Explainability-Aware One Point Attack for Point Cloud Neural Networks

Hanxiao Tan Helena Kotthaus
AI Group, TU Dortmund
{hanxiao.tan,helena.kotthaus}@tu-dortmund.de

Abstract

Recent studies have shown an increased interest to investigate the reliability of point cloud networks by adversarial attacks. However, most of the existing studies aim to deceive humans, while few address the operation principles of the models themselves. In this work, we propose two adversarial methods: One Point Attack (OPA) and Critical Traversal Attack (CTA), which target the points crucial to predictions more precisely by incorporating explainability methods. Our results show that popular point cloud networks can be deceived with high success rate by shifting only one point from the input instance. We also show the interesting impact of different point attribution distributions on the adversarial robustness of point cloud networks. We discuss how our approaches facilitate the explainability study for point cloud networks. To the best of our knowledge, this is the first point-cloud-based adversarial approach concerning explainability. Our code is available at `https://github.com/Explain3D/Exp-One-Point-Atk-PC`.

1. Introduction

Developments in the field of autonomous driving and robotics have heightened the need for the research of point cloud (PC) data since PCs are advantageous over other 3D representations for real-time performance. However, compared with 2D images, the robustness and reliability of PC networks have only attracted considerable attention in recent years and still not been sufficiently studied, which potentially threatens human lives e.g. driverless vehicles with point cloud recognition systems are unreliable unless they are sufficiently stable and transparent.

Several attempts have been made to investigate the adversarial attacks on PC networks, e.g. [22] and [54]. The first series (*shape-perceptible*), represented by [22], although produce geometrically continuous adversarial examples with external generative models, fundamentally aim at deceiving the human eyes and therefore ignore the constraints on the perturbation dimensions. Another series (*point-shifting*) represented by [54] have shown a possibility that moving (dropping or adding) points at crucial positions can successfully fool the classifier. Nevertheless, most of such studies have only focused on minimizing perturbation distances for imperceptibility. Conversely, we start from a different perspective by exploring attacks on PC networks with a minimal number of perturbed points. Additionally, we argue that existing choices of critical points could be further optimized incorporating explainability approaches.

In comparison to previous studies, our work is motivated by the following reasons:

Model operating principle: Part of the point-shifting methods also deceived the classifier by perturbing critical points, however, we argue that their selection of critical points is flawed. Since most of the selection methods for critical points are based on gradients only, and existing studies [1, 45] have demonstrated that raw gradients suffer from saturation issues and are therefore biased. On the other hand, [16] demonstrated that feature attributions for PC classification networks are extremely sparse, while no work has specifically studied how these attributions are distributed among the critical points as well as their impact on the prediction sensitivity.

Potential for explainability: Another possibility of one-dimensional perturbations is explainability. The explainability method called counterfactuals alters the prediction label by perturbing the input features to provide a convincing explanation to the users. Previous researches have documented that humans are more receptive to counterfactuals with sparse-dimensional perturbations [18, 28, 3]. For high-dimensional decision boundaries like point clouds, reduction of perturbation dimension is an important way to enhance the comprehensibility of the vicinity, which can be regarded as "cutting the input space using very low-dimensional slices" [43]. Furthermore, by incorporating part semantics, Our approach has the potential to be extended for generating high-quality counterfactuals. Moreover, ours require only the access of gradients and no additional generative models, and are therefore more intrinsically explainable.

Altogether, the contribution of this work can be summa-

rized as follows:

- We propose two explainability method-based adversarial attacks: One Point Attack (OPA) and Critical Traversal Attack (CTA). Incorporating the attribution from explainable AI, our methods fool the popular PC networks with high success rate. Supported by extensive experiments, a significant margin is established with existing approaches in terms of the perturbation sparsity.
- We investigate diverse pooling architectures as alternatives to existing point cloud networks, which have an impact on the internal vulnerability against critical points shifting.
- We discuss the research potential of adversarial attacks from an explainability perspective, and present an application of our methods on facilitating the evaluation of explainability approaches.

The rest of the paper is organized as follows: We introduce the related researches of PC attacks in Sec. 2, then we detailed our proposed methods in Sec. 3. In Sec. 4, we present the visualization of the adversarial examples and demonstrate comparative results with existing studies. In Sec. 5 we discuss interesting observations derived from experiments with respect to robustness and explainability. We finally summarize our work in Sec. 6.

2. Related Work

As the first work [47] on adversarial examples was presented, an increasing variety of attacks against 2D image neural networks followed [14, 8, 21, 32, 11, 29]. However, due to the structural distinctions with PC networks (see Supplementary Sec. 7.1.1), we do not elaborate on the attack methods of image deep neural networks (DNN)s. Relevant information about image adversarial examples refers to [2]. It is notable that [43] investigated one-pixel attack for fooling image DNNs and also aimed at exploring the boundary of inputs. Nevertheless, their approach is a black-box attack based on an evolutionary algorithm, which is essentially distinct from ours.

Existing PC attacks are generally categorized into two classes: (i) *Shape-perceptible* generation, which generates human-recognizable adversarial examples with consecutive surfaces or meshes via generative models or spacial geometric transformations [55, 22, 51, 23, 17, 57, 25, 59]. (ii) *Point-shifting* perturbation, which regularize the distance or dimension of the point-wise shifting via perturbing or gradient-aware white-box algorithms [19, 58, 52, 54, 44, 24]. Point-wise perturbations, especially gradient-aware attack methodologies, enable more intrinsic explorations of the model such as stabilities and decision boundaries. On the other hand, from the perspective of explainability, the majority of generative models contain complex network structures that are inherently unexplainable. Utilizing their output to interpret another model is counter-intuitive.

The conception "**critical points**" has been discussed by several previous studies as well as the PointNet proposer [33], which forms the skeletons of the input instances in the classification processes. Existing methods [54, 19, 58, 52] extract the critical points by tracing the ones that remain active from the pooling layer, or by observing the gradient-based saliency maps. While such approaches succeed in generating adversarial examples with minor perturbation distances and sparse shift dimensions, we argue that their modules for selecting critical points can be further optimized. Due to the subsequent FC layers, it is difficult to determine whether the surviving points from the pooling layer conclusively make significant contributions to the prediction. Besides, saliency maps based on raw gradients are defective [1, 45]. The above factors may result in the involvement of fake critical points or omission of real ones during the perturbation process, which severely impairs the performance of the adversarial algorithms.

Explainability has been gaining attention in recent years. Popular explainability methods can be broadly categorized into gradient-based [38, 46, 4, 37, 40, 41], which requires the access of the gradient information, and perturbation-based [35, 26, 36], which is model-agnostic. In addition, counterfactual explanations [7] is proposed for tabular data by modifying selected features to induce the model to make different predictions. The properties of counterfactuals are identical to the adversarial examples, therefore attack methods may possess similar explainability potentials [5].

3. Methods

In this section, we formulate the adversarial problem in general and introduce the critical points set (Subsec. 3.1). We present our new attack approaches (Subsec. 3.2).

3.1. Problem Statement

Let $P \in \mathbb{R}^{n\times d}$ denotes the given point cloud instance, $f : P \rightarrow y$ denotes the chosen PC neural network and $M(a, b) : \mathbb{R}^{n_a\times d} \times \mathbb{R}^{n_b\times d}$ denotes the perturbation matrix between instance a and b. The goal of this work is to generate an adversarial examples $P' \in \mathbb{R}^{n'\times d}$ which satisfies:

$$\begin{aligned} argmin(\,|\{m \in M(P, P'))|m \neq 0\}\,| \\ + \|M(P, P'))\|) : f(P') \neq f(P) \end{aligned} \tag{1}$$

Note that among the three popular attack methods for PC data: point adding ($n' > n$), point detaching ($n' < n$) and point shifting ($n' = n$), this work considers point shifting only.

We address the adversarial task in equation 1 as a gradient optimization problem. We minimize the loss on the input PC instance while freezing all parameters of the network:

$$L = \alpha \times Z[f(P)] + \beta \times D(P, P') \tag{2}$$

where α indicates the optimization rate, $Z[f(P)]$ indicates the neuron unit corresponding to the prediction $f(P)$ which guaranties the alteration of prediction, $D(P, P')$ represents the quantized imperceptibility between the input P and the adversarial example P' and β is the distance penalizing weight. The imperceptibility has two major components, namely the perturbation magnitude and the perturbation sparsity. The perturbation magnitude can be constrained in three ways: Chamfer distance (equation 3), Hausdorff distance (equation 4) or simply Euclidean distance. We ensure perturbation sparsity by simply masking the gradient matrix, and with the help of the saliency map derived by the explainability method we only need to shift those points that contribute positively to the prediction to change the classification results, which are termed as "critical points set".

Critical points set: The concept was first discussed by its proposer [33], which contributes to the features of the max-pooling layer and summarizes the skeleton shape of the input objects. They demonstrated an *upper-bound* construction and proved that corruptions falling between the *critical set* and the *upper-bound* shape pose no impact on the predictions of the model. However, the impairment of shifting those critical points is not sufficiently discussed. Previous adversarial researches studied the model robustness by perturbing or dropping critical points set identified through monitoring the max-pooling layer or accumulating loss of gradients [54, 19, 58, 52]. Nevertheless, capturing the output of the max-pooling layer struggles to identify the real critical points set due to the lack of transparency in the high-level structures (e.g., multiple MLPs following the pooling layer), while saliency maps based on raw gradients suffer from saturation [1, 45], both of which severely compromise the filtering of the critical point set. We therefore introduce Integrated Gradients (IG) [46], the state-of-the-art gradient-based explainability approach, to further investigate the sensitivity and robustness to the critical points set. The formulation of IG is summarized in equation S1.

Similarity metrics for point cloud data: Due to the irregularity of PCs, Manhattan and Euclidean distance are both no longer applicable when measuring the similarity between PC instances. Several previous works introduce Chamfer [19, 56, 54, 22, 25, 59, 55] and Hausdorff [60, 19, 56, 54, 25, 59] distances to represent the imperceptibility of adversarial examples. The measurements are formulated as:

- Bidirectional Chamfer distance

$$\begin{aligned} D_c(P_a, P_b) = &\frac{1}{|P_a|} \sum_{p_m \in P_a} \min_{p_n \in P_b} \|p_m - p_n\|_2 \\ &+ \frac{1}{|P_b|} \sum_{p_n \in P_b} \min_{p_m \in P_a} \|p_n - p_m\|_2 \end{aligned} \tag{3}$$

- Bidirectional Hausdorff distance

$$\begin{aligned} D_h(P_a, P_b) = \max(&\max(\min_{p_n \in P_b} \|p_m - p_n\|_2), \\ &max(\min_{p_m \in P_a} \|p_n - p_m\|_2)) \end{aligned} \tag{4}$$

3.2. Attack Algorithms

One-Point Attack (OPA): Specifically, OPA (see algorithm 1 for pseudo-code) is an extreme of restricting the number of perturbed points, which requires:

$$|\{m \in M(P, P'))|m \neq 0\}| = 1 \tag{5}$$

We acquire the gradients that minimize the activation unit corresponding to the original prediction, and a saliency map based on the input PC instance from the explanation generated by IG. We sort the saliency map and select the point with the top-n attribution as the critical points ($n = 1$ for OPA), and mask all points excluding the critical one on the gradients matrix according to its index. Subsequently the critical points are shifted with an iterative optimization process. An optional distance penalty term can be inserted into the optimization objective to regularize the perturbation magnitude and enhance the imperceptibility of the adversarial examples. We choose Adam [20] as the optimizer, which exhibits better performance for optimization experiments. The optimization process may stagnate by falling into a local optimum, hence we treat every 25 steps as a recording period, and the masked Gaussian noise weighted by W_n is introduced into the perturbed points when the average of the target activation at period $k + 1$ is greater than at period k. For the consideration of time cost, the optimization process is terminated when certain conditions are fulfilled and the attack to the current instance is deemed as a failure.

Critical Traversal Attack (CTA): Due to the uneven vulnerability of different PC instances, heuristically setting a uniform sparsity restriction for the critical points perturbation is challenging. CTA (pseudo-code presented in algorithm 2) enables the constraint of perturbation sparsity to be adaptive by attempting the minimum number of perturbed points for each instance subject to a successful attack. The idea of CTA is starting with the number of perturbed points n as 1 and increasing by 1 for each local failure until the prediction is successfully attacked or globally failed. Similarly, we consider the saliency map generated by IG as the selection criterion for critical points, and the alternative perturbed points are incremented from top-1 to all positively

attributed points. Again, for accelerating optimization we also select Adam [20] as the optimizer. Since most PC instances can be successfully attacked by one-point shifting through the introduction of Gaussian noise in the optimization process, we discarded the noise-adding mechanism in CTA to distinguish the experiment results from OPA. The aforementioned local failure stands for terminating the current n-points attack and starting another $n+1$ round, while the global failure indicates that for the current instance the attack has failed. We detail the stopping criteria for OPA and CTA in Sec. 7.2.1.

Algorithm 1: N_p-critical Point(s) Attack. ($N_p = 1$ for OPA)

Input: $P \rightarrow N \times D$ PC data, $f \rightarrow$ PC neural network, $\alpha \rightarrow$ Optimizing rate, $\beta \rightarrow$ Weight for constrain the perturbing distance(optional), $D \rightarrow$ Distance calculating function(optional), $N_p \rightarrow$ Number of shifting points(1 for *One-point attack*), $W_n \rightarrow$ Gaussian noise weights

Output: $P_{adv} \rightarrow N \times D$ Adversarial example

$A_{idx} = Argsort(IG(P, f))$ // Get IG mask of P
$R_s = list()$ // Activation Recorder
$I_{cur} = 1$ // Current iteration
while *True* **do**
  $a_p \leftarrow f(P)$ // Current activation of predicted class
  $G = \alpha * A_p + \beta * D(P_{adv}, P)$ // Add distance constrain(Optional)
  $P_{adv} = Adam(P_{adv}, G[A_{idx}[0 : N_p]])$ // Adam optimizing N points
  $I_{cur} \mathrel{+}= 1$
  $R_s.append(a_p)$
  /* Add masked Gaussian random noise if activation descending stopped */
  if $R_s[t] < R_s[t+1]$ **then**
    $P_{adv} \mathrel{+}= W_n \times GaussianRandom(P_\delta)[A_{idx}[0 : N_p]]$
  /* Success if predict class changed */
  if $max(a)! = pred$ **then**
    return P_{adv}
  /* Fails if the stopping conditions related to R_a and I_{cur} are fulfilled */
  if *Stopping criteria are fulfilled* **then**
    return *Failed*

Algorithm 2: Critical Traversal Attack (CTA)

Input: $P \rightarrow N \times D$ PC data, $f \rightarrow$ PC neural network, $\alpha \rightarrow$ Optimizing rate, $\beta \rightarrow$ Weight for constrain the perturbing distance(optional), $D \rightarrow$ Distance measuring function(optional)

Output: $P_{adv} \rightarrow N \times D$ Adversarial example

$A_{idx} = Argsort(IG(P, f))$ // Get IG mask of P
$Num_{pos} = count(IG(P, f) > 0)$ // # Points with attribution >0
$R_s = list()$
$I_{cur} = 1$
for N_p *from* 1 *to* Num_{pos} **do**
  while *True* **do**
    $a_p \leftarrow f(P)$ // Activation of predicted class
    $G = \alpha * A_p + \beta * D(P_{adv}, P)$ // Add distance constrain(Optional)
    $P_{adv} = Adam(P_{adv}, G[A_{idx}[0 : N_p]])$ // Adam optimizing N points
    $I_{cur} \mathrel{+}= 1$
    $R_s.append(a_p)$
    /* Success if predict class changed */
    if $argmax(a_p)! = pred$ **then**
      return P_{adv}
    /* Current N_p round fails if the local stopping conditions related to R_a and I_{cur} are fulfilled */
    if ***Local stopping criteria*** *fulfilled* **then**
      break;
  /* Current instance fails if the global stopping conditions are fulfilled */
  if ***Global stopping criteria*** *fulfilled* **then**
    return *Failed*
  return *Failed*

4. Experiments

In this section, we present and analyze the results of the proposed attack approaches. We demonstrate quantitative adversarial examples in Subsec. 4.2 and scrutinize the qualitative result in Subsec. 4.1. Our experiments[1] are primarily conducted on PointNet [33], which in general achieves an overall accuracy of 87.1% for the classification task on ModelNet40. Moreover, we extended our approaches on the most popular PC network PointNet++ [34] and DGCNN [50], which outperform the PC classification task with 90.7% and 92.2% accuracies respectively. We also experiment on PointMLP [27], the state-of-the-art classification model to date, which achieves the best accuracy of 94.5% on ModelNet40. Modelnet40 [53], our main experimental dataset, contains 12311 CAD models (9843 for training and 2468 for evaluation) from 40 common cate-

[1]Our code is available at `https://github.com/Explain3D/Exp-One-Point-Atk-PC`

gories, and is currently the most widely-applied point cloud classification dataset. We randomly sampled 25 instances for each class from the test set, and then selected those instances that are correctly predicted by the model as our victim samples. When configuring parameters, the optimization rate α is empirically set to 10^{-6}, which performs as the most suitable step size for PointNet after grid search. Specifically for OPA, we set the Gaussian weight W_n to 10^{-1}, which proved to be the most suitable configuration. For CTA, we investigate both $\beta = 0$ and $1e-3$. More analytical results of different configuration of β and W_n is demonstrated in Fig. S5. We also validate our methods on ShapeNet [9] dataset. All attacks performed in this section are non-targeted unless specifically mentioned. In all experiments, we only compare the performances among **point-shifting** attacks, motivated by exploring the peculiarities of PC networks. Though previous shape-perceptible approaches such as [57, 17, 59, 22, 55] also addressed adversarial studies of PCs, they were devoted to generate adversarial instances with human-perceptible geometries. Therefore, comparison of perturbation distances and dimensions with their works is not relevant.

4.1. Quantitative evaluations and comparisons

In this section, we compare the imperceptibility of proposed methods with existing attacks via measuring Hausdorff and Chamfer distances as well as the number of shifted points, and demonstrate their transferability among different popular PC networks. Additionally, we show that CTA maintains a remarkably high success rate even after converting to targeted attacks.

Imperceptibility: We compare the quality of generated adversarial examples with other **point-shifting** researches under the aspect of success rate, Chamfer and Hausdorff distances, and the number of points perturbed. As Tab. 1 shows, compared to the approaches favoring to restrict the perturbation magnitude, despite the relative laxity in controlling the distance between the adversarial examples and the input instances, our methods prevail significantly in terms of the sparsity of the perturbation matrix. Simultaneously, our methods achieve a higher success rate, implying that the network can be fooled for almost all PC instances by shifting a minuscule amount of points (even one). Note that while calculating D_c and D_h, we employ the L2-norm. Therefore, despite the large Hausdorff distance, the average perturbation magnitude along each axis is $\mathbf{0.488}$. Considering that each axis of ModelNet40 is regularized into the interval $[-1, 1]$, this magnitude occupies $\mathbf{24.4}\%$ of the interval, which corresponds to an average perturbation of 62.2 gray values in 2D grayscale images. We thus consider the perturbation magnitude to be acceptable (also see Sec. 7.4 for legitimacy check).

To eliminate potential bias, we also test the proposed attack methods with ShapeNet [9] dataset. As Tab. 2 presents, our approaches perform similarly on the two different datasets, and therefore the vulnerable bias in the data distribution of ModelNet40 can be basically excluded.

In addition to PointNet, we also tested the performance of our proposed methods on PC networks with different architectures. Tab. 3 summarize the result of attack PointNet, PointNet++, DGCNN and PointMLP with both OPA and CTA respectively. Both OPA and CTA achieve high success rate fooling those networks while only a single-digit number of points are shifted. PointMLP seems to be the most stable model, and we speculate that this is attributed to the affine module of relative position [27]. Intuitively, PC neural networks appear to be more vulnerable compared to image CNNs ([43] is a roughly comparable study since they also performed one-pixel attack with the highest success rate of 71.66%, see Tab. S3 and Fig. S7 in supplementary for results of OPA). An opposite conclusion has been drawn by [54], they trained the PointNet with 2D data and compared its robustness with 2D CNNs against adversarial images. Nevertheless, we argue that the adversarial examples are generated by attacking a 2D CNN, such attacks may not be aggressive for PointNet, which is specifically designed for PCs.

Transferability: We investigate the transferability of proposed attacks across different PC networks by feeding the adversarial examples generated by one network to the others and recording the success rate. Fig. 1 presents the adversarial transferability between PointNet, PointNet++, DGCNN and PointMLP. What stands out in the figure is that PointNet++, DGCNN, PointMLP show strong stability against the adversarial examples from PointNet. We believe this is because the aggregated adjacency features disperse the attribution of a single point. Recall the *EdgeConv* [50] in DGCNN, which extracts adjacent features in both point and latent spaces, while PointNet++ possesses a similar module that aggregates neighboring points [33], which can be considered as a point-space-only *EdgeConv*. Such an integration distributes the feature contribution to multiple adjacent points, and a modest shifting of one point has limited impacts on the aggregated cluster. For PointMLP, this module transforms into affines of relative-position encodings [27]. Despite the involvement of adjacent points information, the features of relative positions may be severely corrupted if the centroids are perturbed. However, the feature extractor in PointNet can also be regarded as a special *EdgeConv* with $K = 1$, preserving the location information of the central point only, and therefore is more sensitive to the perturbation. Surprisingly, PointNet++ performs stably against adversarial examples from DGCNN and PointMLP, while the opposite fails. We consider the stability of PointNet++ stems from the *multi-scale(resolution) grouping*, where latent features are concatenated by grouping layers at differ-

	S(↑)	$D_c(\downarrow)$	$D_h(\downarrow)$	$N_p(\downarrow)$
L_p Norm [54]	85.9	1.77×10^{-4}	2.38×10^{-2}	967
Minimal selection [19]	89.4	1.55×10^{-4}	1.88×10^{-2}	36
Adversarial sink [25]	88.3	7.65×10^{-3}	1.92×10^{-1}	1024
Adversarial stick [25]	83.7	4.93×10^{-3}	1.49×10^{-1}	210
Random selection [52]	55.6	7.47×10^{-4}	2.49×10^{-3}	413
Critical selection [52]	19.0	$\mathbf{1.15 \times 10^{-4}}$	9.39×10^{-3}	50
Critical frequency [58]	63.2	5.72×10^{-4}	2.50×10^{-3}	303
Saliency map/L [58]	56.0	6.47×10^{-4}	2.50×10^{-3}	358
Saliency map/H [58]	58.4	7.52×10^{-4}	$\mathbf{2.48 \times 10^{-3}}$	424
Ours (OPA)	98.7	8.64×10^{-4}	8.45×10^{-1}	**1**
Ours (CTA$_{\beta=0}$)	**100**	8.92×10^{-4}	8.19×10^{-1}	2
Ours (CTA$_{\beta=1e-3}$)	99.6	7.73×10^{-4}	6.68×10^{-1}	6

Table 1. Comparison of existing point-shifting adversarial generation approaches for PointNet, where S, D_c, D_h and N_p denote the success rate, Chamfer and Hausdorff distances and the number of shifted points respectively. Part of the records sourced from [19]. It is worth noting that we only compare the gradient-based **point-shifting** competitors. The upward (↑) and downward (↓) arrows indicate whether a larger or smaller value is better, respectively.

	Dataset	S(↑)	$D_c(\downarrow)$	$D_h(\downarrow)$	$N_p(\downarrow)$
OPA	ModelNet40	98.7	8.64×10^{-4}	8.45×10^{-1}	1
	ShapeNet	95.1	8.39×10^{-4}	8.06×10^{-1}	1
CTA	ModelNet40	100	8.92×10^{-4}	8.19×10^{-1}	2
	ShapeNet	100	8.91×10^{-4}	7.26×10^{-1}	3

Table 2. Comparison of attack results with ModelNet40 and ShapeNet dataset.

	Model	$S(\uparrow)$	$D_c(\downarrow)$	$D_h(\downarrow)$	$N_p(\downarrow)$
O P A	PN [33]	98.7	8.45×10^{-4}	8.64×10^{-1}	1
	PN++ [34]	99.1	1.58×10^{-2}	1.61×10^{1}	1
	DGCNN [50]	90.9	1.70×10^{-3}	1.69	1
	PointMLP [27]	52.9	1.91×10^{-3}	1.90	1
C T A	PN [33]	100	8.92×10^{-4}	8.19×10^{-1}	6
	PN++ [34]	99.5	1.22×10^{-2}	8.90	6
	DGCNN [50]	100	2.13×10^{-3}	1.48	3
	PointMLP [27]	99.8	3.77×10^{-3}	9.83×10^{-1}	13

Table 3. Comparison of attack results on PN(PointNet), PN++(PointNet++), DGCNN and PointMLP.

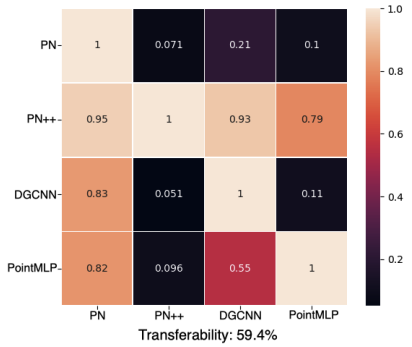

Figure 1. Transferability for PointNet, PointNet++, DGCNN and PointMLP. Networks on the rows and columns denote from which victim networks the adversarial examples are generated and to which those examples are transferred respectively. Brighter squares denote higher transferabilities. The total transferabilities under the matrices are the averages of the off-diagonal values of corresponding methods.

ent scales, resulting in more points involved in the aggregation. In addition, the incorporation of random sampling enhances robustness. See Sec. 7.6 for detailed analyses.

Targeted attack: We also extend the proposed methods to targeted attacks. To alleviate redundant experiment procedures, we employ three alternatives of conducting ergodic targeted attack: *random*, *lowest* and *second-largest* activation attack. In the random activation attack we choose one stochastic target from the 39 labels (excluding the ground-truth one) as the optimization destination. In the lowest and second-largest activation attack, we decrease the activation of ground truth while boosting the lowest or second-largest activation respectively until it becomes the largest one in the logits. The results, as shown in Tab. 4, indicate that though the performance of OPA is deteriorated when converting to targeted attacks due to the rigid restriction on the perturbation dimension, CTA survived even the worst case (the lowest activation attack) with a remarkably high success rate and a minuscule number of perturbation points. We also demonstrate the results from LG-GAN [59], which also dedicates to targeted attack for PC networks. In comparison, CTA achieves an approximated success rate with a much smaller D_c. Note that their approach is based on generative models and the comparison is for reference only.

4.2. Adversarial examples visualization

Fig. 2 visualizes two adversarial examples for OPA and CTA respectively. Interestingly, in CTA, regardless of the absence of the restriction on the perturbation dimension, there are instances (e.g. the car in CTA) where only one-point shifting is required for an adversarial example. More

	Pattern	S(↑)	$D_c(\downarrow)$	$D_h(\downarrow)$	$N_p(\downarrow)$
O	Second-largest	58.5	9.49×10^{-4}	9.33×10^{-1}	1
P	Random	20.9	1.06×10^{-2}	1.08×10^{1}	1
A	Lowest	6.3	4.73×10^{-3}	4.80	1
C	Second-largest	99.5	1.55×10^{-3}	8.14×10^{-1}	5
T	Random	97.7	5.75×10^{-3}	2.31	10
A	Lowest	99.0	8.52×10^{-3}	3.06	13
LG-GAN [59]		98.3	3.80×10^{-2}	-	-

Table 4. Targeted OPA and CTA on PointNet. Targeting all labels for each instance in the test set is time-consuming. Therefore, we generalize it with three substitutes: random, the second-largest and the lowest activation in the logits. We also show the results of LG-GAN as a reference.

qualitative visualizations are presented in Fig. S1 and S2.

5. Discussion

In this section, we present our viewpoint concerning the robustness of PC networks (5.1) and discuss the potential of OPA from the viewpoint of explainability (5.2).

5.1. Structural stability of PC networks

Plenty of researches have discussed defense strategies against intentional attacks for PC networks [56, 24, 60, 25, 19, 44, 59, 55], the majority of which were with respect to embedded defense modules, such as outlier removal. However, there has been little discussion about the stability of the intrinsic architectures. Inspired by [44] who investigated the impacts of different pooling layers on the robustness, we replace the max-pooling in PointNet with multifarious pooling layers. As Tab. 5 shows, although PointNet with average and sum-pooling sacrifice 3.3% and 10.4% accuracies in the classification task, the success rates of OPA on them plummet from 98.7% to 44.8% and 16.7% respectively, and the requested perturbation magnitudes are dramatically increased, which stands for enhanced stabilization. We speculate that it depends on how many points from the input instances the model employs as bases for predictions. We calculate the normalized IG contributions of all points from the instances correctly predicted among the 2468 test instances, and we also introduce the Gini coefficient [12] to quantify the dispersion of the absolute attributions which is formulated as:

$$G = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} ||a_i| - |a_j||}{2n^2 \, |\bar{a}|} \tag{6}$$

where a is the attribution mask generated by IG. We demonstrate the corresponding results in Tab. 5, 6 and Fig. S11. There are significant distributional distinctions between the max, average and sum-pooling architectures. PointNet with average and sum-poolings adopt 70.18% (718.5 points) and 84.78% (868.2 points) of the points to positively sustain the corresponding predictions, where the percentages of points attributed to the top 20% are 0.65% (6.7 points) and 1.16% (11.9 points), respectively, while these proportions are only 38.79% (397.2 points) and 0.15% (1.5 points) in the max-pooling structured PointNet. Moreover, the Gini coefficients reveal that in comparison to the more even distribution of attributions in average (0.53) and sum-pooling (0.49), the dominant majority of attributions in PointNet with max-pooling are concentrated in a minuscule number of points (0.91). Hence, it could conceivably be hypothesized that for PC networks, involving and apportioning the attribution across more points in prediction would somewhat alleviate the impact of corruption at individual points on decision outcomes, and thus facilitate the robustness of the networks. Surprisingly, median-pooling appears to be an exception. While the success rate of OPA is as low as 0.9%, the generated adversarial examples only require perturbing $D_h = 9.55 \times 10^{-2}$ in average (all experiments sharing the same parameters, i.e. without any distance penalty attached). On the other hand, despite that merely 53.53% (548.1) points are positively attributed to the corresponding predictions, with only 0.23% (2.4 points) of them belonging to the top 20%, which is significantly lower than the average and sum-pooling architectures, median-pooling is almost completely immune to the deception of OPA. We believe that median-pooling is insensitive to extreme values, therefore the stability to perturbations of a single point is dramatically reinforced.

Another interesting observation about the attribution distribution is based on Activation Maximization (AM), which we report in section 7.9.

5.2. Towards explainable PC models

Despite the massive number of adversarial methods that contribute to the model robustness for computer vision tasks, to our best knowledge, none has discussed the explainability of PC networks. However, we believe that the adversarial methods can facilitate the explainability of the models to some extent. Recall the roles of counterfactuals in investigating the explainability of models processing tabular data [7]. Counterfactuals provide explanations for chosen decisions by describing what changes on the input would lead to an alternative prediction while minimizing the magnitude of the changes to preserve the fidelity, which is identical to the process of generating adversarial examples [10]. Unfortunately, owing to the multidimensional geometric information that is unacceptable to the human brain, existing image-oriented approaches addressed the counterfactual explanations only at the semantic level [15, 49].

Several studies have documented that a better counterfactual needs to be sparse because of the limitations on human category learning [18] and working memory [28, 3]. In addition, previous adversarial studies on images have also suggested that unidimensional perturbations contribute to depicting relatively perceptible vicinities and boundaries

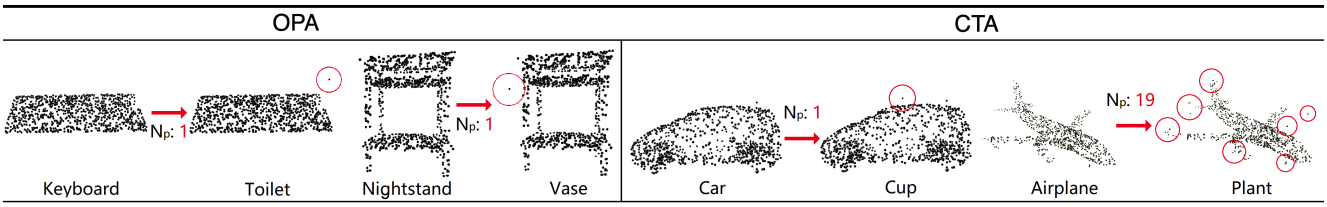

Figure 2. Adversarial examples for OPA and CTA. N_p denotes how many points are shifted.

	Acc.	S↑)	$D_c(\downarrow)$	$D_h(\downarrow)$	N_{pos}	Gini.
Max-pooling	87.1	98.7	8.45×10^{-4}	8.64×10^{-1}	397.2	0.91
Average-pooling	83.8	44.8	2.94×10^{-3}	2.96	718.5	0.53
Median-pooling	74.5	0.9	1.28×10^{-4}	9.55×10^{-2}	548.1	0.57
Sum-pooling	76.7	16.7	2.50×10^{-3}	2.53	868.2	0.49

Table 5. Model accuracies, success attacking rates, average Chamfer and Hausdorff distances of OPA on PointNet with max, average, median and sum-pooling on the last layer respectively. The evaluation accuracy is also presented in the second column. N_{pos} denotes how many points are positively attributed to the prediction, and Gini. denotes the Gini coefficient of the corresponding attribution distributions.

	Top 20%	Top 40%	Positive
Max-pooling	0.15%	0.23%	38.79%
Average-pooling	0.65%	2.12%	70.18%
Median-pooling	0.23%	0.59%	53.53%
Sum-pooling	1.16%	4.53%	84.78%

Table 6. Overview of the percentage of top-20%, top-40% and positive attributed points with four different pooling layers. Complete pie diagrams are shown in Fig. S11.

[43]. Fig. 3 compares the visualization of multidimensional and unidimensional perturbations. The latter, though larger in magnitude, shows more clearly the perturbation process of the prediction from "car" to "radio", and makes it easier to perceive the decision boundary. Conversely, while higher dimensional perturbations perform better on imperceptibility for humans, they are more difficult for understanding the working principles of the model.

In addition, we found another application where the proposed method facilitate the explainability. Evaluating explanations is a major challenge for explainability studies due to the lack of ground truth [6]. An intuitive idea is sensitivity testing, i.e., perturbing features in the explanation that possess high attributions and observing whether the prediction results dramatically change. Theoretically, in our methods, a more accurate explanation induces a more precise selection of critical points, and therefore a higher success rate when perturbing them for generating adversarial examples. Tab. 7 presents the attack performances utilizing gradient-based explainability methods: Vanilla Gradients [39], Guided Back-propagation [42] and IG as the critical identifier respectively. Our results are consistent with [16] and [48], the performance of IG is comparatively better than that of Vanilla Gradients and Back-propagation.

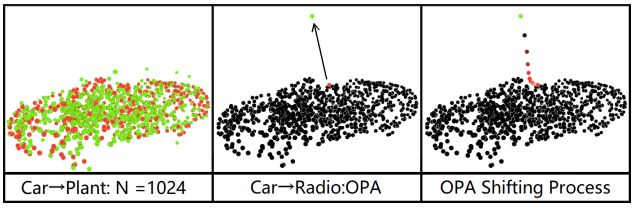

Figure 3. Intuitive visualization of multidimensional shifting(left), unidimensional OPA shifting(middle) and the shifting process of OPA(right). In the right plot, the redder the point the higher the confidence for label "car". In the right plot the green point indicates that the prediction is altered.

Mtd.	S(↑)	$D_c(\downarrow)$	$D_h(\downarrow)$	$N_p(\downarrow)$
VG	82.5	8.20×10^{-4}	8.01×10^{-1}	1
GB	83.4	8.21×10^{-4}	8.02×10^{-1}	1
IG	**98.7**	8.64×10^{-4}	8.45×10^{-1}	1

Table 7. OPA performances utilizing various gradient-based explainability methods to identify the critical points, where VG, GB and IG denote Vanilla Gradients [39], Guided Back-propagation [42] and Integrated Gradients respectively.

6. Conclusion

As the first attack methods for PC networks incorporating explainability, we demonstrate the significance of individual critical points for PC network predictions. For future work, filtering out those critical points in advance is a promising direction to improve explainability. Besides, we are looking forward to higher-quality and human-understandable explanations for PC networks.

Acknowledgements. This research has been funded by the Federal Ministry of Education and Research of Germany and the state of North-Rhine Westphalia as part of the Lamarr-Institute for Machine Learning and Artificial Intelligence, LAMARR22B.

References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *arXiv preprint arXiv:1810.03292*, 2018.

[2] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *Ieee Access*, 6:14410–14430, 2018.

[3] George A Alvarez and Patrick Cavanagh. The capacity of visual short-term memory is set both by visual information load and by number of objects. *Psychological science*, 15(2):106–111, 2004.

[4] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[5] Kieran Browne and Ben Swift. Semantics and explanation: why counterfactual explanations produce adversarial examples in deep neural networks. *arXiv preprint arXiv:2012.10076*, 2020.

[6] Nadia Burkart and Marco F Huber. A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, 70:245–317, 2021.

[7] Ruth MJ Byrne. Counterfactuals in explainable artificial intelligence (xai): Evidence from human reasoning. In *IJCAI*, pages 6276–6282, 2019.

[8] 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.

[9] 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.

[10] Susanne Dandl, Christoph Molnar, Martin Binder, and Bernd Bischl. Multi-objective counterfactual explanations. In *International Conference on Parallel Problem Solving from Nature*, pages 448–469. Springer, 2020.

[11] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018.

[12] Robert Dorfman. A formula for the gini coefficient. *The review of economics and statistics*, pages 146–149, 1979.

[13] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

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

[15] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *International Conference on Machine Learning*, pages 2376–2384. PMLR, 2019.

[16] Ananya Gupta, Simon Watson, and Hujun Yin. 3d point cloud feature explanations using gradient-based methods. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.

[17] Abdullah Hamdi, Sara Rojas, Ali Thabet, and Bernard Ghanem. Advpc: Transferable adversarial perturbations on 3d point clouds. In *European Conference on Computer Vision*, pages 241–257. Springer, 2020.

[18] Mark T Keane and Barry Smyth. Good counterfactuals and where to find them: A case-based technique for generating counterfactuals for explainable ai (xai). In *International Conference on Case-Based Reasoning*, pages 163–178. Springer, 2020.

[19] Jaeyeon Kim, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Minimal adversarial examples for deep learning on 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7797–7806, 2021.

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[21] Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world, 2016.

[22] Kibok Lee, Zhuoyuan Chen, Xinchen Yan, Raquel Urtasun, and Ersin Yumer. Shapeadv: Generating shape-aware adversarial 3d point clouds. *arXiv preprint arXiv:2005.11626*, 2020.

[23] Xinke Li, Zhirui Chen, Yue Zhao, Zekun Tong, Yabang Zhao, Andrew Lim, and Joey Tianyi Zhou. Pointba: Towards backdoor attacks in 3d point cloud, 2021.

[24] 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.

[25] Daniel Liu, Ronald Yu, and Hao Su. Adversarial shape perturbations on 3d point clouds. In *European Conference on Computer Vision*, pages 88–104. Springer, 2020.

[26] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[27] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual mlp framework. *arXiv preprint arXiv:2202.07123*, 2022.

[28] George A Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.

[29] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[30] Anh Nguyen, Jason Yosinski, and Jeff Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *arXiv preprint arXiv:1602.03616*, 2016.

[31] Anh Nguyen, Jason Yosinski, and Jeff Clune. Understanding neural networks via feature visualization: A survey. In *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 55–76. Springer, 2019.

[32] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017.

[33] 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.

[34] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017.

[35] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[36] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[37] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR, 2017.

[38] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[39] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps, 2014. arXiv preprint, arXiv:1312.6034.

[40] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[41] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[42] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net, 2015. arXiv preprint, arXiv:1412.6806.

[43] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

[44] 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.

[45] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Gradients of counterfactuals. *arXiv preprint arXiv:1611.02639*, 2016.

[46] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.

[47] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[48] Hanxiao Tan and Helena Kotthaus. Surrogate model-based explainability methods for point cloud nns. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2239–2248, 2022.

[49] Tom Vermeire and David Martens. Explainable image classification with evidence counterfactual. *arXiv preprint arXiv:2004.07511*, 2020.

[50] 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)*, 38(5):1–12, 2019.

[51] Yuxin Wen, Jiehong Lin, Ke Chen, C. L. Philip Chen, and Kui Jia. Geometry-aware generation of adversarial point clouds, 2020.

[52] Matthew Wicker and Marta Kwiatkowska. Robustness of 3d deep learning in an adversarial setting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11767–11775, 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] Chong Xiang, Charles R Qi, and Bo Li. Generating 3d adversarial point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9136–9144, 2019.

[55] Jinlai Zhang, Lyujie Chen, Binbin Liu, Bo Ouyang, Qizhi Xie, Jihong Zhu, and Yanmei Meng. 3d adversarial attacks beyond point cloud. *arXiv preprint arXiv:2104.12146*, 2021.

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

[57] Yue Zhao, Yuwei Wu, Caihua Chen, and Andrew Lim. On isometry robustness of deep 3d point cloud models under adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1201–1210, 2020.

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

[59] 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.

[60] 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/CVF International Conference on Computer Vision*, pages 1961–1970, 2019.