TOWARDS EFFECTIVE ADVERSARIAL ATTACK AGAINST 3D POINT CLOUD CLASSIFICATION

Chengcheng Ma[1,4], *Weiliang Meng*[2,3,1,4], *Baoyuan Wu*[5,6], *Shibiao Xu*[1,4,α], *Xiaopeng Zhang*[1,2,4,β]

[1]NLPR, Institute of Automation, Chinese Academy of Sciences,[2]Zhejiang Lab,
[3]The State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences,
[4] School of Artificial Intelligence, University of Chinese Academy of Sciences
[5]School of Data Science, The Chinese University of Hong Kong, Shenzhen, China
[6]Shenzhen Research Institute of Big Data

2021 IEEE International Conference on Multimedia and Expo (ICME) | 978-1-6654-3864-3/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICME51207.2021.9428207

ABSTRACT

In the domain of 3D point cloud classification, deep learning based classifiers have made significant progress, while they have been also proven to be vulnerable on the adversarial attack at the same time. Some recent works employ the attack methods that devised for image classification such as projected gradient descent (PGD) to attack the 3D classifiers, but their performances seem quite limited when faced with statistical operations including point cloud denoising and point cloud upsampling. In this paper, we propose 'SmoothAttack', a new attack that can craft adversarial point clouds robust to statistical operations. SmoothAttack can be easily applied in both global constraint and pointwise constraint. Besides, we analyze the directions of perturbations onto the point cloud during the iteration process, where SmoothAttack can somehow stabilize the direction and make full use of the adversarial budgets. Experiments validate that our 'SmoothAttack' can raise the attack success rates against statistical defenses up to 98% for untargeted attack and 91% for targeted attack on ModelNet40 database when fooling the classifiers PointNet and DGCNN.

***Index Terms*—** Adversarial example, point cloud classification, SOR defense, denoiser and upsampler network defense, deep learning

1. INTRODUCTION

The deep neural networks (DNNs) based classifiers towards 3D point data have been rapidly developed in the last few years. Different sorts of classifiers including [1] and [2] can achieve a classification accuracy around 90% on the popular ModelNet40 database [3]. With the rapid development in the domain of adversarial attack towards 2D image classification task, several recent attackers(*e.g.*, [4, 5, 6, 7, 8]) employ the traditional attack algorithms to the 3D domain by carefully crafting adversarial point clouds that cannot be visually discriminated, and bring the classification accuracy down to nearly 0%, meaning that 3D point cloud classifiers are very vulnerable to adversarial examples. As a category of existing 3D adversarial attacks, point shifting focuses on how to modify the positions of the component points within a small range (named *adversarial budget*) and maximize the classification loss. For example, Liu *et al.* [5] and Yang *et al.* [8] employ the well-known projected gradient descent (PGD) attack ([9]) coming from the image adversarial attack, which greedily moves the positions of all points in the direction of gradient with a α-stepsize during each iteration.

By applying the statistical outlier removal (SOR) and point cloud upsampling of the PU-Net ([10]), Zhou *et al.* propose to restore 3D adversarial objects to benign objects. In fact, SOR is a simple defense strategy that removes outlier

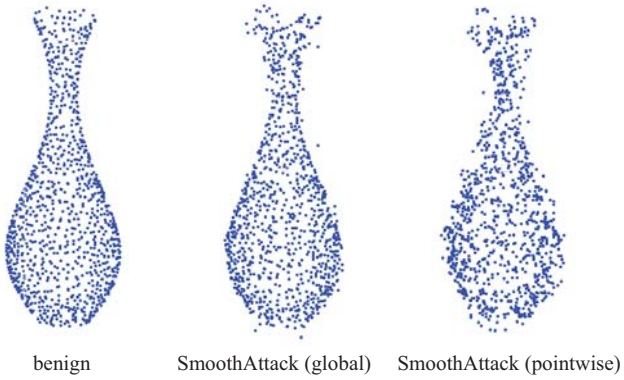


Fig. 1: Visualization of adversarial point clouds crafted by our proposed SmoothAttack with the global constraint and the pointwise constraint.

[α]Corresponding Author: shibiao.xu@nlpr.ia.ac.cn
[β]Corresponding Author: xiaopeng.zhang@ia.ac.cn

This work is being supported in part by the National Key R&D Program of China (No. 2018YFB2100602), in part by the National Natural Science Foundation of China (Nos. 61620106003, U2003109, 61971418, 61761003, and 61972459), and by Open Research Projects of Zhejiang Lab(No. 2021KE0AB07).

978-1-6654-3864-3/21/$31.00 ©2021 IEEE

points within the point cloud. The SOR method computes the average distance from the k-nearest neighbors for each component point firstly, and those points whose average distance exceed the threshold $\mu + c \cdot \sigma$ are considered as outlier points and discarded, where μ and σ are the mean and standard deviation of the average distances. Here k and c are hyperparameters, which are default set as 10 and 1.0, respectively. The SOR defense can effectively defend various of existing attacks reported in [11], and can raise the classification accuracy on adversarial examples crafted by [4] from 0% to 81%. Besides, the point cloud upsampling operation of PU-Net can further improve the performance by 4%. According to our experimental results, SOR can suppress the attack success rate of PGD method from 100% to 41% and 60% with global constraint and pointwise constraint respectively on database ModelNet40 ([3]) with the classifier PointNet([1]).

In this paper, we introduce a new attack named 'SmoothAttack' to overcome the statistical defense in [11]. We observe that the gradient information from neighbourhoods of the current data will enhance the robustness against simple statistical operations, because the directions of average of gradients tend to be more consistent during iteration process, which avoids dropping into poor local maxima and being easily defended. We view this average of gradients as the smooth operation, and design our 'SmoothAttack' by generating the adversarial point clouds based on updating the point position on the 'smoothed' gradient.

The main contributions of our work are:

- We propose a new attack method named 'SmoothAttack', which can craft adversarial point clouds that robust to statistical operations, and can be easily applied in both global constraint and pointwise constraint. Our 'SmoothAttack' can stabilize the direction and make full use of the adversarial budget by smoothing the gradients of several sample times.

- We implement comprehensive evaluations on our 'SmoothAttack', validating the effectiveness on the ModelNet40 database with classifiers PointNet and DGCNN, in which the attack success rates up to 98% for untargeted attack and 91% for targeted attack.

2. RELATED WORK

The point cloud adversarial attack is a new and developing research field, where the attack methods can be divided into three categories: point shifting, point adding, and point dropping.

As for the point shifting attack, it is quite similar with the adversarial perturbation on 2D image pixels, which means the number of points in the point cloud keeping unchanged. Naturally, the attacking algorithms on image classifiers can be freely adapted onto the field of point cloud classification. The general idea of point shifting attack is either to search the minimum magnitude of perturbation or to iteratively perturb the point data with a fixed magnitude. For instance, Xiang et al. [4] is the first to craft adversarial point clouds by utilizing the optimization framework of CW-ℓ_2, where they apply binary search to find the most imperceptible shifting of points. Subsequently, Wen et al. [12] add more terms related with local curvatures information to the loss function of [4], in order to generate adversarial point clouds with more natural geometry properties. On the other hand, Liu et al. [5] [13] add perturbations onto the given point cloud iteratively, same as the image adversarial attack [9]. Furthermore, they propose two heuristic resampling methods (*i.e.*farthest point sampling (FPS) and radial basis function sampling (RBF)) to refine the perturbed point cloud after each iteration, trying to keep the density of points uniform all the time. Hamdi et al. [6] also use the iteration equation in [9], but expanded with a novel term. They need to train a deep neural network with an encoder-decoder architecture model to do point cloud denoising at first, then attack both the original point cloud and the denoised point cloud all together during iterations.

However, none of the above methods even tries to analyze the defense mechanism of DUP-Net [11], not to mention the way to evading the defense. We have studied the success rate of dodging DUP-Net defense in later experiments for those methods with released codes ([4], [5], [13]), and have referred to their own reports about the success rate for those without released codes ([12], [6]). The success rates of [4], [12] and [6]) are less than 40%, which is quite limited. Although the success rate of [5] and [13] can be up to 86%, but this is accompanied by a extreme distortion on point cloud shape.

The point adding attack has only been practiced by Xiang et al. [4], including adding points, clusters, or tiny objects near the critical points for the classifier. The point dropping attack such as [14] and [15] iteratively drop points with strategies. However, Zhou et al. [11] report that the success rates of the two attacks can be suppressed down to about 8% and 52%. Since both the adding methods and the dropping methods change the number of point cloud, we will not discuss them in this paper.

3. PROPOSED METHOD

3.1. Notations

The point cloud classification model is denoted as $f : \mathcal{X} \to \mathcal{Y}$, with $\mathcal{X} \in \mathbb{R}^{N\times 3}$ being the space of the point cloud, and $\mathcal{Y} = \{1, 2, \ldots, K\}$ being the output classification space. The pair of the benign point cloud and the ground-truth label is denoted as $(\mathbf{X}, y)$, where $\mathbf{X} = [\boldsymbol{x}_1; \ldots; \boldsymbol{x}_N] \in \mathbb{R}^{N\times 3}$ with $\boldsymbol{x}_i \in \mathbb{R}^{1\times 3}$ being the i-th point. Our goal is to fool f by crafting an adversarial point cloud $\mathbf{X}^*$ which is similar with its corresponding $\mathbf{X}$ in appearance. Besides, let $f(\cdot)$ denote the probability vector predicted by the classifier, $L(f(\mathbf{X}), y)$

be the classifier loss function taking a point cloud $\mathbf{X}$ and a label y as inputs, while $D(\mathbf{X}, \mathbf{X}^*)$ denotes a distance metric between a benign point cloud and its adversarial counterpart, and $S(\mathbf{X})$ denotes the point cloud processed by the SOR defense.

3.2. Our SmoothAttack

The performances of 3D classifiers are deteriorated when faced with statistical operations such as the statistical outlier removal(SOR) in the DUP-Net defense proposed by Zhou et al. [11].

The SOR algorithm computes the average of the k-nearest neighboring distances for each point $\boldsymbol{x}_i$, formulated as

$$d_i = \frac{1}{k} \sum_{\boldsymbol{x}_j \in knn(\boldsymbol{x}_i)} \|\boldsymbol{x}_j - \boldsymbol{x}_i\|_2 . \tag{1}$$

Here, $knn(\boldsymbol{x}_i)$ denotes the set of the k-nearest points around $\boldsymbol{x}_i$. Then the mean and standard deviation of all d_i ($i \in \{1, \cdots, N\}$) can be calculated and represented as μ_d and σ_d. Finally, the points contained in the set $\{d_i | d_i > \mu_d + t_d \cdot \sigma_d\}$ are removed from the point cloud, so the point number of denoised point cloud will be less than the original point cloud. Intuitively, those outliers, which are relatively far away from their neighboring points, are more likely to be abandoned. In terms of the point shifting attack, the attacker tends to make just a few points off the surface of the object, and such outliers contribute most to causing the wrong classification [11]. After this denoising operation, the denoised point cloud may be non-malicious to classifiers. The DUP-Net defense is by far the most efficient defense against the adversarial attacks to point cloud classification. Both the denoising and the upsampling steps endeavor to pull the adversarial point clouds back to the natural manifold of benign ones.

Our 'SmoothAttack' can overcome the statistical defense methods by gather the information from neighbourhoods of the current data during the attack process, and the whole process is equivalent of attacking an ensemble of classifiers termed as *smooth classifier*. Similar with [16] and [17], the classification are formulated as

$$g(\mathbf{X}) = \frac{1}{T} \sum_{i=1}^{T} f_i(\mathbf{X}) = \frac{1}{T} \sum_{i=1}^{T} f(\mathbf{X} + \delta_i), \tag{2}$$

where δ_i denotes a random noise following a uniform distribution $\delta \sim \mathcal{U}(r)$, and T denotes the sample times. Furthermore, the objective function of untargeted attack against the smooth classifier can be written as

$$\max L(g(\mathbf{X}^*), y), \quad \text{subject to } D(\mathbf{X}, \mathbf{X}^*) \leqslant \epsilon. \tag{3}$$

When employing the targeted attack, y should be the target class t, and the $\max$ operator should be replaced with the $\min$ operator. We choose the projected gradient descent method to optimize the objective function. For untargeted attack, the update process is specified as

$$\mathbf{X}^* \leftarrow \mathbf{X}^* + \alpha \cdot \frac{\nabla_{\mathbf{X}^*} L(g(\mathbf{X}^*), y)}{\|\nabla_{\mathbf{X}^*} L(g(\mathbf{X}^*), y)\|} \tag{4}$$

$$= \mathbf{X}^* + \alpha \cdot \frac{\sum_{i=1}^{T} \nabla_{\mathbf{X}^*} L(f_i(\mathbf{X}^*), y)}{\left\| \sum_{i=1}^{T} \nabla_{\mathbf{X}^*} L(f_i(\mathbf{X}^*), y) \right\|}. \tag{5}$$

Note that the $\cdot / \|\cdot\|$ denotes a method of normalization according to the type of norm $\|\cdot\|$. If considering the ℓ_2 norm between the entire clean point cloud and the entire adversarial point cloud, we can view the attack method as the global PGD ([9]), which utilizes the global constraint. If considering the ℓ_2 norm for each component point, we can view it as the pointwise PGD ([8]), which utilize the pointwise constraint.

Our SmoothAttack is summarized in Algorithm 1. In order to generate the adversarial point cloud $\mathbf{X}^*$, we use the original point cloud $\mathbf{X}$ to initial $\mathbf{X}^*$, then repeat to update $\mathbf{X}^*$ in n steps based on Eqa 5. After this computation, the adversarial point cloud $\mathbf{X}^*$ can be obtained that can fool the classifiers effectively even after different defense methods like SOR or DUP.

Algorithm 1 SmoothAttack Algorithm.

Input: A point cloud classifier f with loss function L; benign point cloud $\mathbf{X}$; true label (untargeted) or target label (targeted) t; number of iterations n; point-wise perturbation constraint ϵ; step size α.
Output: Adversarial point cloud $\mathbf{X}^*$

```
Initialize X* ← X
If targeted attack: L ← −L
for i = 1 to n do
    Initialize Δ ← 0
    for i = 1 to T do
        Compute and accumulate the gradient Δ ← Δ + ∇_{X*} L(f_i(X*), y)
    end for
    Update the point cloud as X* ← X* + α · Δ/‖Δ‖.
end for
return Adversarial point cloud X*
```

4. EXPERIMENTAL RESULTS

4.1. Setup

Our experiments are implemented on the public database ModelNet40 ([3]), which contains 9843 samples for training and 2468 samples for testing. ModelNet40 contains 40 different classes, and all the point clouds have 2048 points with xyz-coordinates. For a fair comparison, we make a downsampled version of ModelNet40 by randomly sampling 1024 points from 2048 points for each object in both the training and testing set, which are same with all the existing works. We evaluate the attacks on two classifiers PointNet [1] and

Table 1: The attack success rates (%) on untargeted attack against different settings, namely No Defense (ND), SOR defense (SOR), and DUP-Net defense (DUP). Best results in each column are highlighted in bold, and second best results are in blue.

Attacks	PointNet			DGCNN		
	ND	SOR	DUP	ND	SOR	DUP
CW [18]	99.9	75.6	41.8	99.7	95.5	91.5
PGD (global) [9]	**100.0**	58.7	41.6	99.9	70.7	41.8
PGD (pointwise) [5]	**100.0**	75.1	59.1	**100.0**	90.7	72.3
Our SmoothAttack (global)	**100.0**	77.7	68.2	**100.0**	99.0	90.5
Our SmoothAttack (pointwise)	**100.0**	**97.7**	**95.9**	**100.0**	**99.5**	**93.7**

DGCNN [2], both of which are pretrained with the downsampled version of ModelNet40 training set and achieve 86.8% and 92.0% accuracies on the testing set.

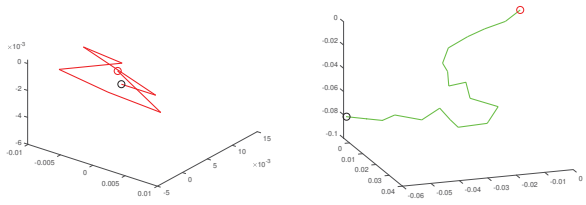
4.2. Attack Evaluation

We compare the attack success rates of our 'SmoothAttack' with the PGD method under global constraint and pointwise constraint, as well as the CW attack[1] proposed in [18]. We use $\epsilon = 3.0$ and $\alpha = 0.1$ under the global constraint, and use $\epsilon = 0.1$ and $\alpha = 0.01$ under the pointwise constraint, with iteration step being 40. Note that only the examples which can be correctly predicted are chosen for the attack evaluation for each classifier, and the attack success rate is a ratio between the number of successful adversarial examples and the number of correctly classified examples.

4.2.1. Untargeted Attack

Table.1 shows the untargeted attack performance on both PointNet and DGCNN for different settings, namely no defense, SOR defense, and DUP-Net defense. In general, our SmoothAttack examples can successfully fool the classifiers with a 100% success rate when there is no defense. Aiming at the SOR defense, the success rates of PGD attack decrease at different level. Our 'SmoothAttack' can improve the success rate by 19.0% for PointNet and 28.3% for DGCNN under the global constraint, and improve by 22.6% for PointNet and 8.8% for DGCNN under the pointwise constraint. Additionally, the attack ability of CW seems limited, which shows a similar performance with PGD (pointwise). We find that the attacks under the pointwise constraint shows stronger ability, which have been also reported in [5] and [7]. The reason of this fact is that the global constraint tolerates the perturbation vectors containing large values on some certain elements, making the points in corresponding positions moved off the object surface too far, which are further considered as the outlier points and discarded by the SOR algorithm, as these outlier points contribute most to the adversarial effect. In comparison, the pointwise constraint avoids this situation and still maintains the object surface after point shifting, which enhances the attack ability to overcome the SOR defense.

[1] https://github.com/jinyier/ai_pointnet_attack



(a) PGD [5] (b) SmoothAttack

Fig. 2: The update trajectories of a certain point within the point cloud for PGD (*red*) and our SmoothAttack (*green*) during the iteration process when fooling PointNet. The update trajectory of our SmoothAttack shows more continuously trend to the extreme value.

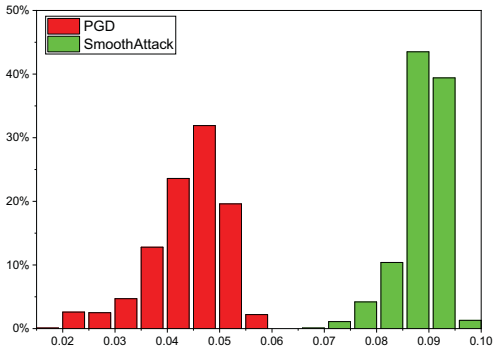


Fig. 3: The histograms of perturbation size for PGD examples (*red*) and our SmoothAttack (*green*) examples under pointwise constraint, with adversarial budget being 0.1. The x-axis indicates the perturbation size, and the y-axis shows the proportion by observing 1000 adversarial examples. We can see that our SmoothAttack can use the adversarial budget more effectively.

4.2.2. Targeted Attack

Both the best case and the average case are considered in our experiment, whose definition are detailed as

- *Best Case:* select the "runner-up" class as the target.
- *Average Case:* select the target class randomly among K classes excluding the most probable one.

Table.2 shows the targeted attack performance for no defense, SOR defense, and DUP defense. OUr 'SmoothAttack' outperforms PGD and CW when fooling PointNet, and can achieve a relative lower rate with CW when fooling DGCNN, as DGCNN performs less robust than PointNet. In

Table 2: The success rates (%) on the targeted attack (best case and average case) against different settings, namely No Defense (ND), SOR defense (SOR), and DUP-Net defense (DUP). Best results in each column are highlighted in bold, and second best results are in blue.

Attacks	PointNet						DGCNN					
	best case			average case			best case			average case		
	ND	SOR	DUP	ND	SOR	DUP	ND	SOR	DUP	ND	SOR	DUP
CW [18]	99.9	67.8	36.6	**97.1**	21.3	6.7	**100.0**	**77.6**	13.7	**99.7**	**58.4**	7.2
PGD (global) [9]	**100.0**	43.2	25.0	90.2	6.5	2.9	98.9	51.1	17.3	90.8	15.3	2.7
PGD (pointwise) [5]	99.7	59.0	38.8	79.1	14.8	6.6	96.1	55.2	21.6	80.5	24.2	4.9
Our SmoothAttack (global)	99.9	66.3	52.9	93.1	17.8	12.2	97.0	77.3	**43.9**	84.1	45.6	**15.7**
Our SmoothAttack (pointwise)	99.5	**90.6**	**86.2**	83.6	**53.3**	**46.6**	93.8	74.9	32.0	79.8	47.5	15.0

fact, we evaluate the classification accuracy on the testing set perturbed by non-malicous Gaussian noise ($\epsilon = 0.1$), and PointNet misclassifies 10.7% examples and DGCNN misclassifies 56.8% examples. Besides, the performance of our 'SmoothAttack' still surpass that of PGD under global constraint and pointwise constraint.

4.3. Update Directions

To further explain why our 'SmoothAttack' performs better against statistical defenses, we examine the update trajectory of a certain point within the point cloud during employing the PGD attack and our 'SmoothAttack' respectively as shown in Figure.2. We find that by attacking a smooth classifier instead of the original classifier, our 'SmoothAttack' can stabilize the update directions during the iteration process.

The PGD attack greedily moves the positions of all component points in the gradient direction with a specified stepsize, while it is lack of guided information to avoid the adversarial examples getting stuck in poor local maxima during each iteration. As a result, we find that the size of perturbation computed by the PGD attack preserves limited after 40 iterations, no matter how large the adversarial budget is. Figure.3 demonstrates that our 'SmoothAttack' uses the adversarial budget more effectively than PGD attack when adversarial budget ϵ, stepsize α, and iteration step in the same setting, which further illustrates its stronger attack ability.

5. CONCLUSION

In this paper, we propose a new attack method named 'SmoothAttack' that can craft adversarial point clouds robustly to statistical operations. Our 'SmoothAttack' can stabilize the direction and make full use of the adversarial budget, making the adversarial point clouds generation faster and effectively. Experiments demonstrate that our 'SmoothAttack' can raise the attack success rates up to 95% for the untargeted attack and 88% for the targeted attack on ModelNet40 database against classifiers PointNet and DGCNN.

6. REFERENCES

[1] 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*, 2017, pp. 652–660.

[2] 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)*, vol. 38, no. 5, pp. 1–12, 2019.

[3] 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*, 2015, pp. 1912–1920.

[4] Chong Xiang, Charles R Qi, and Bo Li, "Generating 3d adversarial point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9136–9144.

[5] Daniel Liu, Ronald Yu, and Hao Su, "Extending adversarial attacks and defenses to deep 3d point cloud classifiers," in *Proceedings-International Conference on Image Processing*, 2019.

[6] Abdullah Hamdi, Sara Rojas, Ali Thabet, and Bernard Ghanem, "Advpc: Transferable adversarial perturbations on 3d point clouds," *arXiv preprint arXiv:1912.00461*, 2019.

[7] Chengcheng Ma, Weiliang Meng, Baoyuan Wu, Shibiao Xu, and Xiaopeng Zhang, "Efficient joint gradient based attack against sor defense for 3d point cloud classification," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 1819–1827.

[8] 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.

[9] Alexey Kurakin, Ian Goodfellow, and Samy Bengio,

"Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.

[10] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng, "Pu-net: Point cloud upsampling network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2790–2799.

[11] 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*, 2019, pp. 1961–1970.

[12] Yuxin Wen, Jiehong Lin, Ke Chen, and Kui Jia, "Geometry-aware generation of adversarial and cooperative point clouds," *arXiv preprint arXiv:1912.11171*, 2019.

[13] Daniel Liu, Ronald Yu, and Hao Su, "Adversarial point perturbations on 3d objects," *arXiv preprint arXiv:1908.06062*, 2019.

[14] Matthew Wicker and Marta Kwiatkowska, "Robustness of 3d deep learning in an adversarial setting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11767–11775.

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

[16] Warren He, Bo Li, and Dawn Song, "Decision boundary analysis of adversarial examples," in *International Conference on Learning Representations*, 2018.

[17] Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang, "Provably robust deep learning via adversarially trained smoothed classifiers," in *Advances in Neural Information Processing Systems*, 2019, pp. 11292–11303.

[18] 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*, 2020, vol. 34, pp. 954–962.