Thus we consider $\alpha_t \in \{\epsilon, 0, -\epsilon\}$ for each step t and describe the perturbation on $\mathcal{P}'$ with a sum as $\sum_{t=1}^T \alpha_t q_t$, where $q_t \in \mathcal{S}$ is the searched basis during each step and the total step number T is just the amount of q_t. Note that these bases are orthogonal with each other, *i.e.*, $q_i^\top q_j = 0, \forall q_i, q_j \in \mathcal{S}, i \neq j$. So combined with $\|\mathcal{R}^\top\|_2^2 = \{1\}_{i=1}^N$, the l_2-norm of the perturbation on $\mathcal{P}$ can be upper bounded with this orthogonal property by

$$\|\delta_\mathcal{P}\|_2^2 = \|(\mathcal{R}^\top \tilde{\mathcal{P}}' - \mathcal{T}) - (\mathcal{R}^\top \mathcal{P}' - \mathcal{T}))\|_2^2 \quad (18)$$

$$= \|\mathcal{R}^\top(\tilde{\mathcal{P}}' - \mathcal{P}')\|_2^2 \leq \|\mathcal{R}^\top\|_2^2 \left\|\sum_{t=1}^T \alpha_t q_t\right\|_2^2 \quad (19)$$

$$= \left\|\sum_{t=1}^T \alpha_t q_t\right\|_2^2 = \sum_{t=1}^T \|\alpha_t q_t\|_2^2 + \sum_{i \neq j}\|\alpha_i q_i \alpha_j q_j\|_2^2 \quad (20)$$

$$= \sum_{t=1}^T \|\alpha_t q_t\|_2^2 = \sum_{t=1}^T \alpha_t^2 \|q_t\|_2^2 \leq \sum_{t=1}^T \epsilon^2 \cdot 1 = T\epsilon^2. \quad (21)$$

Thus the tight upper bound τ of adversarial perturbation $\delta_\mathcal{P}$ is relevant to the maximum steps T, *i.e.*, $\tau = \sqrt{T}\epsilon$. $\quad\square$

4. Experiments

4.1. Setup

Dataset and Data Augmentation. We use ModelNet40 [31] and ShapeNetPart [37] to evaluate the attack performance. ModelNet40 consists of 12,311 CAD models from 40 object categories, in which 9,843 models are intended for training and the other 2,468 for testing. ShapeNetPart consists of 16,881 pre-aligned shapes from 16 categories, split into 12,137 for training and 2,874 for testing. In order to standardize and unify the verification for our method,

Attack	Defense	PointNet [5]				DGCNN [29]				CurveNet [34]			
		ASR↑ (%)	CD↓ (10^{-4})	HD↓ (10^{-2})	A.T↓ (s)	ASR↑ (%)	CD↓ (10^{-4})	HD↓ (10^{-2})	A.T↓ (s)	ASR↑ (%)	CD↓ (10^{-4})	HD↓ (10^{-2})	A.T↓ (s)
I-FGM [13]	-	100.0	6.96	3.04	1.17	99.5	16.85	2.39	2.02	100.0	15.61	2.89	**10.77**
MI-FGM [8]	-	99.5	35.99	4.16	1.31	95.9	120.10	4.99	2.14	99.0	119.17	5.10	10.88
PGD [19]	-	**100.0**	7.00	3.05	**1.17**	99.4	16.77	2.40	**2.02**	100.0	15.32	2.86	10.78
3d-Adv [32]	-	99.9	3.25	2.11	4.94	**100.0**	10.12	2.46	18.73	100.0	7.48	3.51	116.24
AdvPC [15]	-	99.8	16.57	3.43	2.90	98.8	14.35	1.48	8.10	99.9	18.36	2.78	64.76
Ours	-	99.8	**2.15**	**2.04**	1.32	99.9	**6.33**	**1.27**	3.87	**100.0**	**6.25**	**1.99**	21.53

Table 1. Quantitative comparison between our white-box shape-invariant attack and existing white-box attacks in terms of attack success rate (ASR), Chamfer distance (CD), Hausdorff distance (HD) and average time budget for each adversarial point cloud generation (A.T), where CD is multiplied by 10^4 and HD is multiplied by 10^2 for better comparison.
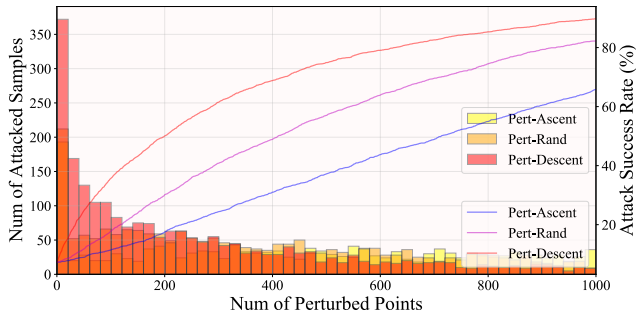

Figure 4. Attack success rate (**curve**) and successfully attacked sample numbers (**histogram**) when perturbing different number of input points on PAConv, the step size is unified as 0.03.

Defense	3d-Adv [32]	AdvPC [15]	GeoA[3] [30]	**Ours**
-	99.9	99.8	**100.0**	99.8
SOR [41]	12.0	33.6	69.6	**97.4**
DUP-Net [41]	12.3	29.0	73.7	**95.8**
AT [19]	94.8	78.6	**99.6**	90.0

Table 2. Attack success rate (ASR) results of different white-box point cloud attacks on PointNet equipped with various defenses.

4.2. Efficiency of Sensitivity Maps

We first conduct the experiment to verify the theoretical validity of the proposed point-cloud sensitivity map. In practice, we follow different orders to perturb each single point of the point cloud input on PAConv [35], where the perturbation direction is along the shape surface and the step size is unified as 0.03. Here "Pert-Descent" means following the descending order of sensitivity maps, and "Pert-Ascent" means following the ascending order by contrast. We also validate the attack performance on the random order denoted by "Pert-Rand". As indicated in Figure 4, the recognition accuracy gets much more degradation in "Pert-Descent", proving the effectiveness brought by the proposed sensitivity map and its descending score ranking.

4.3. White-box Performance

Comparison with Existing Methods. We comprehensively compare our shape-invariant white-box attack with different baselines (*i.e.*, regular gradient-based l_2 attack including I-FGM [13], MI-FGM [8] and PGD [19]; optimization-based point cloud attack including 3d-Adv [32], AdvPC [15] and GeoA[3] [30]. Specifically, our method are implemented with the step size 0.007 for 50 iterations. All adversarial examples are equally constrained by the l_∞-norm ball (radius is 0.16). To unify the verification, all baselines adopt the untargeted mode under the same setting. We conduct the comparison on a same RTX 3090Ti GPU in terms of attack success rate (ASR), Chamfer distance (CD), Haudorff Distance (HD) and average time cost (A.T). The
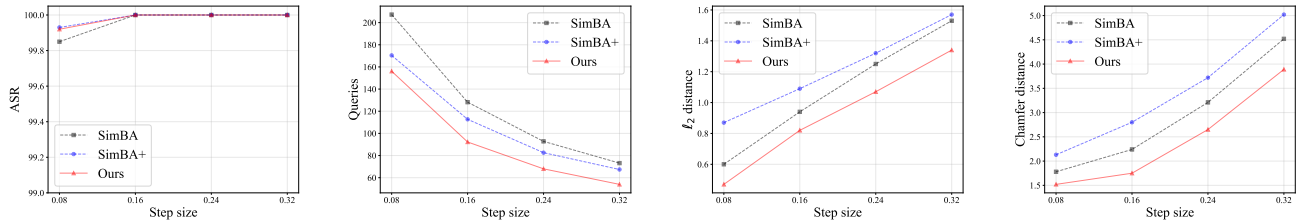
we uniformly sample 1,024 points from the surface of each object and rescale them into a unit cube, as done by Qi *et al.* [5]. Besides, several kinds of data augmentation such as random point-dropping and rotation are applied to preprocess all of point clouds in the training set.

Applied Models. Our approach is evaluated on six categories of the most popular 3D recognition models, *i.e.*, PoinetNet [5], PointNet++ (MSG) [21], DGCNN [29], PAConv [35], SimpleView [12] and CurveNet [34]. In the manuscript, we mainly report the experimental results on ModelNet40 [31], in which we choose DGCNN [29] as the surrogate model for black-box attack evaluation. The results on ShapeNetPart [37] or with PointNet [5] as the surrogate model can be found in supplementary materials.

Evaluation Metrics. To verify the imperceptibility of the generated adversarial point clouds, we compute l_2-norm distance, Chamfer distance (CD) [10] and Hausdorff distance (HD) [26] between the original point cloud and the adversarial output to measure the modification caused by attack methods. To investigate the attack performance quantitatively, attack success rate (ASR) is further defined as the ratio of perturbed examples that are incorrectly classified by the black-box model. Besides, both query numbers and time cost are used for measuring the algorithm efficiency.

Figure 5. Black-box performance when applying different attack step size, in terms of attack success rate (ASR), query cost, l_2-norm distance and Chamfer distance results between adversarial point clouds and originals.

Attack	PointNet++ [21]					PAConv [35]					CurveNet [34]				
	ASR↑ (%)	A.Q↓ (times)	CD↓ (10^{-4})	HD↓ (10^{-2})	A.T↓ (s)	ASR↑ (%)	A.Q↓ (times)	CD↓ (10^{-4})	HD↓ (10^{-2})	A.T↓ (s)	ASR↑ (%)	A.Q↓ (times)	CD↓ (10^{-4})	HD↓ (10^{-2})	A.T↓ (s)
SimBA [14]	94.9	683.8	3.05	4.80	112.04	100.0	73.2	4.51	4.93	0.36	100.0	114.5	4.67	4.93	13.51
SimBA+ [36]	95.5	603.4	3.41	8.21	99.57	100.0	67.4	5.02	9.58	0.36	**100.0**	98.9	5.29	9.76	12.94
Ours	**95.8**	**417.0**	**2.66**	**4.44**	**76.81**	**100.0**	**53.9**	**3.89**	**4.58**	**0.31**	99.9	**81.5**	**3.97**	**4.59**	**8.77**

Table 3. Quantitative comparison between our black-box shape-invariant attack and different black-box attacks with unified step size 0.32 on attack success rate (ASR), average query cost (A.Q), Chamfer distance (CD), Hausdorff distance (HD) and average time budget (A.T).

Defense	PAConv [35]					SimpleView [12]					PointNet [5]				
	ASR↑ (%)	A.Q↓ (times)	CD↓ (10^{-4})	HD↓ (10^{-2})	A.T↓ (s)	ASR↑ (%)	A.Q↓ (times)	CD↓ (10^{-4})	HD↓ (10^{-2})	A.T↓ (s)	ASR↑ (%)	A.Q↓ (times)	CD↓ (10^{-4})	HD↓ (10^{-2})	A.T↓ (s)
-	100.0	53.9	3.89	4.58	0.31	100.0	101.6	7.38	4.77	0.54	99.9	29.9	2.39	4.31	0.14
Drop (30%)	75.4	651.6	1.68	4.12	3.26	51.1	1090.6	2.20	4.25	6.02	96.9	150.9	1.84	4.26	0.47
Drop (50%)	84.4	466.1	1.21	3.71	2.18	63.7	850.0	1.49	3.77	4.94	93.6	260.1	1.70	4.20	0.78
SOR [41]	49.2	1187.6	2.45	3.34	7.82	93.3	576.0	9.73	3.77	3.96	89.7	638.9	9.11	4.20	2.33

Table 4. Resistance of our black-box attack on defended point cloud models. "Drop (30%)" means randomly dropping 30% of input points.

results listed in Table 1 show that our method takes the least geometry distance cost to achieve nearly 100% ASR and the lower time budget compared to regular gradient-based attacks. It is consistent with our intuition of better inheriting the shape invariance from original point clouds during adversarial example generation.

Resistance to Defenses. To verify the attack performance when facing the well-defended point cloud recognition models, we conduct the experiments on defense methods including standard PGD-based adversarial training (AT), point cloud statistical outlier removal SOR [41] and DUP-Net [41]. Trough comparing to three point cloud attack methods in Table 2, we find that our method outperforms the baselines a lot on outlier removal based defenses, which further proves the proposed shape-invariant perturbations are mainly along the shape surface with fewer outliers.

4.4. Black-box Performance

Efficiency on Different Black-box Models. We compare the proposed shape-invariant query-based attack with two related black-box attack algorithms as baselines: SimBA

[14] and SimBA+ [36]. SimBA [14] is one of the most famous query-based black-box adversarial attacks that works well in fooling image recognition models, which randomly searches perturbation direction in the whole orthonormal space to perturb each basis. Fortunately, we can regard all $N \times 3$ dimensions as the orthonormal bases for SimBA implementation on point clouds. SimBA+ [36] further improve SimBA by narrowing the perturbation searching space with the gradient prior obtained on a pretrained surrogate model. But different from image recognition, there are little overlap among pure gradient maps computed on different point cloud models, due to their diverse designs for point feature extraction. So the pure gradient prior only brings limited improvement for SimBA++ on unseen point cloud models. As results shown in Figure 5 and Table 3, our attack achieves comparable ASR, fewer queries and less point modification than baselines on unseen black-box models. In other words, the proposed point-cloud sensitivity map can well depict the internal vulnerability of point cloud recognition models, making the perturbation searching more concrete while maintaining the shape invariance.
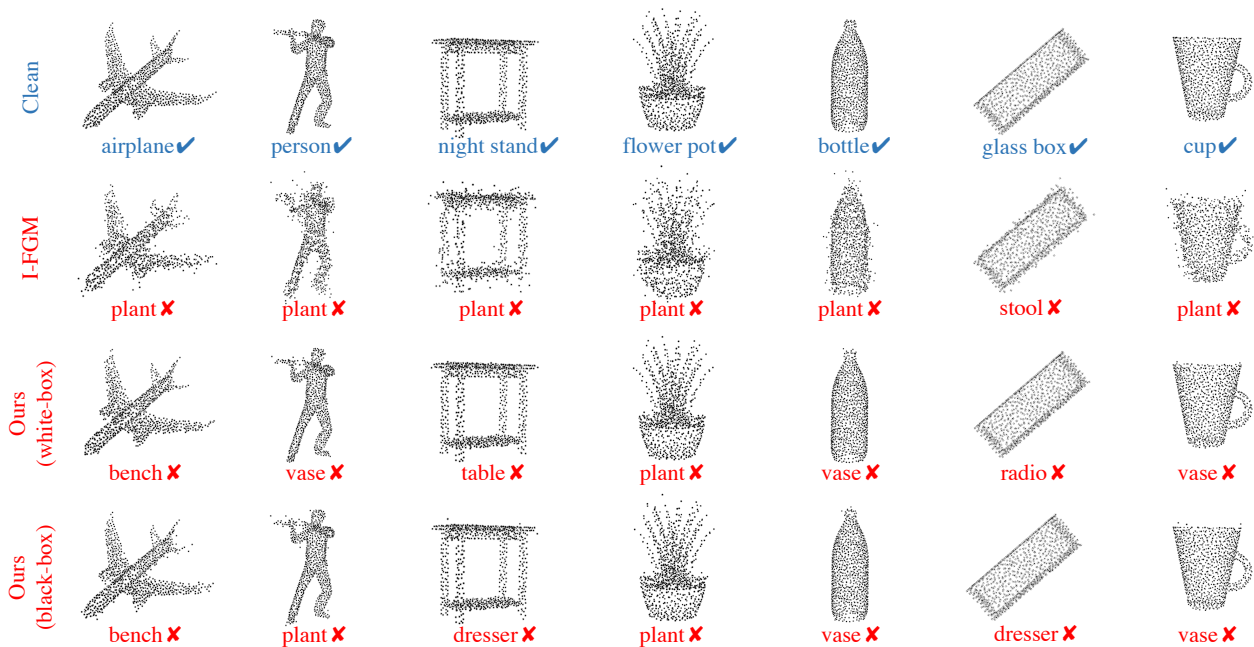
Figure 6. Visualization of clean/adversarial point clouds generated by I-FGM and ours method.

Different Step Size ϵ. The attack step size is a crucial and configurable parameter for query-based attack performance. Accordingly, we conduct the evaluation on a variety of step size ϵ from 0.08 to 0.32. In Figure 5, it can be easily found that the needed queries decreases rapidly under nearly 100% ASR as the step size gets larger. While it significantly strengthens the attack efficacy, the higher l_2-norm distance and Chamfer distance indicate more visual quality it has to sacrifice.

Resistance to Defenses. To investigate the robustness of our shape-invariant black-box attack against different kinds of adversarial defenses, we conduct the experiments on three kinds of defense methods: random point-dropping and input prepocessing based defense SRS [41]. As shown in Table 4, for PAConv [35], SimpleView [12] and Point-Net [5], the ASR on point dropout just degrades up to 36.3% although the needed queries become more. It is worth noted that our method performs even better on PAConv and SimpleView when the dropout defense get enhanced from 30% to 50%. Empirically, we attribute it to the larger natural accuracy degradation of these models when dropped points are increasing. Overall, our black-box shape-invariant attack has the considerable resistance to these defenses.

4.5. Qualitative Results

We further visualize the adversarial point clouds and provide the results in Figure 6. Since our perturbation is almost

along the shape surface, we can hardly find any outliers on the generated adversarial point clouds. Notably, our attack is proved to be more threatening with its well-inherited shape invariance and satisfying attack imperceptibility.

5. Conclusion

In this paper, we propose Point-Cloud Sensitivity Map by calculating the variation rate of recognition confidence when adding the explicitly constrained shape-invariant perturbation on each point, to reveal the vulnerability of point cloud recognition models when encountering the slight perturbation. With the guidance of this map, we further propose Shape-invariant Adversarial Point Clouds to improve the attack imperceptibility and extend the white-box attack to the black-box case. To the best of authors' knowledge, it is the first query-based black-box attack on point cloud recognition. Experiments show the effectiveness and practicality of our method with its superior shape invariance.

6. Acknowledgement

References

[1] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: A query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision (ECCV)*, 2020. 3

[2] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations (ICLR)*, 2018. 3

[3] Yulong Cao, Ningfei Wang, Chaowei Xiao, Dawei Yang, Jin Fang, Ruigang Yang, Qi Alfred Chen, Mingyan Liu, and Bo Li. Invisible for both camera and lidar: Security of multi-sensor fusion based perception in autonomous driving under physical-world attacks. In *42nd IEEE Symposium on Security and Privacy (SP)*, 2021. 1

[4] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy (SP)*, 2017. 1, 3, 4

[5] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 6, 7, 8

[6] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. ZOO: zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017*, 2017. 3

[7] Xiaoyi Dong, Dongdong Chen, Hang Zhou, Gang Hua, Weiming Zhang, and Nenghai Yu. Self-robust 3d point recognition via gather-vector guidance. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[8] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 6

[9] Yinpeng Dong, Hang Su, Baoyuan Wu, Zhifeng Li, Wei Liu, Tong Zhang, and Jun Zhu. Efficient decision-based black-box adversarial attacks on face recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[10] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6

[11] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015. 1

[12] Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. Revisiting point cloud shape classification with a simple and effective baseline. In *International Conference on Machine Learning (ICML)*, 2021. 2, 6, 7, 8

[13] Shixiang Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015. 1, 5, 6

[14] Chuan Guo, Jacob R. Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Q. Weinberger. Simple black-box adversarial attacks. In *International Conference on Machine Learning (ICML)*, 2019. 2, 3, 5, 7

[15] Abdullah Hamdi, Sara Rojas, Ali K. Thabet, and Bernard Ghanem. Advpc: Transferable adversarial perturbations on 3d point clouds. In *16th European Conference on Computer Vision (ECCV)*, 2020. 2, 6

[16] Hugues Hoppe, Tony DeRose, Tom Duchamp, John Alan McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. In *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1992. 3

[17] Daniel Liu, Ronald Yu, and Hao Su. Extending adversarial attacks and defenses to deep 3d point cloud classifiers. In *IEEE International Conference on Image Processing (ICIP)*, 2019. 3

[18] Hongbin Liu, Jinyuan Jia, and Neil Zhenqiang Gong. Pointguard: Provably robust 3d point cloud classification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR)*, 2021. 3

[19] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018. 6

[20] Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J. Guibas. Frustum pointnets for 3d object detection from RGB-D data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[21] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 2, 6, 7

[22] Jiachen Sun, Yulong Cao, Qi Alfred Chen, and Z. Morley Mao. Towards robust lidar-based perception in autonomous driving: General black-box adversarial sensor attack and countermeasures. In *29th USENIX Security Symposium, (USENIX Security)*, 2020. 1

[23] Jiachen Sun, Yulong Cao, Christopher B Choy, Zhiding Yu, Anima Anandkumar, Zhuoqing Morley Mao, and Chaowei Xiao. Adversarially robust 3d point cloud recognition using self-supervisions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 3

[24] Jiachen Sun, Karl Koenig, Yulong Cao, Qi Alfred Chen, and Z. Morley Mao. On adversarial robustness of 3d point cloud classification. *CoRR*, abs/2011.11922, 2021. 1

[25] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2013. 1

[26] Abdel Aziz Taha and Allan Hanbury. Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool. *BMC Medical Imaging*, 2015. 6

[27] Chun-Chen Tu, Pai-Shun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming

Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*, 2019. 3

[28] James Tu, Mengye Ren, Sivabalan Manivasagam, Ming Liang, Bin Yang, Richard Du, Frank Cheng, and Raquel Urtasun. Physically realizable adversarial examples for lidar object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[29] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 2019. 2, 6

[30] Y. Wen, J. Lin, K. Chen, C. L. P. Chen, and K. Jia. Geometry-aware generation of adversarial point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 1, 2, 6

[31] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2, 5, 6

[32] Chong Xiang, Charles R. Qi, and Bo Li. Generating 3d adversarial point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 6

[33] Chong Xiang, Charles R. Qi, and Bo Li. Generating 3d adversarial point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[34] Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2, 6, 7

[35] Mutian Xu, Runyu Ding, Hengshuang Zhao, and Xiaojuan Qi. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 6, 7, 8

[36] Jiancheng Yang, Yangzhou Jiang, Xiaoyang Huang, Bingbing Ni, and Chenglong Zhao. Learning black-box attackers with transferable priors and query feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2, 3, 7

[37] Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas J. Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Trans. Graph.*, 2016. 2, 5, 6

[38] Tianhang Zheng, Changyou Chen, Junsong Yuan, Bo Li, and Kui Ren. Pointcloud saliency maps. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 1, 2

[39] Tianhang Zheng, Changyou Chen, Junsong Yuan, Bo Li, and Kui Ren. Pointcloud saliency maps. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 3

[40] Hang Zhou, Dongdong Chen, Jing Liao, Kejiang Chen, Xiaoyi Dong, Kunlin Liu, Weiming Zhang, Gang Hua, and Nenghai Yu. LG-GAN: label guided adversarial network for flexible targeted attack of point cloud based deep networks.

In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[41] Hang Zhou, Kejiang Chen, Weiming Zhang, Han Fang, Wenbo Zhou, and Nenghai Yu. Dup-net: Denoiser and upsampler network for 3d adversarial point clouds defense. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 3, 6, 7, 8

[42] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1