On Adversarial Robustness of Point Cloud Semantic Segmentation

Jiacen Xu[1], Zhe Zhou[2], Boyuan Feng[3], Yufei Ding[3], and Zhou Li[1]

[1]*University of California, Irvine*
[2]*Fudan University*
[3]*University of California, Santa Barbara*

***Abstract*—Recent research efforts on 3D point cloud semantic segmentation (PCSS) have achieved outstanding performance by adopting neural networks. However, the robustness of these complex models have not been systematically analyzed. Given that PCSS has been applied in many safety-critical applications like autonomous driving, it is important to fill this knowledge gap, especially, how these models are affected under adversarial samples. As such, we present a comparative study of PCSS robustness. First, we formally define the attacker's objective under performance degradation and object hiding. Then, we develop new attack by whether to bound the norm. We evaluate different attack options on two datasets and three PCSS models. We found all the models are vulnerable and attacking point color is more effective. With this study, we call the attention of the research community to develop new approaches to harden PCSS models.**

***Index Terms*—Point Cloud, Semantic Segmentation, Adversarial Perturbation**

I. Introduction

Accurate and robust perception are keys to the success of autonomous systems, with applications on autonomous driving, autonomous food delivery, etc. The main equipments for perception include LiDAR (Light Detection and Ranging) sensor, which uses laser light to measure distances, camera, etc. [9]. These sensors can model the environment as dense, geo-referenced and accurate *3D point cloud*, which is a collection of 3D points that represent the surface geometry.

To process the point cloud data, deep-learning models based on Convolutional Neural Network (CNN) and Graph Convolutional Network (GCN) have been extensively leveraged [21], [22], [50]. Due to their usage in safety-critical applications, a number of works have attempted to generate adversarial examples against such deep-learning models, which migrate the existing attacks against 2D images, like Fast Gradient Sign Method (FGSM) [12], iterative FGSM (iFGSM) [20], Projected Gradient Descent (PGD) [28], and Carlini & Wagner (CW) [7], to the 3D point cloud setting [17], [18], [26], [27], [46], [52], [55], [56], [58], [62], [63]. However, we found all of them focused on the task of *objection recognition*, which identifies objects within an image or a video stream and assigns *one* class label to the *whole* point cloud. Though objection recognition is an important task, *point cloud semantic segmentation (PCSS)* is probably more relevant to real-world autonomous systems, as it is the process of labeling *each point* in a 3D point cloud with a semantic class label, such as "ground", "building", "car", or "tree" and aims to classify *many* objects in a real-world *scene*. The major use cases of PCSS include obstacle avoidance and boundary detection. As far as we know, the work from Zhu et al. [66] is the only one attacking PCSS models, but it is only tested under the setting of LiDAR sensing, and only one outdoor dataset is evaluated. Hence, the robustness of PCSS models under the adversarial samples has not been systematically explored, and there is an urgent need to answer questions like which PCSS model design is more robust and in what setting (e.g., indoor or outdoor) the attack is more likely to succeed, to guide the development of assured autonomy.

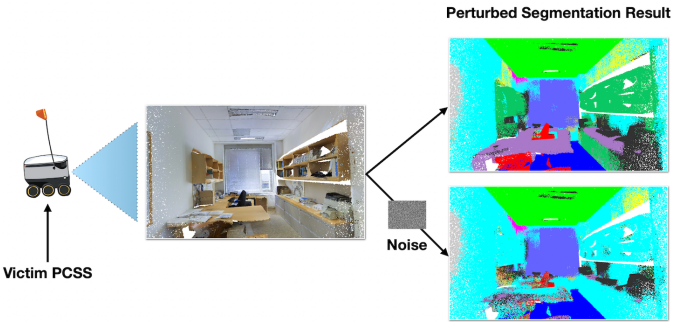

Fig. 1. One example of color-based perturbation under the object hiding attack setting. Different objects are colored differently. Multiple objects (desk, chair, and bookcase) are misclassified as wall after the attack.

Yet, answering these questions is non-trivial if directly applying the adversarial perturbation against 2D images or point cloud object recognition to PCSS. First, in the segmentation task, the class label is assigned to *every* point, and the segmentation result of a point is also determined by its surrounding points, which increases the uncertainty of the attack outcome. Second, various data pre-processing procedures such as point sampling have been applied by the PCSS models, and the attack accuracy can be impacted by them. Finally, perturbation is only performed on the point coordinates by prior works, but there are other point features used for semantic segmentation. For instance, 9 features are included in a point cloud of the S3IDS dataset [57] and 6 features are included in Semantic3D dataset [15]. Whether and how the features undermine the robustness of the PCSS models

have not been studied.

Our Study. We perform the first systematic and comparative analysis on the robustness of PCSS, by developing a *holistic* attack framework that incorporate different attack configurations. We first convert attacker's objectives under *object hiding attack* and *performance degradation attack* into the forms that can be solved through optimization. Under each objective, we develop a norm-bounded attack method adjusted from *PGD* attack [28] and a norm-unbounded attack adjusted from *CW* attack [7], to compare their effectiveness. Previous attacks against point cloud *all* focused on perturbing the point coordinates. However, we found their effectiveness is questionable under PCSS, as point sampling can make the attack outcome hard to control. As such, we exploit the point *features* like color, and adjust the attack methods accordingly. Hence, our attack framework supports *8* attack configurations (object hiding/performance degradation $\times$ norm-bounded/norm-unbounded $\times$ coordinate-based/color-based), which enables a comprehensive analysis.

Evaluation Results. We evaluate the attack framework against three popular PCSS models, including ResGCN-28 [22], PointNet++ [36] and RandLA-Net [16], as they represent different directions in processing point cloud. For the datasets, we use S3IDS and Semantic3D, representing both indoor and outdoor scenes. Below we highlight key findings: **(1)** We compare the perturbation on point features (color in particular) against point coordinates, and our result shows that point features are more vulnerable. **(2)** Under the performance degradation attack, we found all tested PCSS models are vulnerable, with the norm-unbounded attack being more effective (e.g., dropping the segmentation accuracy *from 85.90% to 6.75%* when attacking ResGCN-28). In the meantime, the perturbation added to the original point cloud is still small. Notably, PointNet++, ResGCN-28, and RandLA-Net belong to very different model families, suggesting our developed attacks are universally effective. **(3)** For the object hiding attack, as the attacker needs to select the source objects and determine what they should be changed to, the selection makes a big difference, as some objects are easier to manipulate (e.g., changing board to wall in S3IDS). Figure 1 shows an example of the attack. **(4)** The outdoor scenes are similarly vulnerable comparing to the indoor scenes (e.g., accuracy drops from 98.25% to 16% when RandLA-Net is used to segment Semantic3D point cloud).

Overall, our study demonstrates that the robustness of the deep-learning models under PCSS is questionable, and we outline a few directions for improving their robustness.

Contributions. 1) We develop a holistic framework to enable various attack configurations against PCSS models and extend the previous attacks that are coordinate-based to color-based. 2) We evaluate different attack configurations against three types of PCSS models and indoor and outdoor datasets. Our code is released in a GitHub repository[1].

[1]PointSecGuard: https://github.com/C0ldstudy/PointSecGuard.

II. Background and Related Works

A. Deep Learning on Point Cloud

3D point cloud generated by sensors has become a popular medium to represent the environment interacted by autonomous systems. Two primary tasks have been explored with point clouds, namely objection recognition (or classification), and semantic segmentation. We focus on the second task.

To process the data stored in a point cloud, early works transformed the data into regular 3D voxel grids or images for the conventional CNN, which makes the data unnecessarily voluminous. PointNet [35] addressed this issue by using a shared Multilayer perceptron (MLP) on every individual point, and a global max-pooling to convert the input into a fixed-length feature vector. Since then, variations like PointNet++ [36], Sonet [23], PointCNN [25], KPConv [44], and PointNeXt [37] have been proposed to use sophisticated modules and hierarchical architectures to aggregate local neighborhood information and extract local structure. Besides CNN, DeepGCN [22] shows that GCN can be leveraged to process point clouds. It solves the gradient vanishing problem when models become deeper, as the data are sparse in the geometry space but nodes in adjacency have strong relations. To further reduce the overhead of handling large point clouds like the outdoor Semantic3D dataset, RandLA-Net [16] leverages random point sampling and local feature aggregation, and shows 200× speedup. In this work, we evaluate the point cloud models from the aforementioned three directions, including PointNet++ under CNN, ResGCN under GCN, and RandLA-Net under random point sampling.

Recently, some techniques are proposed to improve the performance of semantic segmentation in particular, including contrastive boundary learning for better scene boundary analysis [43] and multi-view aggregation to leverage the information from the associated 2D images [38]. Other deep models like GAN [40] and transformer [29], [60] have been used to process point cloud. In Section VI, we discuss the adaptability of our attacks to these new models.

Point Sampling. The number of points included by a point cloud for semantic segmentation is often larger than object recognition. For example, 1,024 points are used to represent each object in the ModelNet40 dataset [53], while 4,096 points and 10^8 points are used to represent a scene in the S3DIS dataset [3] and the Semantic3D dataset [15]. Due to that the pre-processing and voxelization steps are computation-intensive, sampling the points in a point cloud becomes a standard approach by PCSS models, such as farthest point sampling [36], and k-NN uniform sampling [22], learning-based sampling [11], [59] and random sampling [16]. We found point sampling makes it more difficult for the existing attacks that perturb point coordinates (to be discussed in Section V-B), which motivates us to explore the new attack methods that exploit point features.

B. Adversarial Examples

The output of a point cloud model can be manipulated under adversarial examples. In the setting of 3D point cloud, existing works [17]–[19], [26], [27], [46], [52], [55], [56], [58], [62], [63] took a gradient-based approach to generate adversarial examples. For example, Kim et al. [19] provide a unified framework to perturb and add points into a point cloud while minimizing the level of point manipulations. Liu et al. [27] adapt the attack and defense methods against the 2D image to 3D point cloud. GAN has also been used to create adversarial examples [64]. However, these works aim to fool object recognition, which is a different task from this paper's focus.

Though attacks against semantic segmentation have been explored, most of the existing works [4], [31], [32] including FGSM [12], iFGSM [20], PGD [28], and CW [7], generate adversarial examples against *2D images*, which have very different properties compared to 3D point clouds. The closest work comes from Zhu et al. [66], which attacked LiDAR PCSS. Yet, only one outdoor dataset is evaluated and the perturbation is only applied to the coordinates. We believe the robustness of PCSS models has not been systematically evaluated, as PCSS can be used by applications other than autonomous driving (e.g., indoor navigation) and point features could also play an important role in addition to coordinates. In fact, we found models like RandLA-Net and PointNet++ extensively leverage point features to boost the accuracy. In this paper, we make the *first* attempt to comparatively analyze the robustness of PCSS, in order to fill this knowledge gap.

C. Point Cloud Datasets

To evaluate the performance of point cloud models, a number of public datasets have been released. For object classification, ModelNet [53], ScanObjectNN [48], ShapeNet [8] and PartNet [30] are widely used. For semantic segmentation, S3DIS [3], Semantic3D [15] and KITTI [5] are the major datasets. Different datasets are created for object classification and semantic segmentation because the number of objects and labels differ: a scene for objection recognition has only one object and one label, while a scene for semantic segmentation usually has multiple objects and labels. It is also more challenging to perform semantic segmentation, as the classification result on one object can be impacted by the nearby objects in the same scene and some objects might only have a partial outline in the scene based on the separation of the point clouds.

In the paper, we select S3DIS [2] and Semantic3D [3] as the datasets to evaluate our attacks in both indoor and outdoor scenes. They both contain coordinate and color, enabling a fair assessment of the attack effectiveness on these two fields.

[2]S3DIS is collected by Matterport scanners that are used for 3D space capture [3].

[3]Semantic3D is collected by high-resolution cameras and survey-grade laser scanners [15].

D. Threat Model

Adversary's Goals. The attacker aims to change the perception results from the PCSS models deployed on autonomous systems like autonomous vehicles and delivery robots and the attack is also known as the evasion attack.

In the real-world setting, for example, the attacker can realize two objectives by carefully introducing adversarial objects [66], patches [47], or laser beam [18] in the surrounding environment perceived by the sensors of the autonomous system. The attack consequences include rear-ending collision, sudden stop, abrupt driving direction change, etc.

We consider two attack scenarios. (1) *Performance Degradation Attack*: this attack tampers the availability [34] of a PCSS model by forcing it to misclassify a large number of points, such that its prediction becomes entirely unreliable. (2) *Object Hiding Attack*: this attack breaks the integrity [34] of a PCSS model by fooling it to classify points under an object as another object or the same as the background.

Adversary's Capabilities. The adversary has white-box access to the victim PCSS model. In other words, the adversary has the read access to the model's structure and parameters and also access to the input of the autonomous system like the physical objects to be sensed. The attacker can generate a point cloud using the same PCSS model as the victim autonomous system and then perturb the points.

We assume the point coordinates or features (or both) can be perturbed by the attacker. We are motivated to investigate both fields because the recent point cloud datasets like S3IDS and Semantic3D contain both point coordinates and features like color. The authors collect data from both frequency-modulated continuous-wave (FMCW) LiDARs and high-resolution cameras, and apply *multi-sensor fusion (MSF)* to assign features onto points. MSF is widely used by autonomous-driving companies like Google Waymo, Pony.ai, and Baidu Apollo to collect environmental information, and a recent work showed it is possible to generate adversarial object that is effective against both FMCW LiDAR and camera [6], suggesting our adversarial samples are potentially realizable. Alternatively, high-end multi-spectral LiDAR can obtain both point coordinates and color, and our attack is expected to be effective as well.

III. Problem Formulation

In this section, we give a formal definition of point clouds and the attacker's goals. Table I lists the main symbols used in this paper and their description.

A point cloud can be defined as a set of N points, i.e., $\{p_i\}_{i=1}^{N}$, where each point $p_i = (pos_x, pos_y, pos_z)$ represents the 3D coordinates of a point. This basic form is usually sufficient for single- object recognition [35], [36]. We denote the features associated with a point p_i as c_i, so a point cloud $X = \{x_i | i = 1 \dots N, x_i = \{p_i, c_i\}\}$ where $c_i = (feat_1, feat_2, ..., feat_k)$ for k features.

Below we formalize the two attack goals. First, we assume $\mathcal{F}_\theta : \mathcal{X} \to \mathcal{Y}$ is the segmentation model which maps an input

TABLE I
MAIN SYMBOLS USED IN THE PAPER.

Symbol	Description
X	a point cloud
Y	the labels of all points in the point cloud
x_i	a point
y_i	a class label on x_i
p_i	the coordinates of x_i
c_i	the features of x_i
R	the perturbation values on the point cloud
r_i	the perturbation values on x_i
T	the set of point indices
$\mathcal{F}_\theta(\cdot)$	the model for PCSS
$Z(\cdot)_i$	the logits of the model's prediction
$\mathcal{T}(\cdot)$	the function selecting a subset of X or Y
$\mathcal{D}(\cdot)$	the distance function

point cloud $X = \{x_i | i = 1 \dots N, x_i \in \mathcal{X}\}$ to the labels of *all points* $Y = \{y_i | i = 1 \dots N, y_i \in \mathcal{Y}\}$. $\mathcal{X}$ is the universe of points, and $\mathcal{Y}$ is the universe of class labels, e.g., desk, wall, and chair. We design methods under both norm-bounded and norm-unbounded principles for the object hiding attack and the performance degradation attack.

Object Hiding Attack. In this setting, the adversary chooses a subset of points $X_T = \{x_i | i \in T, x_i \in X\}$, where T is the set of indices, and perturbs X_T to change their predicted labels to $Y_T = \{y_i | i \in T, y_i \in \mathcal{Y}\}$. For a point $x_i = \{p_i, c_i\}$, we assume the attacker either perturbs its coordinates p_i or (and) its features c_i. We treat coordinates and features separately as the perturbation methods have to be designed differently under their unique constraints. The perturbation values on the original point cloud X can be represented as $R = \{r_i | i \in T\}$, and the new point cloud will be $X' = \{x_i | i \notin T, x_i \in X\} + \{x_i + r_i | i \in T, x_i \in X\}$. Under coordinate-based perturbation, $r_i = \{r_{p_i}, 0^k\}$, where r_{p_i} denotes the changes on the 3D coordinates. Under feature-based perturbation, $r_i = \{0^3, r_{c_i}\}$, where r_{c_i} denotes the changes on the k features.

We first consider the norm-bounded attack, by which the attacker tries to minimize the difference between the predicted labels on X_T and the targeted labels Y_T, while the perturbation is bounded by ϵ. Hence, the attack goal can be formalized as:

$$\arg\min_R \mathcal{L}_T(X', Y_T), \quad \text{s.t. } \mathcal{D}(R) \leq \epsilon \tag{1}$$

where $\mathcal{D}(\cdot)$ is the distance function measuring the magnitude of the perturbation R and $\mathcal{L}_\mathcal{T}(\cdot)$ is the adversarial loss that measures the effectiveness of the attack. Notably, the attacker's goal is quite different from the attacks against single-object recognition [55], where one label is assigned to the whole point cloud (i.e., the cardinality of Y' is 1) and the number of points after perturbation can differ (i.e., X' and X have different cardinalities).

Under the norm-unbounded attack, the attacker tries to find the minimum perturbation values that can change the labels of X_T to Y_T. Hence, the attacker's goal can be formalized as:

$$\arg\min_R \mathcal{D}(R), \quad \text{s.t. } \mathcal{T}(\mathcal{F}_\theta(X'), T) = Y_T \tag{2}$$

where $\mathcal{T}(\cdot)$ selects the prediction results indexed by T from X'.

Directly solving Equation 2 is difficult because the constraint $\mathcal{T}(\mathcal{F}_\theta(X'), T) = Y_T$ is non-differentiable [7]. As a result, we reformulate Equation 2 to enable gradient-based optimization by introducing $\mathcal{L}_T(\cdot)$ to replace this constraint, as shown in Equation 3. Besides, we add a smoothness penalty $\mathcal{S}(\cdot)$ to encourage the optimizer to keep X' smooth, i.e., that the differences between the neighboring points are not drastic.

$$\arg\min_R \{\mathcal{D}(R) + \lambda_1 \cdot \mathcal{L}_T(X', Y_T) + \lambda_2 \cdot \mathcal{S}(X')\} \tag{3}$$

where λ_1 and λ_2 are pre-defined hyper-parameters.

Performance Degradation Attack. In this setting, the adversary does not have a specific target Y_T, but just manipulates the prediction $\mathcal{F}_\theta(X')$ to be different from the ground-truth labels of all points in X_T (termed Y_{GT}). Under norm-bounded attack, the attacker's goal can be adjusted from Equation 1 to:

$$\arg\max_R \mathcal{L}_{NT}(X', Y_{GT}), \quad \text{s.t. } \mathcal{D}(R) \leq \epsilon \tag{4}$$

where $\mathcal{L}_{NT}(\cdot)$ is the adversarial loss regarding Y_{GT}.

Under norm-unbounded attack, the attacker's goal can be adjusted from Equation 3 to:

$$\arg\min_R \{\mathcal{D}(R) - \lambda_1 \cdot \mathcal{L}_{NT}(X', Y_{GT}) + \lambda_2 \cdot \mathcal{S}(X')\} \tag{5}$$

IV. ATTACK DESIGN

As reviewed in Section II, none of the prior attacks against PCSS consider the point features, so we highlight the design of feature-based attacks here, which is supposed to be more resilient against point cloud sampling (see Section II-A). In particular, we select the *color* features as the perturbation target, which turns c_i to $(color_r, color_g, color_b)$ for the three color channels, where $color_*$ is the pixel value and $*$ represents red, green, and blue.

A. Attack Components

Here we elaborate the distance function $\mathcal{D}(\cdot)$, adversarial loss functions $\mathcal{L}_T(\cdot)$ and $\mathcal{L}_{NT}(\cdot)$, smoothness penalty $\mathcal{S}(\cdot)$, which are all listed in Section III.

Distance Function. When the color-based attack is chosen, we use L_2 distance to measure the magnitude of the perturbation, as shown by Equation 6, because L_2 distance is commonly used by 2D image models [7]. Hence, the distance will be:

$$\mathcal{D}(R) = \sum_{i \in T} ||r_{c_i}||_2^2 \tag{6}$$

As pointed out by Nicholas et al. [7], optimization on c_i encounters a box constraint, which is hard to solve. Hence, we map $c_i \in [a, b]$ to a new variable W_i, and perform optimization over w_i, as shown in Equation 7.

$$r_{c_i} = a + \frac{b-a}{2}(\tanh(w_i) + 1) \tag{7}$$

where $\tanh(\cdot)$ makes the gradient smoother and always falls in $[-1, 1]$, so the optimizer could find the right perturbation

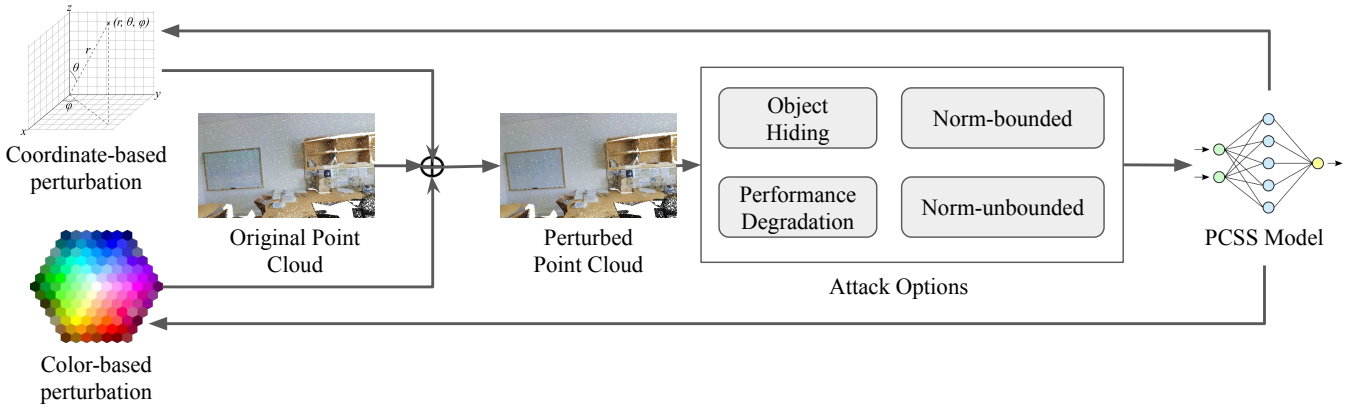

Fig. 2. The attack workflow.

sooner. a and b are adjusted based on the normalized value range of the PCSS model.

When the coordinate-based attack is chosen, we use L_0 distance, i.e., how many points are changed, and the distance can be represented as:

$$\mathcal{D}(R) = \sum_{i \in T} ||r_{p_i}||_0 \tag{8}$$

We use L_0 distance since other distance metrics like L_2 and L_∞ yield different value ranges based on the input range of coordinates. Specifically, a color channel has a fixed range from 0 to 255 regardless of the point clouds, but a coordinate can have different value ranges for different point clouds. Equation 7 is also applied on the coordinate for the ease of optimization.

Smoothness Penalty. The penalty is designed to make X' smooth. In Equation 9, the distance between each point and its top α nearest neighbors is encouraged to be minimized.

$$\mathcal{S}(X', \alpha) = \sum_{x'_i \in X'} \sum_{x'_j \in \text{Nei}(x'_i, \alpha)} (||x'_i - x'_j||_2) \tag{9}$$

where $\text{Nei}(x'_i, \alpha)$ returns the top α nearest neighbors. Noticeably, we consider all points rather than only points in X_T.

Adversarial Loss. We use the logits (the output of the layer before the last softmax layer) of $\mathcal{F}_\theta$ to represent the adversarial loss for the object hiding attack. This loss encourages the optimizer to minimize the logits of the class rather than the target label.

$$\mathcal{L}_T(X', Y_T) = \sum_{\substack{x'_i \in X' \\ y_i \in Y_T}} \max(\max_{j \neq y_i}(Z(x'_i)_j) - Z(x'_i)_{y_i}, 0) \tag{10}$$

where $Z(\cdot)_j$ represents the j^{th} element of the logits of the adversarial example, and $Z(x'_i)_{y_i}$ means the target label's logits for x'_i. The largest logit that is not related to the target label is derived by $\max_{j \neq y_i}(Z(x'_i)_j)$.

For performance degradation attacks, the adversarial loss is adapted to encourage the prediction of points to be any class other than the ground-truth classes. The loss function can be changed from Equation 10 to:

$$\mathcal{L}_{NT}(X', Y_{GT}) = \sum_{\substack{x'_i \in X' \\ y_i \in Y_{gt}}} \max(Z(x'_i)_{y_i} - \max_{j \neq y_i}(Z(x'_i)_j), 0) \tag{11}$$

Algorithm 1: Norm-bounded attack.

Input: the original point cloud X, the ground-truth labels Y_{GT}, the maximal number of iterations $Steps$, the target labels Y_T, the attack type $type = \{color, coordinate\}$, the attack boundary ϵ, the target points mask $\mathcal{T}(\cdot)$ (all points for non-target attack), the targeted points T, the step size γ

Output: the adversarial example X'

$X_0 \leftarrow X, r \leftarrow X(type), i \leftarrow 1$;

while $i \leq Steps$ **do**

  $X_T^i(type) = \mathcal{T}(r)$;

  $\mathcal{T}(r) = X_T^i(type) - X_T^{i-1}(type)$;

  $X_T^i = X_T^{i-1} + \mathcal{T}(r)$;

  if *object hiding attack* **then**

    $X_T^i = \text{clip}_{(-\epsilon,\epsilon)}(X_T^{i-1} - \gamma \cdot \text{sign}(\nabla_{X_T} \mathcal{L}_T(X_T^{i-1}, Y_T)))$;

  else if *performance degradation attack* **then**

    $X_T^i = \text{clip}_{(-\epsilon,\epsilon)}(X_T^{i-1} + \gamma \cdot \text{sign}(\nabla_{X_T} \mathcal{L}_{NT}(X_T^{i-1}, Y_{GT})))$;

  if $Converge$ **then**

    return X_T^i;

  $i \leftarrow i + 1$;

end

return X_T^i;

B. Attack Workflow

Our norm-bounded attack is adjusted from *Projected Gradient Descent (PGD)* [28] to PCSS. In essence, in each iteration, the attack adds noise to the perturbed point cloud

of the previous iteration X_T^{i-1} to derive X_T^i, following the goals defined in Equation 1 (object hiding) and Equation 4 (performance degradation). Then it clips the changes to $(-\epsilon, \epsilon)$ on X_T^i. Random initialization is used to set up X_T^0. To avoid the imbalance during the update, a sign operator is applied on the gradient. We set an upper bound of iterations as Steps. In each step, we use Converge$(\cdot)$ to determine if the adversarial example is satisfactory, based on the attacker's evaluation metrics, e.g., the dropped accuracy. Algorithm 1 lists the main steps.

Our norm-unbounded attack is adjusted from *Carlini and Wagner (CW)* attack [7]. There are two main differences between our implementation with the norm-bounded attack. First, each time when the target color is mapped to a new variable W_i from Equation 7 ($a = 0$ and $b = 1$ for color), it utilizes the inverse function of Equation 7 before perturbing the original data points. Second, instead of using attack boundary ϵ [7], we add a distance function item in the loss function under the goals defined in Equation 3 (object hiding) and Equation 5 (performance degradation) to optimize both the perturbation norm and attack success rate. Like norm-bounded attack, we also bound the attack iterations by Steps. In each step, the gain over the attack is examined by Converge$(\cdot)$ as well. In addition for norm-unbounded attack, if the gain does not increase in 10 steps, the perturbation will add random noise following the uniform distribution in $(0, 1)$. When the adversarial example is invalid, e.g., $c_i \notin [0, 1]^3$, a new noise will replace the previous one.

When the attacker chooses to perturb the coordinates, we select a set of most impactful points and only perturb them, such that the L_0 criteria can be met. Especially, in each iteration i, we assume the points allowed to be perturbed is $X_i \subseteq X_T$. After perturbation, the n least impactful points are selected by the function MinImp(X_i, n), and the point cloud is restored. The next iteration $i + 1$, the perturbation will be only executed on $X_{i+1} = X_i \setminus \text{MinImp}(X_i, n)$. When the remaining points that can be perturbed are less than 10% after a number of iterations, the point cloud will be perturbed without restoration. The equation below shows how the n points are selected.

$$\text{MinImp}(X_T^i, n) = \arg\min_n g_n \cdot r_n \tag{12}$$

where g_n is the gradient and r_n means the perturbation value.

Our default setting either perturbs color or coordinate, but it can be readily adjusted to perturb both fields. Specifically, our method generates the gradients of color and coordinate concurrently in each iteration during optimization, after they are pre-processed separately. The distance function of Equation 8 and adversarial loss from Equation 10 and Equation 11 are reused. An alternate approach is to perturb them in turns at different iterations. However, we found this approach has worse result, because the gradient updates of color and coordinate offset each other.

Noticeably, though our attacks are adjusted from PGD and CW, notable changes exist. For instance, our norm-bounded attack does not use the original cross-entropy loss directly and our norm-unbounded attack adds a new smoothness penalty. Besides, we extend the attacks by customizing data preprocessing procedures by Equation 7.

In Section V-A, the hyper-parameter values used to evaluation both attacks are described.

V. Evaluation

In this section, we report our evaluation results on various attack settings, target models, and datasets. Each experiment is conducted to answer one or few research questions and the findings are highlighted in the end.

A. Experiment Settings

Target Models. We use the pre-trained PointNet++ [36], ResGCN-28 [22], and RandLA-Net [16] as the target models to evaluate our attacks, mainly because their codes and pre-trained models are publicly available[4][5][6], and they represent different directions in the point cloud domain. In Section II, we give an overview about their designs. Below, we elaborate on their details.

PointNet++ consists of 4 abstraction layers and 4 feature propagation layers with 1 voter number for its multi-angle voting. The overall point accuracy and average Intersection-over-Union (aIoU) of the pre-trained PointNet++ on the indoor dataset S3DIS are 82.65% and 53.5% respectively, as reported in its GitHub repo. ResGCN-28 uses dynamically dilated *k*-NN and residual graph connections, and the pre-trained model configures *k* to 16. It has 64 filters and 28 blocks with 0.3 drop-out rate and 0.2 stochastic epsilon for GCN. the accuracy and aIoU of the pre-trained ResGCN-28 model on S3DIS are 85.9% and 60.00%, as reported in its GitHub repo. The pre-trained RandLA-Net downsamples large point clouds. Its accuracy and mIoU are 88.00% and 70.00% on S3DIS, and 94.8% and 77.4% on Semantic3D.

Dataset. We evaluate the attacks on two large-scale 3D datasets: an indoor dataset S3DIS [2] and an outdoor dataset Semantic3D [15]. They have been extensively used as the benchmark for PCSS. The S3DIS dataset is composed of 3D point clouds with color channels, which were collected at 6 areas in 3 different buildings with 13 class labels. Each point cloud contains 4,096 points, and each point has 9 features. The point cloud data could be pre-processed differently by their segmentation models. As for PointNet++, each point cloud is segmented into several parts, and the coordinate and color are normalized to $[0, 3]$ and $[0, 1]$ by themselves. As for ResGCN-28, the coordinate is normalized to $[-1, 1]$ while the color feature is normalized to $[0, 1]$ by itself. RandLA-Net regenerates the point clouds with 40,960 points by randomly duplicating and selecting the points. The color feature is also normalized to $[0, 1]$ by itself.

We evaluate against the three models with the point clouds of Area 5, of which 198,220 point clouds (78,649,818 points)

[4]PointNet++: https://github.com/yanx27/Pointnet_Pointnet2_pytorch
[5]ResGCN-28: https://github.com/lightaime/deep_gcns_torch
[6]RandLA-Net: https://github.com/QingyongHu/RandLA-Net

are included [2]. For the performance degradation attack on S3DIS, we choose the point clouds with high accuracy (over 30%) as targeted point clouds. For the object hiding attack, we randomly selected the 100 point clouds in Office 33 of Area 5 when evaluating against PointNet++ and ResGCN-28. As RandLA-Net has different requirements of the point number, for each class (e.g., table), we randomly selected 100 point clouds in Area 5 which at least contains 500 points in the class.

The Semantic3D dataset contains 30 point clouds including coordinates, color channels, and intensity in 8 classes. Each point cloud has over 10^8 points to compose a $160 \times 240 \times 30$ m^3 area. PointNet++ and ResGCN-28 are not designed to handle such big point clouds, so we only evaluate against RandLA-Net.

Evaluation Metrics. We use the drop of accuracy and *average Intersection over Union* (aIoU), which measures the overall accuracy across all classes, as the indicators of the attack's effectiveness. On a point cloud, the accuracy and aIoU are defined as follows: assuming the number of all points and correctly classified points are N and TP, accuracy equals to $\frac{TP}{N}$. For a class i, the aIoU is defined as $\frac{TP_i}{FN_i+FP_i+TP_i}$, where FN_i, FP_i, TP_i are the number of false negatives, false positives and true positives for the class-related points. Below, we will primarily show the accuracy and aIoU averaged over point clouds.

For the object hiding attack, the drop of accuracy and aIoU only measure whether the classification results are changed, but they neglect whether the predictions are misled toward the target classes. Therefore, we define another metric, *point success rate (PSR)*, as the ratio of *points* that are correctly perturbed over all the attacked points in X_T. Besides measuring the success rate of attacks, we are also interested in whether the segmentation results of points outside of X_T are changed, so we compute the accuracy and aIoU on the subset of these points separately, and call the metrics *"out-of-band" (OOB)* accuracy and aIoU. From the attacker's perspective, the drop of OOB accuracy and aIoU should be as low as possible.

Attack Hyper-parameters. We set Steps to 50 and 1,000 for norm-bounded and norm-unbounded attacks. Both λ_1 and λ_2 used by the adversarial loss are set to 1 and 0.1 based on empirical analysis. The step size (γ) of norm-bounded attack is 0.01 while the Adam optimizer of norm-unbounded attack with 0.01 learning rate (lr) is used. The nearest neighbor α for Equation 9 is set to 10. The batch sizes are set to 8 when attacking PointNet++ while to 1 when attacking ResGCN-28 and RandLA-Net. For the performance degradation attack, we examine whether the accuracy is dropped below 7.6% (*i.e.,* 1/13, 13 classes) for S3IDS and 12.5% (*i.e.,* 1/8) for Semantic3D, which means the model's prediction is the same as random guessing. When coordinate is attacked, n least impactful points should be selected in each iteration, and we set n to 100 during the experiment.

Experiment Platform. We run the experiments on a workstation that has an AMD Ryzen Threadripper 3970X 32-Core Processor and 256 GB CPU memory with 2 NVIDIA GeForce RTX 3090. Our attacks run on PyTorch 1.7.1 for the pre-trained PointNet++ and ResGCN-28, and Tensorflow 1.15 for the pre-trained RandLA-Net.

B. Evaluation on the Attacked Fields

TABLE II
THE RESULTS OF PERFORMANCE DEGRADATION ATTACK ON RESGCN-28. "ACC", "AVG" AND "COORD" ARE SHORT FOR "ACCURACY", "AVERAGE" AND "COORDINATE".

Case		Norm-unbounded			Norm-bounded		
		L_0	**Acc** (%)	**aIoU** (%)	**L_0**	**Acc** (%)	**aIoU** (%)
Color	Best	496.00	0.24	0.12	496.00	0.07	0.04
	Avg	**1635.17**	9.04	4.81	**1130.04**	12.13	6.71
	Worst	4096.00	27.86	16.18	4096.00	67.65	51.12
Coord	Best	2396.00	8.74	4.57	496.00	1.61	0.81
	Avg	4065.21	27.63	16.48	2993.71	16.49	9.21
	Worst	4096.00	63.43	46.44	4096.00	47.88	31.47
Both	Best	496.00	3.54	1.80	496.00	8.15	4.25
	Avg	2519.00	12.57	7.05	2407.00	31.14	21.85
	Worst	4096.00	82.08	69.61	4096.00	94.26	89.15

As described in Section II-B, the prior attacks against point cloud objection recognition and semantic segmentation all perturbed the coordinate field, leaving other channels like color feature unexplored. Hence, we first assess how the attacked fields impact the attack effectiveness. The experiment is conducted on the S3IDS dataset under performance degradation attack and we show the results when ResGCN-28 is the target model. Similar trend is observed on other models.

Due to that the range of coordinates varies by PCSS models, L_2 distance is unsuitable to measure perturbation, as explained in Section IV. Hence, we use L_0 distance for the coordinate-based attack. For a fair comparison, we also change the L_2 distance used by the color-based attack to L_0.

In Table II, we show the best-case, average-case, and worst-case ("best" and "worst" show the examples most vulnerable and robust against the attack) among the attacked point clouds in terms of accuracy, aIoU and L_0 distance (higher accuracy and aIoU are worse for attack). The results show that the average L_0 distance is significantly lower when perturbing color than perturbing coordinate and both (1130.04 for norm-bounded and 1635.17 for norm-unbounded under color, while more than 2000 in any other case). The accuracy and aIoU also observed a deeper drop for color-based perturbation.

A thorough investigation indicates the coordinate-based perturbation performs worse because of the point sampling (elaborated in Section II-A) by the PCSS model. For example, when a point cloud is fed into ResGCN-28, it is sampled by a k-NN algorithm that aggregates the neighborhood points to the centric points. When the coordinates of a point are perturbed, the nearby points are also changed due to point aggregation, so the result of attack might not be controllable. As a supporting evidence, we sampled the whole point clouds on Area 5 from the S3IDS dataset and found *over 88%* of the neighborhood points are changed after coordinate-based perturbation. Perturbing both coordinate and color also leads

to unsatisfactory result for the same reason. On the other hand, perturbing color will not impact how neighborhood points will be sampled, so the attack outcome is more controllable.

Given that perturbing color is much more effective than perturbing coordinates, for the following experiments, all attacks are conducted under color-based perturbation, and we switch the distance back to the default L_2.

Finding 1: The color feature is more vulnerable than point coordinates under perturbation.

C. Evaluation on Performance Degradation Attack

In this subsection, we conduct experiments under performance degradation attack, focusing on the attack effectiveness under different methods and target models. We use S3IDS as the dataset. For the attack methods, in addition to norm-unbounded and norm-bounded attacks, we also implement another method that adds random noises to the color channels, as a baseline to compare against.

In Table III, we show accuracy and aIoU across the point clouds. We also show the L_2 distance between the original point cloud and the perturbed one. It turns out norm-unbounded and norm-bounded attacks both significantly reduce the accuracy and aIoU. For example, norm-unbounded attack *drops the average accuracy of PointNet++, ResGCN-28, and RandLA-Net from 82.65% to 7.86%, 85.90% to 6.75%, and 87.2% to 7.45% separately*. The average drop rate of aIoU ranges from 48.45% to 76.23% under norm-unbounded attack, and from 20.26 to 54.61% under norm-bounded attack. Norm-unbounded attack performs much better for the worst-case scenario (the most difficult sample): e.g., when ResGCN-28 is attacked, the accuracy on the most difficult sample is dropped to 18.34%, but norm-bounded attack has nearly no impacts on the accuracy (99.85%).

Regarding the perturbation distance, we found norm-unbounded attack generates the adversarial examples under smaller or equal distance in the best-case scenario and average scenarios for PointNet++ and ResGCN-28, but the distance becomes larger for RandLA-Net (e.g., 17.06 compared to 16.83 for the average). For the worst-case scenario, it has to add much larger noises to drop accuracy and aIoU.

Regarding the baseline method, we found its effectiveness is quite limited, with the dropping of average accuracy and aIoU ranging from 3.54% to 6.24% and 0.69% to 6.6% respectively. The result suggests PCSS models are robust against random noises and carrying out a successful attack is non-trivial.

Regarding the impact of the target model on the attack effectiveness, we found norm-unbounded attack is similarly effective against PointNet++, ResGCN-28, and RandLA-Net, but norm-bounded attack is much more effective against PointNet++ than ResGCN-28 and RandLA-Net. Hence, we suggest that norm-unbounded attack should be used if the attacker prefers effectiveness, or finding better adversarial examples. In the meantime, it usually takes a longer time to execute due to the longer time to reach the converge requirement.

Finally, we measure the overhead of conducting the attacks. The time to generate an adversarial example is proportional to the number of Steps (see Algorithm 1), and each step takes 0.3 seconds for norm-bounded attack, and 0.2 for norm-unbounded attack. This overhead is acceptable if the attacker uses physical patches [47] or adversarial objects [66], as they are generated offline. The overhead can be further reduced by using a smaller number of Steps. Recently, Guesmi et al. showed it is possible to conduct real-time adversarial attacks by introducing an offline component [13], and our attacks can be adjusted following this direction.

Finding 2: Under performance degradation attack, norm-unbounded attack is more effective, especially for the most difficult point cloud samples.
Finding 3: All tested models are similarly vulnerable under norm-unbounded attack, but PointNet++ is much more vulnerable under norm-bounded attack.

D. Evaluation on Object Hiding Attack

We used Area 5 of S3IDS for evaluation, which contains objects under 13 classes, and we perturb the points from 6 classes, including window (label=5), door (label=6), table (label=7), chair (label=8), bookcase (label=10), and board (label=11), because the quantity of points of each class is not too small. We set the target class as wall (label=2).

Table IV shows the results for the 6 objects under norm-unbounded attack. It turns out *PSR can be over 90%* for window, door, bookcase, and board, against all targeted models. However, PSR is much lower for table and chair. We speculate the reason is that table and chair have more complex shapes, so changing the class labels on these objects is more difficult. In the meantime, the drop in the accuracy of the OOB points is moderate, mostly within 10%, which suggests norn-unbounded is able to confine the changes to the selected objects.

Table V shows the results under norm-bounded attack. Similar to the trend of the performance degradation attack, it performs worse than norm-unbounded attack, i.e., lower PSR for every perturbed object. Regarding the impact of objects, we also found PSR is higher when less complex objects like window, door, bookcase and board are perturbed, but table and chair see much lower PSR, e.g., under 10% for PointNet++ and ResGCN-28. Moreover, norm-bounded attack results in a larger drop of OOB accuracy and aIOU, especially for PointNet++ and ResGCN-28. Since the design of norm-bounded attack bounds the perturbation distance, we found the L_2 distance of adversarial samples is much smaller in most cases, though it comes at the price of lower PSR.

Regarding the target model, we found it is easier to achieve high PSR when attacking RandLA-Net. For example, even table and chair have over 80% PSR for both attack methods.

TABLE III

PERFORMANCE DEGRADATION ATTACK AGAINST POINTNET++, RESGCN-28, AND RANDLA-NET ON S3DIS. THE COLOR FEATURE IS ATTACKED AND L_2 DISTANCE IS USED. ↓ SHOWS THE DROP OF PCSS ACCURACY OR AIOU.

Case		Random Noise			Norm-unbounded			Norm-bounded		
		$\mathbf{L_2}$	**Acc** (%)	**aIoU** (%)	$\mathbf{L_2}$	**Acc** (%)	**aIoU** (%)	$\mathbf{L_2}$	**Acc** (%)	**aIoU** (%)
PointNet++	Best	2.68	14.51	14.26	2.68	4.91	2.31	15.51	5.71	5.19
	Avg	18.19	77.26(5.39↓)	70.02(0.69↓)	18.27	**7.86**(74.79↓)	**8.85**(61.86↓)	18.27	19.11(63.54↓)	16.10(54.61↓)
	Worst	30.02	100	100	30.03	20.33	59.27	22.01	56.71	42.37
ResGCN-28	Best	1.29	7.79	4.11	1.29	0.31	0.16	5.05	0.0	0.0
	Avg	4.30	82.36(3.54↓)	73.17(6.55↓)	4.30	**6.75**(79.15↓)	**3.49**(76.23↓)	6.51	42.16(43.74↓)	30.13(49.59↓)
	Worst	9.81	100	100	9.81	18.34	10.10	7.96	99.85	99.70
RandLA-Net	Best	6.65	34.07	12.56	6.65	6.13	0.92	0.33	18.52	13.62
	Avg	17.06	78.01(6.24↓)	44.82(6.6↓)	17.06	**7.45**(76.75↓)	**2.96**(48.45↓)	16.83	59.60(24.65↓)	31.15(20.26↓)
	Worst	54.62	97.18	70.08	54.62	7.69	8.33	17.00	85.88	66.18

TABLE IV

THE RESULTS OF NORM-UNBOUNDED ATTACK ON WINDOW (LABEL=5), DOOR (LABEL=6), TABLE (LABEL=7), CHAIR (LABEL=8), BOOKCASE (LABEL=10), BOARD (LABEL=11). "PN" MEANS POINTNET++, "RGCN" MEANS RESGCN-28, "RNET" MEANS RANDLA-NET, "SC" MEANS SOURCE CLASS. "OOB/ACC" MEANS THE OUT-OF-BAND ACCURACY AND OVERALL ACCURACY WHILE "OOB/AIOU" IS SIMILAR.

Model	SC	L_2	PSR (%)	OOB/Acc (%)	OOB/aIoU (%)
PN	5	7.67	**93.92**	53.76 / 77.67	46.59 / 60.77
	6	5.39	**93.11**	58.54 / 62.00	45.37 / 49.24
	7	10.55	37.70	79.48 / 86.26	56.04 / 69.09
	8	6.69	17.63	86.09 / 90.65	62.91 / 73.19
	10	15.26	**93.25**	52.73 / 68.43	49.63 / 57.88
	11	5.28	**93.16**	80.47 / 93.97	60.99 / 74.27
RGCN	5	14.57	**95.44**	70.88 / 71.21	39.57 / 58.67
	6	12.17	**94.71**	77.96 / 84.62	65.11 / 76.75
	7	9.29	66.43	81.81 / 91.66	49.08 / 84.80
	8	12.62	51.63	82.22 / 83.84	63.39 / 75.59
	10	16.01	**90.48**	65.10 / 68.43	55.56 / 56.52
	11	9.69	**96.08**	78.37 / 88.85	56.58 / 66.43
RNet	5	3.76	**95.13**	83.98 / 84.41	50.39 / 51.03
	6	2.79	**95.23**	88.42 / 88.78	49.52 / 51.12
	7	11.82	83.10	83.11 / 83.98	45.01 / 50.58
	8	9.06	85.98	82.78 / 82.94	47.50 / 47.94
	10	8.55	**95.05**	84.02 / 84.71	50.07 / 51.27
	11	2.37	**94.29**	84.80 / 85.57	52.22 / 54.06

TABLE V

THE RESULTS OF NORM-BOUNDED ATTACK ON WINDOW (LABEL=5), DOOR (LABEL=6), TABLE (LABEL=7), CHAIR (LABEL=8), BOOKCASE (LABEL=10), BOARD (LABEL=11). "PN" MEANS POINTNET++, "RGCN" MEANS RESGCN-28, "RNET" MEANS RANDLA-NET, "SC" MEANS SOURCE CLASS."OOB/ACC" MEANS THE OUT-OF-BAND ACCURACY AND OVERALL ACCURACY WHILE "OOB/AIOU" IS SIMILAR.

Model	SC	L_2	PSR (%)	OOB/Acc (%)	OOB/aIoU (%)
PN	5	5.44	81.12	34.55 / 81.66	32.53 / 70.31
	6	3.78	42.85	88.72 / 94.67	52.42 / 66.60
	7	5.78	3.84	60.71 / 85.67	52.42 / 67.20
	8	3.02	1.04	65.33 / 85.24	42.25 / 67.20
	10	5.26	42.22	47.60 / 71.44	41.24 / 61.49
	11	6.85	70.58	74.55 / 89.23	48.37 / 63.41
RGCN	5	4.25	65.80	44.60 / 80.90	42.53 / 69.31
	6	4.16	26.27	65.76 / 88.03	65.72 / 79.88
	7	4.23	1.24	63.89 / 88.85	61.24 / 81.73
	8	4.35	0.90	62.00 / 93.02	59.73 / 88.02
	10	5.87	7.35	45.46 / 83.72	58.00 / 73.93
	11	3.83	26.20	67.59 / 91.15	64.87 / 84.67
RNet	5	3.86	82.42	81.50 / 84.42	44.93 / 50.48
	6	3.97	83.05	81.82 / 84.57	44.28 / 50.18
	7	4.42	80.95	80.85 / 84.55	44.14 / 51.20
	8	4.19	83.59	81.55 / 83.96	45.31 / 50.95
	10	3.99	91.87	82.88 / 84.66	47.38 / 51.28
	11	2.83	93.95	86.45 / 86.74	52.73 / 56.69

Finding 4: Under object hiding attack, norm-unbounded attack is also more effective.

Finding 5: Source class has big impact on the attack effectiveness, as changing the labels on the complex objects is much harder.

E. Evaluation on Outdoor Dataset

All previous evaluations are carried out on S3IDS, an indoor dataset. Since the outdoor scenes have different properties (e.g., different object classes and larger sizes), we evaluate the attacks against another outdoor dataset, Semantic3D. Only RandLA-Net is attacked because the other two PCSS models cannot handle the scale of point clouds in Semantic3D. We show the result of norm-unbounded attack only as it is more effective in previous experiments.

Table VI shows the results of the performance degradation attack. Similar as the attack against S3IDS, norm-unbounded attack greatly decreases the accuracy and aIoU comparing to the baseline (random noises) when they target the same L_2 distance: the average accuracy and the aIoU drop from 98.25% and 63.26% to 16.00% and 7.70%. The baseline only drops the average accuracy and the aIoU to 79.30% and 37.22%. We also observe that the result on RandLA-Net has bigger variance by samples: e.g., the accuracy can drop to 0% for the best case, and 90.82% for the worst case.

As for the object hiding attack, we manipulate the source points labeled as car (label=8) to mislead the model to predict them as man-made terrain (label=1), natural terrain (label=2), high vegetation (label=3) and low vegetation (label=4). From Table VII, PSR is near 95% when vegetation is the target class. Though outdoor scene is expected to be more complex, our result shows object hiding attack is still effective.

Finding 6: Norm-unbounded attack is also effective when attacking an outdoor scene, under both the performance degradation and the object hiding attacks.

F. Defense Methods

To mitigate the threats from the proposed adversarial attacks, gradient obfuscation, adversarial training, and anomaly

TABLE VI
THE RESULTS OF THE PERFORMANCE DEGRADATION ATTACK FROM RANDLA-NET ON SEMANTIC3D.

Case	Random Noise			Norm-unbounded		
	$\mathbf{L_2}$	**Acc** (%)	**aIoU** (%)	$\mathbf{L_2}$	**Acc** (%)	**aIoU** (%)
Best	19.31	0.00	0.00	19.31	0.00	0.00
Average	25.84	79.30(18.95↓)	37.22(26.04 ↓)	25.84	**16.00**(82.25↓)	**7.70**(55.56%↓)
Worst	280.79	100.0	100.0	280.79	90.82	25.42

TABLE VII
THE RESULTS OF THE OBJECT HIDING ATTACK AGAINST RANDLA-NET ON SEMANTIC3D. CAR (LABEL=8) IS PERTURBED TO MAN-MADE TERRAIN (LABEL=1), NATURAL TERRAIN (LABEL=2), HIGH VEGETATION (LABEL=3), LOW VEGETATION (LABEL=4).

Target Class	$\mathbf{L_2}$	**PSR**	**OOB Acc /Acc**(%)	**OOB aIoU /aIoU**(%)
Man-made terrain	10.41	85.30	73.03 / 73.64	30.74 / 33.56
Natural terrain	5.61	73.96	84.76 / 84.89	46.63 / 48.19
High vegetation	3.61	**94.26**	97.95 / 97.99	58.86 / 61.51
Low vegetation	3.60	**94.86**	74.18 / 74.70	39.57 / 42.52

detection can be tested on PCSS. These ideas have been initially examined on 2D image classification and were recently migrated to 3D point cloud object recognition. For gradient obfuscation, DUP-Net [65] includes a denoiser layer and upsampler network structure. GvG-PointNet++ [10] introduces gather vectors in the points clouds. Recently, Li et al. [24] proposed implicit gradients, which could lead the attackers to the wrong updating directions, to replace obfuscated gradients. For adversarial training, DeepSym [42] uses a sorting-based pooling operation to overcome the issues caused by the default symmetric function. Sun et al. [41] analyze the robustness of self-supervised learning 3D point cloud models with adversarial training. For anomaly detection, Yang et al. [58] and Rusu et al. [39] measured the statistics of point cloud to detect or mitigate the attacks.

Here, we measure how our attacks are impacted when the defenses are deployed, and we select the approaches under anomaly detection, as they are lightweight (e.g., adversarial training is heavyweight because it incurs high training overhead). We select two defense methods: Simple Random Sampling (SRS) [58] and Statistical Outlier Removal (SOR)[7] [65]. More specifically, SRS filters out a subset of points from a point cloud to mitigate the impact of perturbations while SOR removes the outlier points based on a k-NN distance. SRS can be directly used in semantic segmentation. For SOR, we revise the k-NN distance function by using both color and coordinate. The sampling number of SRS is 50 (around 1% of the whole point cloud number) and k is 2 for SOR. We follow the experiment setting from IF-Defense [54] and randomly select 100 point clouds to make a fair comparison. We evaluate our attacks on S3DIS with ResGCN-28 as the PCSS model.

As the results in Table VIII show, norm-bounded attack is still effective even when the two defense methods are applied (similar accuracy and aIoU with or without defenses). For norm-unbounded attack, as the L_2 distance cannot be fixed to a value, we try different attack parameters to make the L_2 distance fall in a similar range. It turns out in this case, the changes are more likely considered as outlier and detected by SOR (SOR reaches higher accuracy than SRS). Still none of the defenses are able to restore the accuracy and aIoU to the original levels, i.e., over 70%. The similar observation was also made in [54].

[7]We use the code from https://github.com/Wuziyi616/IF-Defense.

TABLE VIII
THE RESGCN-28 RESULT WITH SRS AND SOR UNDER PERFORMANCE DEGRADATION ATTACK.

Attack	**Defense**	L_2	**Acc** (%)	**aIoU** (%)
Norm-bounded	None	6.50	42.06	30.13
	SRS	6.57	46.06	36.18
	SOR	6.56	46.88	36.92
Norm-unbounded	None	4.30	6.85	3.79
	SRS	6.22	10.54	5.70
	SOR	9.44	41.48	29.86

Finding 7: The defenses based on anomaly detection are ineffective against norm-bounded attacks. Norm-unbounded attacks are affected more under Statistical Outlier Removal (SOR).

G. Attack Transferability

TABLE IX
THE UPPER ROW SHOWS THE RESULTS OF ATTACK TRANSFERABILITY ON POINTNET++. THE LOWER ROW DISPLAYS THE RESULTS FROM TRANSFERRING RESGCN-28 ADVERSARIAL SAMPLES TO POINTNET++.

PCSS Model	**Acc** (%)	**aIoU** (%)
PointNet++ (Pre-trained)	7.24	9.44
PointNet++ (Self-trained)	34.35	31.39
ResGCN-28	7.11	3.68
PointNet++	39.01	25.30

Existing research has shown an adversarial sample targeting 2D image classification has transferability [33], i.e., that a sample generated against one model is also effective against another model. We are interested in whether our adversarial samples have the same property. To this end, we first evaluate the attack transferability on models with different parameters. We select 400 adversarial samples generated by

norm-unbounded attack on the pre-trained PointNet++, and feed them into another PointNet++ trained by ourselves (the weights and biases are different). The results in Table IX show the accuracy and aIoU on the 400 samples, and both are less than 40%, suggesting our adversarial samples are transferable under different model parameters.

Then we test transferability across model families: we generate adversarial examples for ResGCN-28 and test if they can fool PointNet++. We do not transfer the attack against RandLA-Net due to its highly different approach of pre-processing. Due to different normalization steps (i.e., the coordinate ranges in $[-1, 1]$ for ResGCN-28 while $[0, 3]$ for PointNet++), the adversarial samples do not directly transfer, so we perform an extra step to map the attacked fields to the same range. We compute the accuracy and aIoU in the two settings, and the results suggest our adversarial examples are also transferrable (Table IX).

Finding 8: The adversarial example is transferable under different model parameters and across model families.

H. Visualization of Adversarial Examples

In this subsection, we visualize the adversarial examples generated under color-based norm-unbounded attack. For each sample, we show the original and perturbed scenes and their segmentation results.

First, we show the scenes in S3IDS under performance degradation attack and PointNet++ is the target model. We choose different types of scenes like the conference room, hallway, and lobby. From Figure 3, we can see the small perturbation generated by our attack leads to prominent changes in the segmentation results.

Next, we show an example about the object hiding attack in Figure 4. We set PointNet++ as the target model, and board as the source class. Since most of its points are classified as wall after the attack, our attack could make the board nearly "disappear" from the view of the segmentation model.

Finally, we show an example about an outdoor scene under Semantic3D in Figure 5, under performance degradation attack, with RandLA-Net as the targeted model. The visualized result also suggests seemingly small perturbations can drastically change the segmentation results.

VI. Discussion

Sub-sampling. Sub-sampling is a key technique in dealing with large-scale point cloud data, as described in Section II-A. Defenses can also leverage sub-sampling, as shown in Section V-F. We found when the sub-sampling is done by the PCSS models on the point cloud, such as farthest point sampling from PointNet++, k-neighbor Sampling from ResGCN and random sampling from RandLA-Net, our attacks are still effective. When the sub-sampling is done before PCSS, e.g., storing a fraction of video frames from camera [45], our attacks could be impacted if the sampling procedure is unknown to the attacker, as the adversarial input cannot be directly constructed from the perturbed point cloud. When sub-sampling is done as a defense, different sampling methods have different effectiveness (e.g., SOR is more effective against norm-unbounded attacks).

Other models. We select three representative PCSS models to attack. Section II-A overviews the other types of models. We expect our attacks are applicable to the models which generate gradients. One example is Point Cloud Transformer (PCT), which captures the context of a point with the Transformer architecture (e.g., through positional encoder and self-attention) [14]. PCT still computes gradients and recent works showed that Vision Transformer (ViT) in the 2D image domain is vulnerable under perturbation [1], [51].

Limitations. (1) Currently we evaluate the attacks on two datasets. Admittedly we could extend the study scope by including more datasets. (2) For RandLA-Net, we did not implement the coordinate-based attack as the its point sampling mechanism makes it more difficult to locate the points for attack. (3) The focus of this study is to examine the robustness of different PCSS models and settings. Unlike previous papers [6], [61], [66], we did not convert the perturbation on the point cloud into the changes in the physical world, e.g., using irregular objects or stickers. (4) Our attacks target one point cloud at a time. In the real-world autonomous driving setting, a sequence of point clouds needs to be processed by a PCSS model, so the attacker should consider how to sufficiently attack multiple point clouds. Previous studies on the 2D image domain show that an attacker can add the same perturbation on multiple images, after assigning different weights on each image [49]. We expect a similar approach can be applied on 3D point clouds. (5) Among the point cloud features, we only attack the color feature because it provides more information than the others (e.g., the intensity feature of Semantic3D). The distance function described in Section IV-A might need to be changed for other features.

VII. Conclusion

In this work, we present the first comparative study of adversarial attacks on 3D point cloud semantic segmentation (PCSS). We systematically formulate the attacker's objectives under the object hiding attack and the performance degradation attack, and develop two attack methods based on norm-bounded attack and norm-unbounded attack. In addition to the point coordinates that are exploited by all existing adversarial attacks, we consider point features to be perturbed. We examine these attack combinations on an indoor dataset S3IDS and an outdoor dataset Semantic3D dataset, to examine the impact of each attack option. Overall, we found all examined PCSS models are vulnerable under adversarial perturbation, in particular to norm-unbounded attack that is applied on the color features. We hope with this study, more efforts can be make to improve the robustness of PCSS models.

Acknowledgment

We thank the valuable comments from the anonymous reviewers and our shepherd. The authors also thank Lijie Huang from the UCInspire program for the help. The authors from the

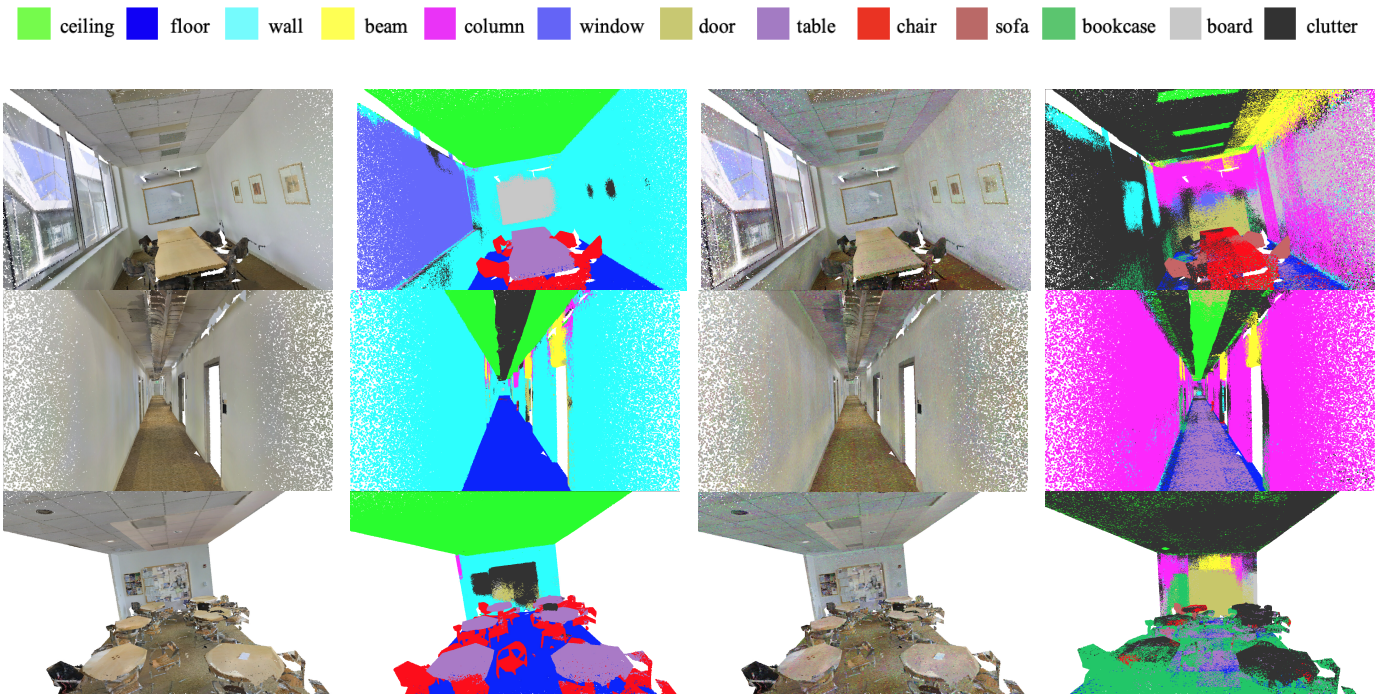

Fig. 3. The Performance degradation attack with Conference room 1 (first row), Hallway 2 (second row), Lobby 1 (third row) of Area 5 in S3DIS. The first to fourth columns show the original scene, the original segmentation results, perturbed scene and perturbed segmentation results.

Fig. 4. The object hiding attack with Office 33 of Area 5 in S3DIS. The upper left, upper right, lower left, and lower right show the original scene, the perturbed scene, and the segmentation results of the original and perturbed scene.

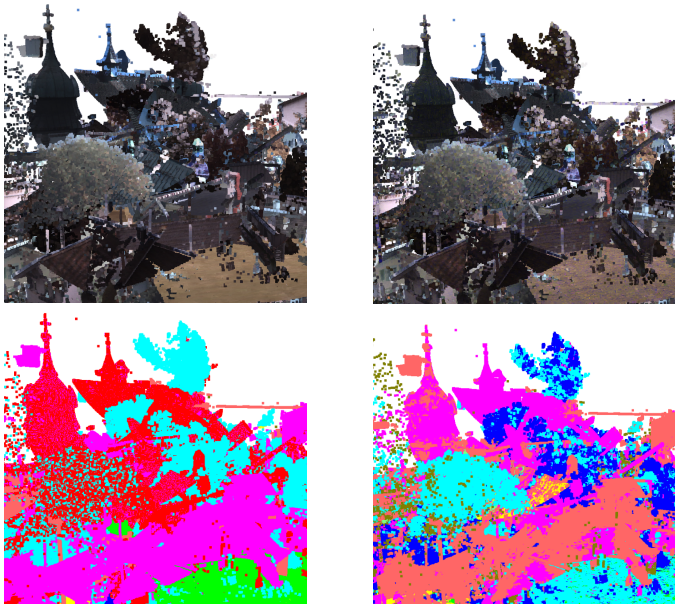

Fig. 5. An example of performance degradation attack against Semantic3D.

United States are supported under NSF 2039634 and 2124039. The author from China is supported under National Key R&D Program of China (Grant No.2022YFB3102901) and the Natural Science Foundation of Shanghai (No. 23ZR1407100).

REFERENCES

[1] Ahmed Aldahdooh, Wassim Hamidouche, and Olivier Deforges. Reveal of vision transformers robustness against adversarial attacks. *arXiv preprint arXiv:2106.03734*, 2021.

[2] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017.

[3] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1534–1543, 2016.

[4] Anurag Arnab, Ondrej Miksik, and Philip HS Torr. On the robustness of semantic segmentation models to adversarial attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 888–897, 2018.

[5] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven

Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9297–9307, 2019.

[6] 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 *2021 IEEE Symposium on Security and Privacy (SP)*, pages 176–194. IEEE, 2021.

[7] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

[8] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[9] Siheng Chen, Baoan Liu, Chen Feng, Carlos Vallespi-Gonzalez, and Carl Wellington. 3d point cloud processing and learning for autonomous driving: Impacting map creation, localization, and perception. *IEEE Signal Processing Magazine*, 38(1):68–86, 2020.

[10] Xiaoyi Dong, Dongdong Chen, Hang Zhou, Gang Hua, Weiming Zhang, and Nenghai Yu. Self-robust 3d point recognition via gather-vector guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11516–11524, 2020.

[11] Oren Dovrat, Itai Lang, and Shai Avidan. Learning to sample. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2760–2769, 2019.

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[13] Amira Guesmi, Khaled N Khasawneh, Nael Abu-Ghazaleh, and Ihsen Alouani. Room: Adversarial machine learning attacks under real-time constraints. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE, 2022.

[14] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7:187–199, 2021.

[15] Timo Hackel, Nikolay Savinov, Lubor Ladicky, Jan D Wegner, Konrad Schindler, and Marc Pollefeys. Semantic3d. net: A new large-scale point cloud classification benchmark. *arXiv preprint arXiv:1704.03847*, 2017.

[16] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11108–11117, 2020.

[17] Qidong Huang, Xiaoyi Dong, Dongdong Chen, Hang Zhou, Weiming Zhang, and Nenghai Yu. Shape-invariant 3d adversarial point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15335–15344, 2022.

[18] Zizhi Jin, Ji Xiaoyu, Yushi Cheng, Bo Yang, Chen Yan, and Wenyuan Xu. Pla-lidar: Physical laser attacks against lidar-based 3d object detection in autonomous vehicle. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 710–727. IEEE Computer Society, 2023.

[19] Jaeyeon Kim, Binh-Son Hua, Duc Thanh Nguyen, and Sai-Kit Yeung. Minimal adversarial examples for deep learning on 3d point clouds. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7777–7786. IEEE Computer Society, 2021.

[20] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[21] Loic Landrieu and Simonovsky Martin. Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018)*, Salt Lake City, United States, June 2018.

[22] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9267–9276, 2019.

[23] Jiaxin Li, Ben M Chen, and Gim Hee Lee. So-net: Self-organizing network for point cloud analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9397–9406, 2018.

[24] Kaidong Li, Ziming Zhang, Cuncong Zhong, and Guanghui Wang. Robust structured declarative classifiers for 3d point clouds: Defending adversarial attacks with implicit gradients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15294–15304, 2022.

[25] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on χ-transformed points. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 828–838, 2018.

[26] Binbin Liu, Jinlai Zhang, and Jihong Zhu. Boosting 3d adversarial attacks with attacking on frequency. *IEEE Access*, 10:50974–50984, 2022.

[27] Daniel Liu, Ronald Yu, and Hao Su. Extending adversarial attacks and defenses to deep 3d point cloud classifiers. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 2279–2283. IEEE, 2019.

[28] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[29] Kirill Mazur and Victor Lempitsky. Cloud transformers. *arXiv preprint arXiv:2007.11679*, 2020.

[30] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 909–918, 2019.

[31] Krishna Kanth Nakka and Mathieu Salzmann. Indirect local attacks for context-aware semantic segmentation networks. In *European Conference on Computer Vision*, pages 611–628. Springer, 2020.

[32] Federico Nesti, Giulio Rossolini, Saasha Nair, Alessandro Biondi, and Giorgio Buttazzo. Evaluating the robustness of semantic segmentation for autonomous driving against real-world adversarial patch attacks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2280–2289, 2022.

[33] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

[34] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael P Wellman. Sok: Security and privacy in machine learning. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 399–414. IEEE, 2018.

[35] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[36] 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*, pages 5099–5108, 2017.

[37] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Abed Al Kader Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *arXiv preprint arXiv:2206.04670*, 2022.

[38] Damien Robert, Bruno Vallet, and Loic Landrieu. Learning multi-view aggregation in the wild for large-scale 3d semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5575–5584, 2022.

[39] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Mihai Dolha, and Michael Beetz. Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927–941, 2008.

[40] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 3d point cloud generative adversarial network based on tree structured graph convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3859–3868, 2019.

[41] 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. *Advances in Neural Information Processing Systems*, 34:15498–15512, 2021.

[42] Jiachen Sun, Karl Koenig, Yulong Cao, Qi Alfred Chen, and Z Morley Mao. On the adversarial robustness of 3d point cloud classification. *arXiv preprint arXiv:2011.11922*, 2020.

[43] Liyao Tang, Yibing Zhan, Zhe Chen, Baosheng Yu, and Dacheng Tao. Contrastive boundary learning for point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8489–8499, 2022.

[44] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the*

IEEE/CVF International Conference on Computer Vision, pages 6411–6420, 2019.

[45] Matthew Tomei, Alexander Schwing, Satish Narayanasamy, and Rakesh Kumar. Sensor training data reduction for autonomous vehicles. In *Proceedings of the 2019 Workshop on Hot Topics in Video Analytics and Intelligent Edges*, pages 45–50, 2019.

[46] Tzungyu Tsai, Kaichen Yang, Tsung-Yi Ho, and Yier Jin. Robust adversarial objects against deep learning models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 954–962, 2020.

[47] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13716–13725, 2020.

[48] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1588–1597, 2019.

[49] Jingkang Wang, Tianyun Zhang, Sijia Liu, Pin-Yu Chen, Jiacen Xu, Makan Fardad, and Bo Li. Adversarial attack generation empowered by min-max optimization. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 16020–16033. Curran Associates, Inc., 2021.

[50] Jiacheng Wei, Guosheng Lin, Kim-Hui Yap, Tzu-Yi Hung, and Lihua Xie. Multi-path region mining for weakly supervised 3d semantic segmentation on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[51] Zhipeng Wei, Jingjing Chen, Micah Goldblum, Zuxuan Wu, Tom Goldstein, and Yu-Gang Jiang. Towards transferable adversarial attacks on vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2668–2676, 2022.

[52] Matthew Wicker and Marta Kwiatkowska. Robustness of 3d deep learning in an adversarial setting. *CoRR*, abs/1904.00923, 2019.

[53] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[54] Ziyi Wu, Yueqi Duan, He Wang, Qingnan Fan, and Leonidas J Guibas. If-defense: 3d adversarial point cloud defense via implicit function based restoration. *arXiv preprint arXiv:2010.05272*, 2020.

[55] Chong Xiang, Charles R Qi, and Bo Li. Generating 3d adversarial point clouds. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2019.

[56] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1369–1378, 2017.

[57] Qiangeng Xu, Xudong Sun, Cho-Ying Wu, Panqu Wang, and Ulrich Neumann. Grid-gcn for fast and scalable point cloud learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5661–5670, 2020.

[58] Jiancheng Yang, Qiang Zhang, Rongyao Fang, Bingbing Ni, Jinxian Liu, and Qi Tian. Adversarial attack and defense on point sets. *arXiv preprint arXiv:1902.10899*, 2019.

[59] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. Modeling point clouds with self-attention and gumbel subset sampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3323–3332, 2019.

[60] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021.

[61] Yue Zhao, Hong Zhu, Ruigang Liang, Qintao Shen, Shengzhi Zhang, and Kai Chen. Seeing isn't believing: Towards more robust adversarial attack against real world object detectors. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 1989–2004, 2019.

[62] Tianhang Zheng, Changyou Chen, Junsong Yuan, Bo Li, and Kui Ren. Pointcloud saliency maps. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1598–1606, 2019.

[63] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[64] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10356–10365, 2020.

[65] 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 *Proceedings of the IEEE International Conference on Computer Vision*, pages 1961–1970, 2019.

[66] Yi Zhu, Chenglin Miao, Foad Hajiaghajani, Mengdi Huai, Lu Su, and Chunming Qiao. Adversarial attacks against lidar semantic segmentation in autonomous driving. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, pages 329–342, 2021.