Hindawi
Security and Communication Networks
Volume 2022, Article ID 1186633, 11 pages
https://doi.org/10.1155/2022/1186633

Research Article

NormalAttack: Curvature-Aware Shape Deformation along Normals for Imperceptible Point Cloud Attack

Keke Tang,[1] **Yawen Shi**,[1] **Jianpeng Wu**,[1] **Weilong Peng**,[2] **Asad Khan**,[2] **Peican Zhu**,[3] **and Zhaoquan Gu**[1]

[1]*Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China*
[2]*School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou 510006, China*
[3]*School of Artificial Intelligence, Optics, and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, China*

Correspondence should be addressed to Weilong Peng; wlpeng@gzhu.edu.cn and Asad Khan; asad@gzhu.edu.cn

Keke Tang, Yawen Shi, and Jianpeng Wu contributed equally to this work.

Received 11 May 2022; Revised 11 June 2022; Accepted 21 June 2022; Published 12 August 2022

Academic Editor: Keping Yu

Many efforts have been made on developing adversarial attack methods on point clouds. However, without fully considering the geometric property of point clouds, existing methods tend to produce clearly visible outliers. In this paper, we propose a novel NormalAttack framework towards imperceptible adversarial attacks on point clouds. First, we enforce the perturbation to be concentrated along normals to deform the underlying surface of 3D point clouds, such that tiny perturbation can make the shape deformed for better attack performance. Second, we guide the perturbation to be located more on regions with larger curvature, such that better imperceptibility is achieved. Extensive experiments on three representative networks, e.g., PointNet++, DGCNN, and PointConv, validate the effectiveness of NormalAttack and its superiority to state-of-the-art methods.

1. Introduction

With the development and popularity of deep neural networks (DNNs) [1], their performance on 3D point cloud perception has been significantly improved [2–5]. However, DNNs are reported to be vulnerable to adversarial attacks [6], in which case imperceptible modifications on input samples can lead to erroneous predictions of victim models. Therefore, point cloud perception solutions based on DNNs suffer from the hidden security risk of adversarial attacks, hindering their deployment in safety-critical applications, e.g., autonomous driving [7], 3D object recognition [8, 9], and grasp planning of robotics [10].

In the last few years, many efforts have been made on developing adversarial attack algorithms for DNNs in the field of point clouds. By utilizing the unstructured nature of point clouds, many early research performed attack by adding adversarial points, clusters, and objects [11], or dropping a small set of salient points [12]. To learn better from the great success of adversarial attacks on images [13–15], other branches of attempts focus on applying point-wise perturbation to change point coordinates, by extending the popular 2D C&W attack [16] and fast gradient sign method (FGSM) [17] attack methods. However, without intentionally considering the geometric properties of point clouds, the extended methods [11, 18] can hardly adapt well, and tend to produce clearly visible outliers, hindering their imperceptibility to humans.

In the view of geometry, point clouds of 3D objects are 2-manifold surfaces embedded in the 3D space [19]. Therefore, attacking point clouds by perturbing their points in the xyz axes can only introduce a small portion of perturbation to change the geometric properties, but leave a large remaining portion to form noise, hindering the imperceptibility. Besides, different regions of point clouds can withstand perturbations of different sizes, e.g., a large modification can be imperceptible at salient regions, but a tiny modification at flat regions is still conspicuous. Therefore, applying perturbations with a uniform magnitude will lead the attacks to be perceptible easily.

To resolve the above issues, we propose a novel NormalAttack framework, which applies curvature-aware

shape deformation along normals for imperceptible point cloud attack. First, to directly modify the geometric property of the 2-manifold surfaces, we intentionally enforce the applied perturbation on each point to be concentrated along its normal direction, such that the shape is deformed for attack while allowing a tiny shift along the tangent plane, in which way the deformation is averaged by the local shapes for better imperceptibility. Second, considering the fact that regions with larger curvature can tolerate larger modifications, we devise a curvature-aware attack strategy to guide the perturbation to be concentrated more at these areas, and thus the attack is more imperceptible. We validate the effectiveness of our NormalAttack framework by attacking multiple different deep classification models. Extensive experimental results validate that adversarial point clouds generated by our NormalAttack framework are more imperceptible to those generated by state-of-the-art methods. Besides, we also show NormalAttack is undefendable against adversarial defenses and transferable to unseen classification models, as shown in Figure 1.

Overall, our contribution is at least three-fold.

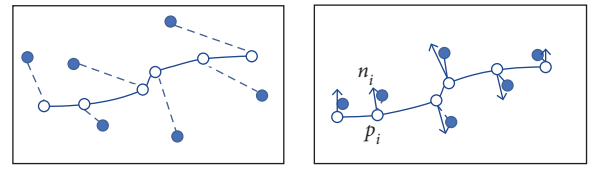
(i) We propose a deformation guiding module that enforces the perturbation to be concentrated along normals to deform the underlying shapes of 3D point clouds for attack.

(ii) We propose a curvature-aware module to guide the perturbation to be concentrated more at regions with larger curvature for imperceptibility.

(iii) We validate the superiority of the NormalAttack framework to the state-of-the-art methods via extensive experiments on PointNet++, DGCNN, and PointConv.

2. Related Work

2.1. Deep Learning for Point Cloud Classification. Deep learning methods have dominated the mainstream solutions for handling point cloud classification. Early attempts first convert irregular point clouds into structured grid representations, e.g., by projecting point clouds into multiview images [20] or rasterizing into 3D voxel grids [21], and then adopting mature 2D convolutional neural networks. However, these methods either suffer from the loss of detailed geometric information or high computation costs. Therefore, since the pioneering PointNet [22] validated that the structure of multilayer perceptrons (MLPs) followed with maximum pooling can overcome the unorder issue of point clouds, recent solutions focus on learning from point clouds directly. To handle the failure of PointNet in recognizing fine-grained patterns, PointNet++ [2] further captures the fine geometric structure of point clouds by hierarchically applying it to the neighborhood of each point. More solutions include convolution-based KPConv [23], PointCNN [24], and graph-based DGCNN [3]. In this paper, we mainly evaluate the adversarial robustness of several representative point cloud classification models.



I-FGSM Ours

AttackType	Imperceptibility	Defense	Transferability
I-FGSM	No	Easy	Low
Ours	Yes	Hard	High

Figure 1: Compared with most other iterative-based methods, e.g., I-FGSM, that perturb points guided by the gradient, our normal attack framework enforces the perturbation concentrated along normals in a curvature-aware manner and is imperceptible, hard to defend, and highly transferable.

2.2. Adversarial Attacks on Deep Learning Models for Point Cloud. Since Szegedy et al. [6] demonstrated the intriguing property of DNNs that an imperceptible perturbation on images can lead them to make mistakes, extensive studies have been made on attacking 2D image classification models [16, 17, 25, 26].

Adversarial attack has been successfully extended to the field of point cloud classification. Due to their unstructured nature, adversarial attack on point clouds can be achieved by adding or deleting points. Xiang et al. [11] performed adversarial attack by adding a limited number of synthetic points, clusters, and objects to the point cloud and showed that PointNet [22] could be fooled in this way. Wicker and Kwiatkowska [27] proposed to determine the critical points in a random and iterative manner and then generated adversarial examples for attack by deleting the critical points. Inspired by the gradient-guided attack method, Yang et al. [28] found key points by calculating the importance scores associated with the labels obtained from the output of the classifier relative to the gradient of the input and then deleted key points in a similar manner. Instead of deleting the points, Zheng et al. [12] devised a more flexible way that moves the points with high saliency towards the center of the shape, such that these points will not influence the surfaces. Another direction of adversarial attack is to perturb point clouds in a similar way as in the field of images. Liu et al. [18] extended the FGSM [17] by adding a l_2-norm constraint to construct imperceptible adversarial 3D point clouds. Lee et al. [29] added adversarial noise to the latent space of an auto-encoder, keeping the decoded shape similar to the original one. To achieve better imperceptibility, Kim et al. [30] proposed to perturb minimal subset of points, instead of all of them. However, very few work exploited the geometric property of point clouds to improve the imperceptibility of generated adversarial point clouds.

2.3. Geometry-Aware Adversarial Attacks. Geometric property is a critical cue for realizing high attack performance and imperceptibility of the point cloud attack task. Tsai et al. [31] incorporated the perturbation constraint into the C&W

framework by introducing a k-nearest neighbor loss to ensure the compactness of the local neighborhoods in the obtained adversarial examples. Wen et al. [32] enforced the consistency of local curvature between the adversarial points and benign ones. Both above studies attempt to apply additional geometric constraints passively to achieve high imperceptibility. However, since these constraints are strict, finding a feasible attack solution while satisfying these geometric constraints is usually very challenging. Considering that point clouds are 2-manifold surfaces embedded in the 3D space, we initiatively guide the perturbation to be concentrated along normals, such that very tiny modification can make the underlying shape deformed, and thus leading to better attack performance and imperceptibility. LG-GAN [33] also exploits the manifold property of point clouds. Differently, they enforce the perturbation to be attached to the manifold, while our NormalAttack attempts to destroy the manifold.

We notice a concurrent work [34] that also moves points along normals. Differently, instead of strictly restrict the moving direction, we allow points to be slightly shifted along the tangent plane, such that more feasible solutions can be searched. Besides, the freedom along tangent plane can make the surfaces after perturbation to be smoother. Last but not least, we adopt a curvature-aware perturbation magnitude to further improve the imperceptibility property.

3. Problem Formulation

3.1. Notations. This work considers the setting in a C-category point cloud classification problem. Let $\mathscr{P}$ be an input point cloud containing a set of unordered points $\mathscr{P} = \{p_i\}_1^n \in \mathbb{R}^{n\times 3}$ sampled from the surface of a 3D object, where each point $p_i \in \mathbb{R}^3$ contains coordinate positions. Let n_i denote the normal of p_i and $F(\cdot)$ denote a classifier, e.g., PointNet++ that predicts the category to which the input point cloud belongs.

3.2. Formulation of Adversarial Attack. Suppose $F(\cdot)$ can originally correctly classify the category of point cloud $\mathscr{P}$

$$y = F(\mathscr{P}),\ y \in \{1, 2, \ldots, C\}, \tag{1}$$

where y denotes the ground truth label of $\mathscr{P}$, adversarial attack aims to find a human-imperceptible perturbation $\triangle$, such that $F(\cdot)$ will make an error prediction on the adversarial point cloud:

$$y' = F(\mathscr{P} + \triangle),\ y' \neq y. \tag{2}$$

Note that the above formulation describes the situation of untargeted attack, while targeted attack can be achieved by additionally designating the expected category to be predicted. If not specifically mentioned, we only consider untargeted attack in this paper.

3.3. Traditional Solution for $\triangle$. By borrowing the experience from adversarial attack in the image field, a widely adopted solution is to apply perturbation in the direction which is guided by the gradient, and with the same magnitude

$$\triangle_{xyz} \sim \epsilon \cdot \operatorname{sign}\left(\nabla_{\mathscr{P}} J(F, \mathscr{P})\right), \tag{3}$$

where $J(F, \mathscr{P})$ is the cost for F on the input $\mathscr{P}$, $\operatorname{sign}(\cdot)$ is the direction function, and ϵ is the perturbation step size.

3.4. Weaknesses of traditional solution. Traditional solutions suffer from at least two main drawbacks. First, the inherent property represented by point clouds is 2-manifold surfaces embed in the 3D space, which is a small subset of the entire 3D Euclidean space. Therefore, noise-like perturbation in the Euclidean space cannot affect the underlying surfaces easily and thus requires ϵ to be large, resulting in messy isolated points. Second, different regions of the shapes can withstand different magnitudes of perturbation. Adopting a uniform ϵ for all different points will easily lead regions that have lower tolerances, e.g., flat areas, to be perceived by humans after applying perturbation.

3.5. Our Solution for $\triangle$. To overcome the above drawbacks, we propose to (1) perturb the point clouds mainly along the normal direction of each point, such that the underlying 2-manifold surfaces can be directly modified for better attack performance; and (2) adopt different perturbation magnitude for different points, such that perturbation on regions that have lower tolerance will be suppressed for better imperceptibility. A formal solution is defined as follows:

$$\triangle_n(p_i) \sim \overline{\epsilon}(p_i) \cdot \overline{\operatorname{sign}}\left(\nabla_{\mathscr{P}} J(F, \mathscr{P}), n_i\right), \tag{4}$$

where $\overline{\epsilon}(p_i)$ denotes the perturbation step size for p_i and $\overline{\operatorname{sign}}(\nabla_{\mathscr{P}} J(F, \mathscr{P}), n_i)$ denotes the perturbation direction is guided by the gradient and n_i.

4. NormalAttack

In this section, we introduce the NormalAttack framework that implements our solution for $\triangle$ described in Section 3. We will first present the two main components: the deformation guiding module and the curvature-aware module, and then describe the whole attack framework. Please refer to Figure 2 for demonstration.

4.1. Deformation Guiding Module. To guide the perturbation to be concentrated along normals such that the underlying surfaces are deformed after applying it, we devise a deformation guiding module (DGM).

Instead of strictly restrict normals as the only available moving directions as in ITA [34], DGM applies a much more soft constraint. Suppose $\widehat{\mathscr{P}}$ is the adversarial point cloud generated from $\mathscr{P}$, $\widehat{p}_i$ denotes the corresponding point of p_i in $\widehat{\mathscr{P}}$, and $\overrightarrow{p_i\widehat{p}_i}$ denotes the vector from p_i to $\widehat{p}_i$, the projected perturbation in the tangent direction can be calculated as follows:

$$D_{\text{tangent}}(\mathscr{P}, \widehat{\mathscr{P}}) = \sum_{i=1}^{n} \sqrt{\left|\overrightarrow{p_i\widehat{p}_i}\right|^2 - \left(\frac{\overrightarrow{p_i\widehat{p}_i} \cdot n_i}{|n_i|}\right)^2}. \tag{5}$$

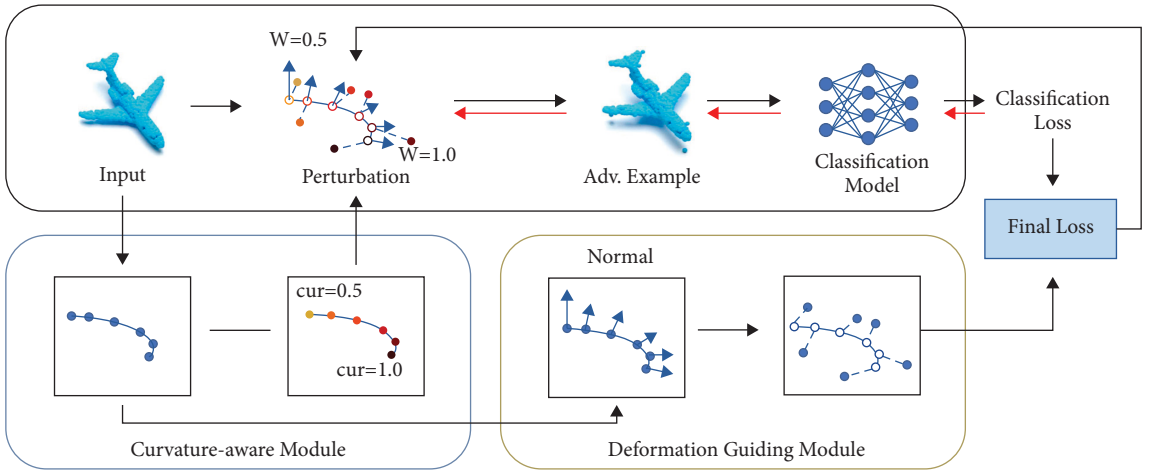


Figure 2: Demonstration of the NormalAttack framework: given an input point cloud, the curvature-aware module enforces the perturbation to be located at regions with larger curvature, and the deformation guiding module leads the perturbation to be along normals, resulting in an imperceptible adversarial attack.

Therefore, by enforcing the value of D_{tangent} to be small, the perturbation is concentrated along the normal directions.

4.1.1. Discussion with ITA. Compared with ITA [34], our DGM allows larger freedom along the tangent plane, and thus more feasible solutions can be searched. Besides, adding these offsets in the tangent direction can be considered as applying an additional re-sampling process, thus making the surfaces after perturbation more smooth, as shown in Figure 3.

4.2. Curvature-Aware Module. To facilitate a flexible perturbation scheme that allows different perturbation magnitudes for different points, e.g., perturb more on regions that have larger tolerances, we devise a curvature-aware module (CM).

Specifically, CM first calculates the curvature of each point, e.g., p_i,

$$\text{cur}_i = \frac{1}{k}\sum_{q\in\mathcal{N}_{p_i}}\left|\left\langle\frac{(q-p_i)}{\|q-p_i\|_2}, n_i\right\rangle\right|, \tag{6}$$

where $\mathcal{N}_p$ is the k-nearest neighbors of p_i.

Then, CM calculates the magnitude weight for each point via applying the sigmoid function:

$$w(i) = \frac{1}{1+e^{-t\cdot\text{cur}_i}}, \tag{7}$$

where t is a temperature scaling parameter.

Therefore, by multiplying the magnitude weight with the original perturbation step size, more perturbation is applied to the regions with larger curvature.

4.3. The Whole Attack Framework. Given the clean point cloud $\mathcal{P}$, our NormalAttack framework first randomly initializes the perturbation to form the adversarial point cloud $\mathcal{P}_1^{\text{adv}}$, and then optimize it iteratively, i.e., $\mathcal{P}_N^{\text{adv}}$.

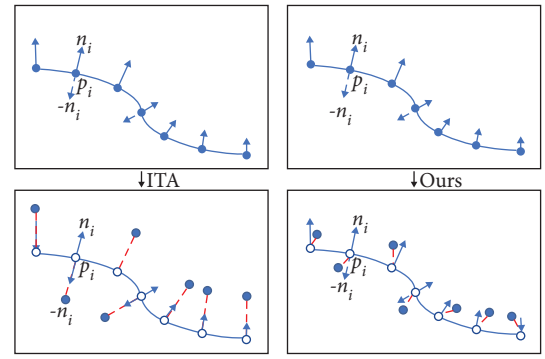


Figure 3: Diagram comparison between the ITA attack method and our NormalAttack framework.

Specifically, the objective loss function of our NormalAttack framework is defined as

$$\begin{aligned}\widehat{\mathcal{J}}\left(F,\mathcal{P}_N^{\text{adv}}\right) = &-L_{\text{class}}\left(F\left(\mathcal{P}_N^{\text{adv}}\right)\right) + \lambda_1 D_{\text{tangent}}\left(\mathcal{P}, P_N^{\text{adv}}\right)\\ &+ \lambda_2 D_h\left(\mathcal{P}, P_N^{\text{adv}}\right) + \lambda_3 D_c\left(\mathcal{P}, \mathcal{P}_N^{\text{adv}}\right),\end{aligned} \tag{8}$$

where L_{class} is the cross-entropy loss for category classification, D_c is the Chamfer distance (CD) loss, and D_h is the Hausdorff distance (HD) loss. By applying gradient descent following (4) iteratively, the adversarial point clouds can be refined via

$$\mathcal{P}_{N+1}^{\text{adv}} = \mathcal{P}_N^{\text{adv}} - \{\mathbf{W}\}(i)\cdot\epsilon\cdot\text{sign}\left(\nabla_{\mathcal{P}}\widehat{\mathcal{J}}\left(F,\mathcal{P}_N^{a\,dv}\right)\right). \tag{9}$$

where the i-th element of $\mathbf{W}$ is $w(i)$. With the help of DGM and CM, NormalAttack deforms the shape of point clouds along normals in a curvature-aware manner, and thus makes adversarial attack imperceptible. Besides, we validate that NormalAttack is also hard to defend and highly transferable in the following experiments.

Table 1: Comparison on the perturbation sizes of different methods required to achieve their best attack success rates.

Attack model	Methods	Attack success rate (%)	Perturbation size		
			l_2-norm	HD	CD
PointNet++	FGSM	100.00	5.5426	0.0193	0.0091
	I-FGSM	100.00	0.6719	0.0063	0.0004
	3D-ADV	100.00	0.3248	0.0381	0.0003
	GeoA^3	100.00	0.4772	0.0357	0.0064
	ITA*	100.00	0.6507	0.0054	0.0004
	Ours	100.00	0.5780	0.0050	0.0003
DGCNN	FGSM	100.00	6.6511	0.0193	0.0093
	I-FGSM	100.00	0.9650	0.0088	0.0007
	3D-ADV	100.00	0.3326	0.0475	0.0005
	GeoA^3	100.00	0.4933	0.0402	0.0076
	ITA*	100.00	1.1601	0.0106	0.0010
	Ours	100.00	0.8232	0.0077	0.0005
PointConv	FGSM	100.00	3.8798	0.0185	0.0050
	I-FGSM	100.00	0.9231	0.0089	0.0007
	3D-ADV	100.00	1.1230	0.0077	0.0011
	GeoA^3	100.00	2.3029	0.0037	0.0005
	ITA*	100.00	1.0034	0.0095	0.0008
	Ours	100.00	0.7735	0.0076	0.0005

5. Experiments

5.1. Implementation. For the attack objective function, i.e., (8), we set the weighting parameters with $\lambda_1 = 1.0$, $\lambda_2 = 0.1$, and $\lambda_3 = 1.0$. For the curvature-aware module, we set $k = 12$ and $t = 20$. We implement NormalAttack and reproduce all the models with PyTorch and report the results on a workstation with an Intel Xeon E5-2678 CPU@2.5 Hz and 64 GB of memory using a single RTX 2080Ti GPU.

5.2. Experimental Setup

5.2.1. Dataset. We evaluate the attack method on ModelNet40 [35], a dataset that is widely used for 3D point cloud classification tasks and contains 40 of the most common object classes, consisting of 12,311 CAD models, of which 9843 models are used for training and another 2468 for testing.

5.2.2. Models. We choose three representative 3D point cloud classification models, such as PointNet++ [2], DGCNN [3], and PointConv [23] for evaluating that attack performance of our NormalAttack framework. These models are trained on the training data following their original papers.

5.2.3. Baseline Attack Methods. We compare the NormalAttack framework with five baseline attack methods, e.g., FGSM, I-FGSM [36], 3D-ADV [11], GeoA^3 [32], and ITA* [34]. Note that ITA* indicates the method that implements the directional perturbation module of ITA with the adversarial transformation model for black-box attack ablated. Besides, since it is not open-sourced, we reimplement it by ourselves.

5.2.4. Defense Methods. We adopt three adversarial defense methods: statistical outlier removal (SOR) [37], simple random sampling (SRS) [28], and denoiser and upsampler network (DUP-Net) [37]. For SOR, we set the number of points to be removed to be 128; for SRS, we set the number of points to be sampled to be 100; for DUP-Net, we set the number of points in the k-neighborhood to 2, the variance of the allowed point cloud distribution to 1.1, and the minimum number of input points for upsampling to 1024.

5.2.5. Evaluation Metrics. We evaluate the effectiveness of our novel NormalAttack framework using the attack success rate (ASR), i.e., the rate of adversarial samples that can successfully fool the classifiers. Besides, we evaluate the imperceptibility by measuring the perturbation size between the original point clouds and their corresponding adversarial examples using three commonly metrics: l_2-norm distance, Chamfer distance (CD), and Hausdorff distance (HD). Note that these three imperceptibility metrics are measured on the adversarial point clouds generated by these methods that just achieve the best attack success rates in the parameter tuning process, e.g., enlarging perturbation step size and iteration.

5.3. Performance Comparison

5.3.1. Quantitative Results. To demonstrate the imperceptibility of our NormalAttack, we compare the distance metrics with FGSM, I-FGSM [36], 3D-ADV [11], GeoA^3 [32], and ITA* [34]. The results reported in Table 1 show that all these methods can achieve 100% attack success rates. In particular, our NormalAttack framework requires the lowest CD and HD distances and a medium l_2-norm to achieve it on all three network models, significantly better than state-of-the-art

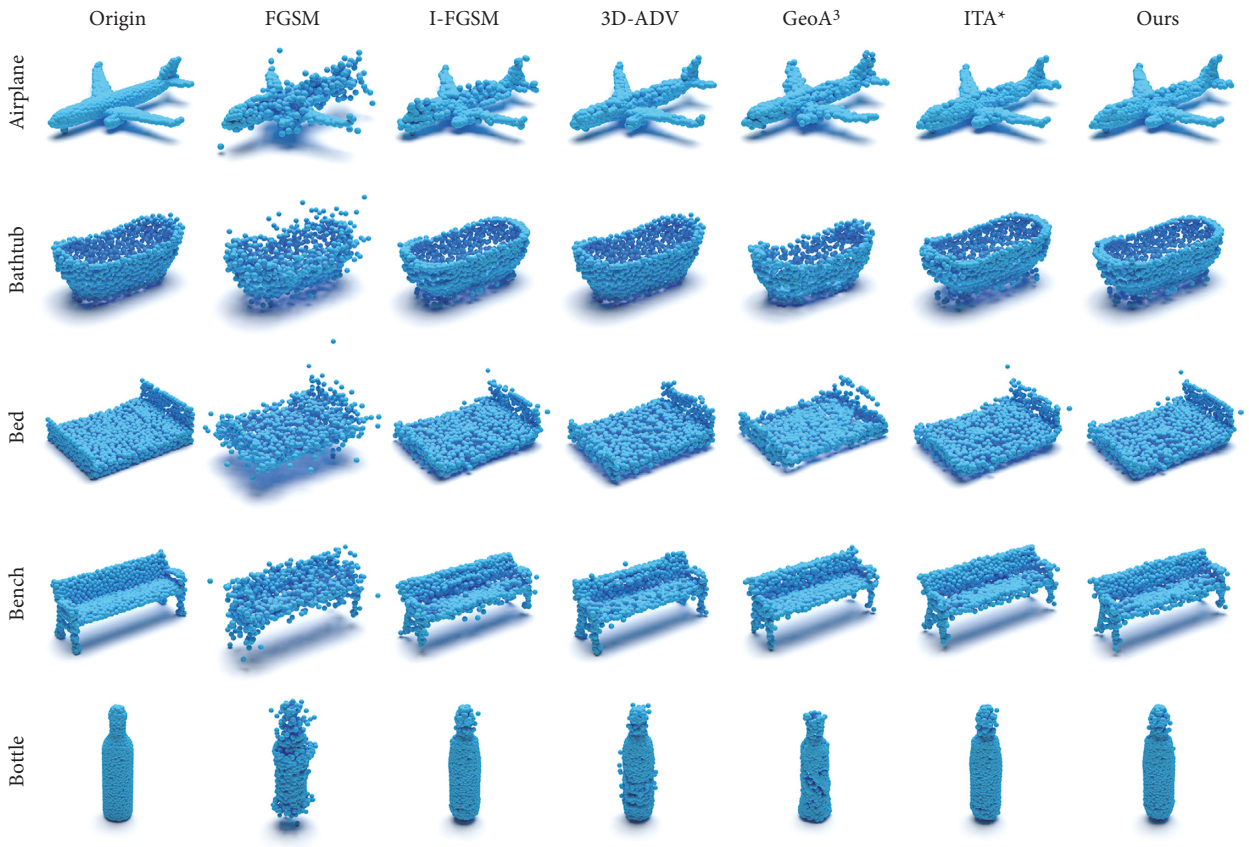


Figure 4: Visualization of original and adversarial point clouds generated by different models for attacking PointNet++.

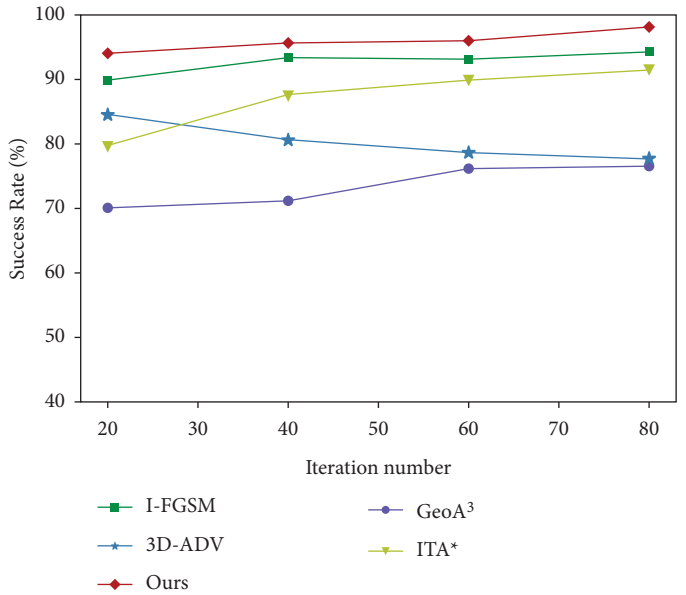


Figure 5: Comparison on the attack success rates of iterative-based attack methods at different iterations.

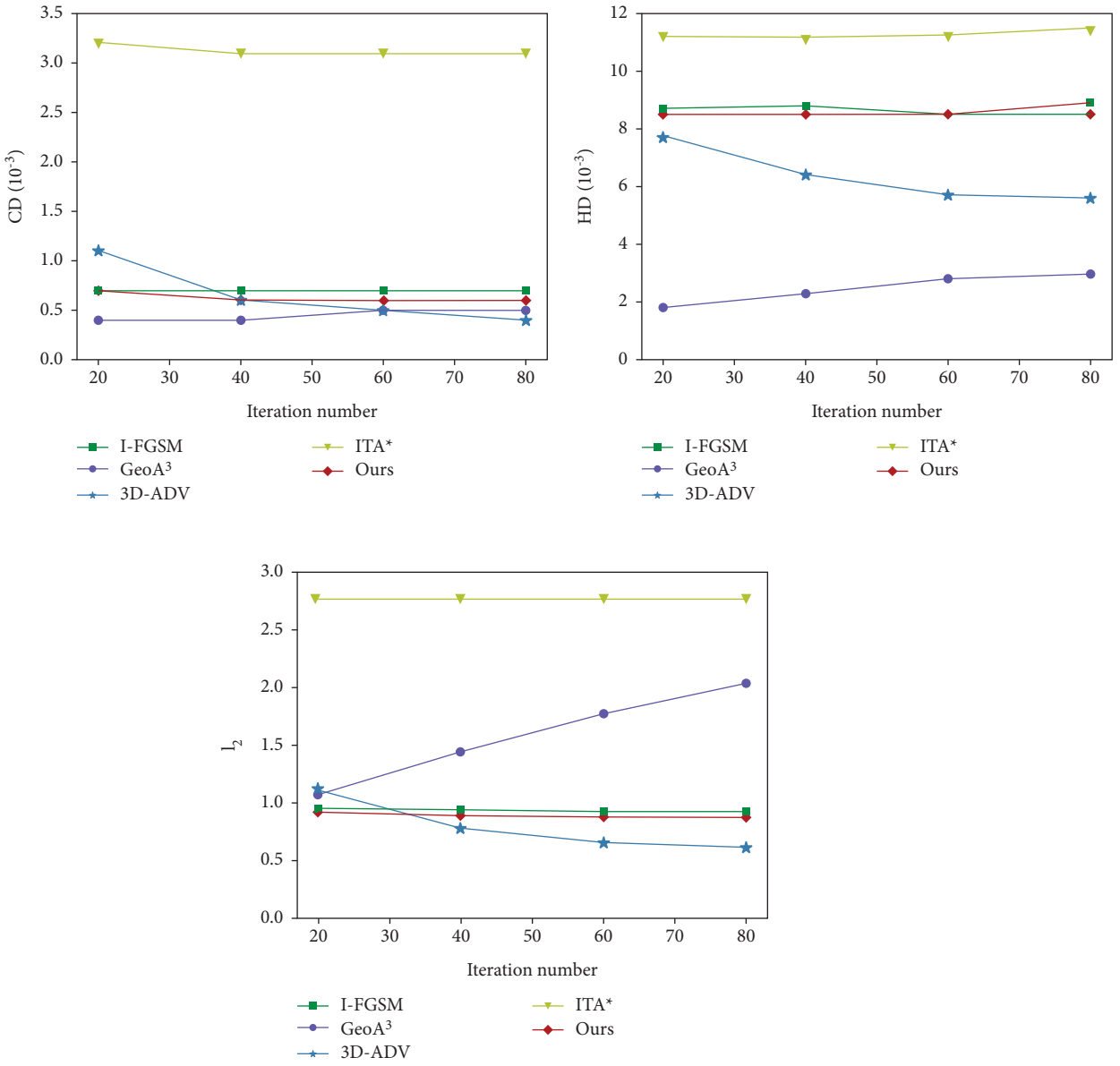


Figure 6: Comparison on the imperceptibility of iterative-based attack methods at different iterations.

Table 2: Comparison on the attack success rates of different methods with and without applying defense methods.

Attack	FGSM	I-FGSM	3D-ADV	GeoA^3	ITA*	Ours
No defense	100.00	100.00	100.00	100.00	100.00	100.00
SRS	9.68	61.53	22.53	67.61	73.74	74.07
SOR	6.26	74.50	17.19	62.47	84.46	87.26
DUP-net	4.38	22.20	5.44	22.63	22.17	24.72

methods. Therefore, we conclude that our proposed NormalAttack framework is imperceptibility.

5.3.2. Visualization Results. To better demonstrate the advantage of our NormalAttack framework in imperceptibility, we visualize the generated adversarial point clouds by different methods in Figure 4. It can be seen that most adversarial point clouds have highly visible outliers, except those generated by ITA* and ours, thus validating the usefulness of applying perturbation along normal.

5.3.3. Evaluation on Efficiency. Since efficiency is also an important factor to perform adversarial attacks, we compare of ours with other iterative-based methods, e.g., I-FGSM, 3D-ADV, GeoA^3, and ITA* on attacking PointNet++. Specifically, we choose the iterations of 20, 40, 60, and 80, and report the attack success rate of all the methods at these iterations in Figure 5. It can be seen that the attack success rates of I-FGSM, GeoA^3, ITA*, and ours increase with larger iterations, while that of 3D-ADV drop slightly. For all 80 iterations, our NormalAttack achieves the highest attack success rate, and the value is slightly lower than 100% at the 80 iteration, validating the efficiency of NormalAttack.

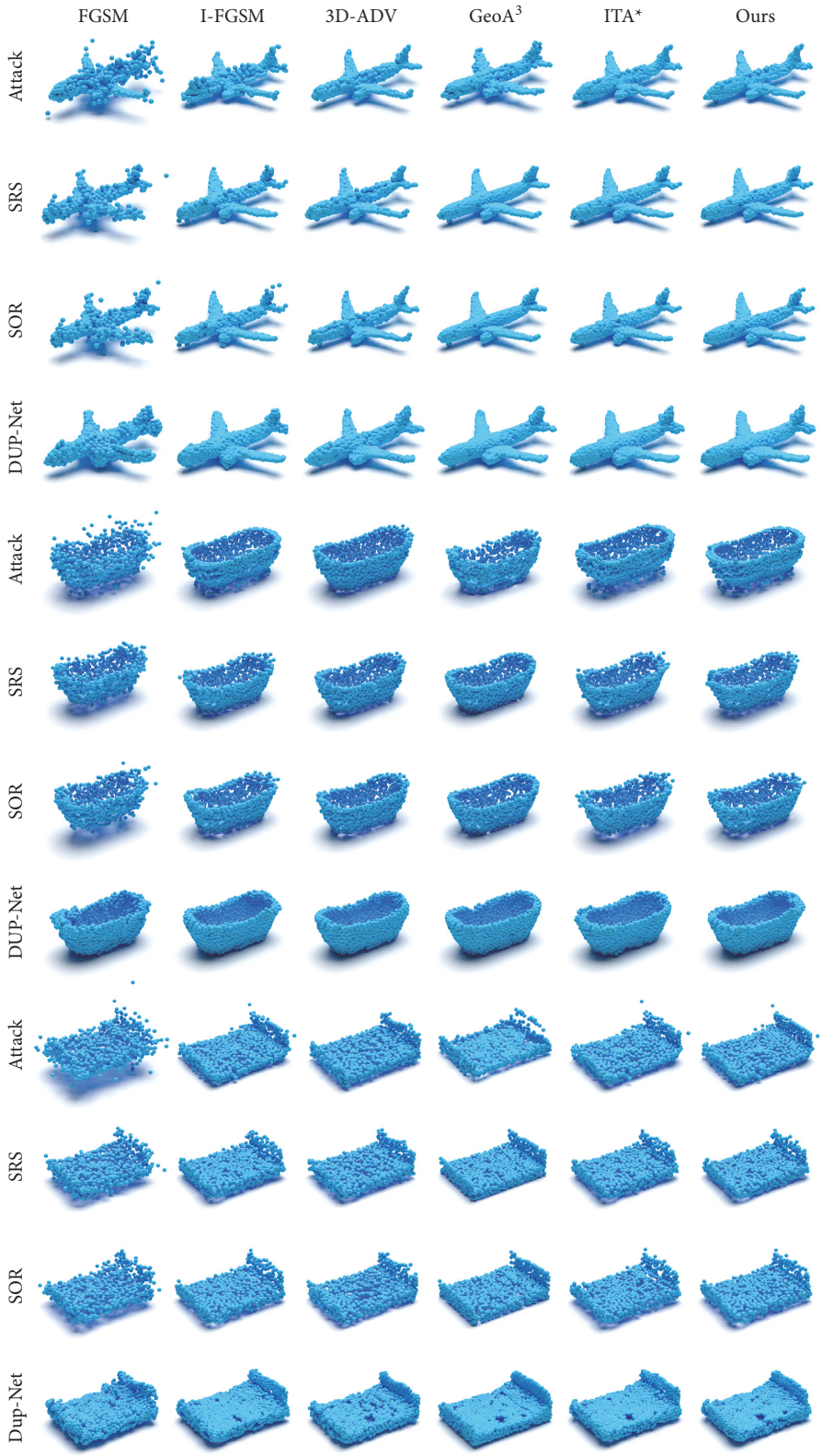


Figure 7: Visualization of generated adversarial point clouds for attacking PointNet++ by different attack methods after applying different adversarial defense solutions.

Besides, we also report the perturbation sizes brought by these attack methods at the same iterations in Figure 6. It can be seen that the CD and l_2-norm distances of ours are small while the HD distance of ours is in a moderate level, validating that our NormalAttack framework achieves high attack success rate and imperceptibility at the same time.

Table 3: Comparison on the transferability performance of different attack methods.

Source	PointNet++		DGCNN		PointConv	
Target	DGCNN	PointConv	PointNet++	PointConv	PointNet++	DGCNN
I-FGSM	24.40	22.40	32.53	32.42	38.02	33.41
3D-ADV	20.44	19.56	32.75	33.96	22.86	54.73
GeoA^3	18.35	22.42	21.65	30.55	17.14	23.19
ITA*	21.42	21.86	33.18	30.98	34.72	31.75
Ours	21.21	22.86	39.23	38.24	34.94	40.98

Table 4: Attack success rates and perturbation sizes of the full NormalAttack and the ones with the curvature-aware module (CM) and deformation guiding module (DGM) ablated.

Victim model	Attack method	Attack success rate (%)	Perturbation size		
			l_2-norm	HD	CD
PointNet++	Ours	97.58	0.5780	0.0050	0.0003
	Ours w/o CM	98.57	0.5882	0.0052	0.0003
	Ours w/o DGM	98.57	0.5781	0.0052	0.0003
DGCNN	Ours	86.59	0.9391	0.0089	0.0007
	Ours w/o CM	87.03	0 t.9736	0.0090	0.0007
	Ours w/o DGM	88.02	0.9409	0.0090	0.0007
PointConv	Ours	94.06	0.8673	0.0085	0.0006
	Ours w/o CM	93.73	0.8850	0.0086	0.0006
	Ours w/o DGM	94.17	0.8680	0.0086	0.0006

Table 5: Attack success rates and perturbation sizes of our NormalAttack framework with different values of t.

Victim model	t	Attack success rate (%)	Perturbation size		
			l_2-norm	HD	CD
PointNet++	20	98.24	0.5782	0.0050	0.0003
	30	98.79	0.5804	0.0051	0.0003
	40	98.35	0.5829	0.0050	0.0003
DGCNN	20	88.13	0.9400	0.0088	0.0007
	30	87.91	0.9486	0.0089	0.0007
	40	87.25	0.9525	0.0089	0.0007
PointConv	20	94.73	0.8678	0.0085	0.0006
	30	93.52	0.8726	0.0085	0.0006
	40	93.63	0.8760	0.0086	0.0006

5.3.4. Attack performance against defenses. To evaluate the attack performance of our NormalAttack framework against defenses, we compare it with FGSM, I-FGSM, 3D-ADV, GeoA^3, and ITA* on PointNet++ with applying three adversarial defense methods, i.e., SRS, SOR, and DUP-Net. The results reported in Table 2 show that the attack success rates of all six attack methods, including ours drop after applying any one of the three defense strategies. In particular, I-FGSM, GeoA^3, ITA*, and NormalAttack still obtain more than 60% success rates after applying SRS and SOR, validating their effectiveness in handling traditional geometric defense methods. However, after applying the DNN-based DUP-Net, their attack performance drops significantly. In all cases, NormalAttack maintains the largest attack success rates, validating its superiority. To investigate why the performance of state-of-the-art methods drops but ours not, we visualize the results after applying the defense methods in Figure 7. It can be seen that the original outliers generated by these methods for fooling network models are filtered and thus lead to performance drops. Instead, our NormalAttack framework attacks models without bringing clearly visible outliers and thus is only slightly affected by the defenses.

5.3.5. Transferability. To investigate the transferability performance of our NormalAttack framework, we compare it with state-of-the-art iterative-based attack methods, e.g., I-FGSM, 3D-ADV, GeoA^{3,} and ITA*, by feeding adversarial point clouds generated by one network model to others. Specifically, we report the adversarial transferability among PointNet++, DGCNN, and PointConv in Table 3. It can be seen that our NormalAttack framework performs the best when transforming from DGCNN to other models, and ranks in the forefront in the other two situations, validating its transferability.

5.4. Ablation Studies and Other Analysis

5.4.1. Curvature-Aware Module. To demonstrate the importance of the curvature-aware module, we compare the results of the full NormalAttack framework with the other framework whose curvature-aware module is ablated with 80 iterations. The results reported in Table 4 show that the attack success rate of the ablated framework is slightly higher than that of the full one, but also with higher distance metrics, validating that the curvature-aware module is critical for maintaining imperceptibility.

5.4.2. Deformation Guiding Module. To demonstrate the importance of the deformation guiding module, we compare the results of the full NormalAttack framework with the

other framework whose deformation guiding module is ablated with 80 iterations. The results reported in Table 4 show that the attack success rate of the ablated framework is slightly higher than that of the full one, and the distance metrics are also much higher, validating that the deformation guiding module is critical for maintaining imperceptibility.

5.4.3. Parameter Analysis on t. We also investigate the effects of the temperature scaling parameter t in the curvature adaptation module. Specifically, we apply NormalAttack with different values of t for 80 iterations to attack PointNet++, DGCNN, and PointConv. The results reported in Table 5 show that both the attack success rate and perturbation size are better when $t = 20$. Therefore, we set $t = 20$ in all the experiments.

6. Conclusion

In this paper, we have proposed a novel NormalAttack framework toward imperceptible adversarial attack on point clouds. The key of the framework is to enforce the perturbation to be concentrated along normals to deform the underlying surface of 3D point clouds and perturb more on regions with larger curvature. Extensive experiments validate the effectiveness of NormalAttack. We hope our work can inspire more research on utilizing geometric properties of point clouds to investigate adversarial robustness.

Data Availability

All datasets that support the findings of this study are available publicly.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Keke Tang, Yawen Shi, and Jianpeng Wu contributed equally to this work.

Acknowledgments

This work was supported in part by the National Key Research and Development Project of China (2020AAA0107704), the National Natural Science Foundation of China (62102105, 62073263, and 61902082), Guangdong Basic and Applied Basic Research Foundation (2020A1515110997, 2022A1515011501, and 2022A1515010138), the Science and Technology Program of Guangzhou (202002030263, 202102010419 and 202201020229), the Open Project Program of the State Key Lab of CAD and CG (Grant no. A2218), Zhejiang University.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] C. R. Qi, L. Yi, H. Su, and Guibas, "Pointnet++: deep hierarchical feature learning on point sets in a metric space," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 5099–5108, Long Beach, CA, USA, December 2017.

[3] Y. Wang, Y. Sun, Z. Liu, M. M. Sarma, J. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics*, vol. 38, no. 5, pp. 1–12, 2019.

[4] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M Bennamoun, "Deep learning for 3d point clouds: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 12, pp. 4338–4364, 2021.

[5] Y. Chen, W. Peng, K. Tang, G. Khan, M. Wei, and M. Fang, "Pyrapvconv: efficient 3d point cloud perception with pyramid voxel convolution and sharable attention," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–9, 2022.

[6] C. Szegedy, W. Zaremba, I. Sutskever et al., "Intriguing properties of neural networks," in *Proceedings of the ICLR 2014 submission conference review*, Banff, Canada, June 2014.

[7] J. Levinson, J. Askeland, J. Becker et al., "Towards fully autonomous driving: systems and algorithms," in *Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 163–168, IEEE, Baden-Baden, Germany, June 2011.

[8] K. Tang, P. Song, and X. Chen, "Signature of geometric centroids for 3d local shape description and partial shape matching," in *Asian Conference on Computer Vision*, pp. 311–326, Springer, New York, NY, USA, 2016.

[9] K. Tang, P. Song, and X. Chen, "3d object recognition in cluttered scenes with robust shape description and correspondence selection," *IEEE Access*, vol. 5, no. 99, pp. 1833–1845, 2017.

[10] N. Lin, Y. Li, K. Tang et al., "Manipulation planning from demonstration via goal-conditioned prior action primitive decomposition and alignment," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1387–1394, 2022.

[11] C. Xiang, C. R. Qi, and B. Li, "Generating 3d Adversarial point Clouds," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9136–9144, Long Beach, CA, USA, June 2019.

[12] T. Zheng, C. Chen, J. Yuan, L. Bo, and R. Kui, "Pointcloud saliency maps," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1598–1606, Buffalo, NY, USA, 2019.

[13] A. Chakraborty, M. Alam, V. Dey, Chattopadhyay, and D. Mukhopadhyay, "A survey on adversarial attacks and defences," *CAAI Transactions on Intelligence Technology*, vol. 6, no. 1, pp. 25–45, 2021.

[14] K. Ren, T. Zheng, Qin, and X. Liu, "Adversarial attacks and defenses in deep learning," *Engineering*, vol. 6, no. 3, pp. 346–360, 2020.

[15] X. Yuan, P. He, Zhu, and X. Li, "Adversarial examples: attacks and defenses for deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 9, pp. 2805–2824, 2019.

[16] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 39–57, San Jose, CA, USA, May 2017.

[17] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proceedings of the International Conference on Machine Learning*, Long Beach, CA, USA, January 2015.

[18] D. Liu, R. Yu, and H. Su, "Extending adversarial attacks and defenses to deep 3d point cloud classifiers," in *Proceedings of the IEEE International Conference on Image Processing (IEEE ICIP)*, pp. 2279–2283, Taipei, Taiwan, September 2019.

[19] X. Gu, S. J. Gortler, and H. Hoppe, "Geometry images," *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 355–361, 2002.

[20] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *Proceedings of the2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 945–953, Santiago, Chile, 2015.

[21] Y. Zhou and O. Tuzel, "Voxelnet: End-To-End Learning for point Cloud Based 3d Object Detection," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4490–4499, New Orleans, USA, 2018.

[22] C. R. Qi, H. Su, K. Mo, and J. G. Leonidas, "Pointnet: deep learning on point sets for 3d classification and segmentation," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 652–660, Honolulu, HI, USA, 2017.

[23] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep Convolutional Networks on 3d point Clouds," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9621–9630, Long Beach, CA, USA, June 2019.

[24] Y. Li, R. Bu, M. Sun, and W. Wu, "Pointcnn: convolution on x-transformed points," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 820–830, Montreal, Canada, 2018.

[25] Y. Dong, F. Liao, T. Pang et al., "Boosting Adversarial Attacks with Momentum," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9185–9193, Salt Lake City, UT, USA, June 2018.

[26] S. M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A Simple and Accurate Method to Fool Deep Neural Networks," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2574–2582, Las Vegas, NV, USA, June 2016.

[27] M. Wicker and M. Kwiatkowska, "Robustness of 3d deep learning in an adversarial setting," *CVPR*, vol. 11, no. 767–11, p. 775, 2019.

[28] J. Yang, Q. Zhang, R. Fang, B. Ni, J. Liu, and Q. Tian, "Adversarial Attack and Defense on point Sets," 2019, https://arxiv.org/abs/1902.10899.

[29] K. Lee, Z. Chen, X. Yan, U. Raquel, and Y. Ersin, "Shapeadv: Generating Shape-Aware Adversarial 3d point Clouds," 2020, https://arxiv.org/abs/2005.11626.

[30] J. Kim, B. S. Hua, T. Nguyen, and S. -K. Yeung, "Minimal adversarial examples for deep learning on 3d point clouds," in *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7797–7806, Montreal, QC, Canada, October 2021.

[31] T. Tsai, K. Yang, Y. Ho, and Y. Jin, "Robust adversarial objects against deep learning models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 1, pp. 954–962, AAAI, Washington, DC, USA, 2020.

[32] Y. Wen, J. Lin, K. Chen, C. L. P. Chen, and J. Kui, "Geometry-aware generation of adversarial point clouds," *IEEE TPAMI*, vol. 44, no. 6, p. 1, 2020.

[33] H. Zhou, D. Chen, J. Liao et al., "Lg-gan: label guided adversarial network for flexible targeted attack of point cloud based deep networks," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, no. 10, p. 365, Seattle, WA, USA, June 2020.

[34] D. Liu and W. Hu, "Imperceptible Transfer Attack and Defense on 3d point Cloud Classification," 2021, https://arxiv.org/abs/2111.10990.

[35] Z. Wu, S. Song, A. Khosla et al., "3d Shapenets: A Deep Representation for Volumetric Shapes," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1912–1920, Boston, MA, USA, June 2015.

[36] X. Dong, D. Chen, H. Zhou, G. Hua, W. Zhang, and N. Yu, "Self-robust 3d point recognition via gather-vector guidance," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 11, no. 513–11, p. 521, Seattle, WA, USA, June 2020.

[37] H. Zhou, K. Chen, W. Zhang, H. Fang, W. Zhou, and N. Yu, "Dup-net: Denoiser and Upsampler Network for 3d Adversarial point Clouds Defense," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1961–1970, Seoul, Korea, October 2019.