Geometry-Aware Generation of Adversarial Point Clouds

Yuxin Wen, Jiehong Lin, Ke Chen, C. L. Philip Chen, *Fellow, IEEE*, and Kui Jia

Abstract—Machine learning models are shown to be vulnerable to adversarial examples. While most of the existing methods for adversarial attack and defense work on 2D image domains, a few recent ones attempt to extend the studies to 3D data of point clouds. However, adversarial results obtained by these methods typically contain point outliers, which are both noticeable and easier to be defended by simple techniques of outlier removal. Motivated by the different mechanisms when humans perceive 2D images and 3D shapes, we propose in this paper a new design of *geometry-aware objectives*, whose solutions favor (discrete versions of) the desired surface properties of smoothness and fairness. To generate adversarial point clouds, we use a misclassification loss of targeted attack that supports continuous pursuing of more malicious signals. Regularizing the targeted attack loss with our proposed geometry-aware objectives gives our proposed method of *Geometry-Aware Adversarial Attack ($GeoA^3$)*. Results of $GeoA^3$ tend to be more adversarial, arguably less defendable, and of the key adversarial characterization of being imperceptible to humans. While the main focus of this paper is to learn to generate adversarial point clouds, we also present a simple but effective algorithm termed *Iterative Tangent Jittering (IterTanJit)*, in order to preserve surface-level adversarial effects when re-sampling point clouds from the surface meshes reconstructed from adversarial point clouds. We quantitatively evaluate our methods on both synthetic and physical object models in terms of attack success rate and geometric regularity. For qualitative evaluation, we conduct subjective studies by collecting human preferences from Amazon Mechanical Turk. Comparative results in comprehensive experiments confirm the advantages of our proposed methods over existing ones. We make our source codes publicly available.

1 Introduction

Modern machine learning models, particularly those based on deep networks, have been achieving remarkable success on a variety of semantic tasks, with classification of 2D images [1], [2], [3] and 3D point clouds [4], [5], [6], [7] as the prominent representatives. In spite of the success, these models are vulnerable to *adversarial examples* [8] — specially crafted perturbations of input data that are as small as imperceptible to our humans would cause failure of these classification models. Such a phenomenon prevails across models and data types [9], [10], [11]. The existence of adversarial examples triggers a great amount of research focusing either on attack/defense studies for safety-critical applications [12], [13], [14], or on the robustness analysis of machine learning models [15], [16], [17].

The seminal work of Szegedy *et al.* [8] discovers adversarial 2D images from analyzing the classification robustness of deep networks. Following [8], subsequent research plays attack-and-defense games and formalizes the attack problem by proposing various algorithms to search for adversarial noises [18], [19], [20], where an important characterization of the produced adversarial images is *imperceptibility*. To achieve imperceptibility, these methods typically constrain the l_p-norms of adversarial noise to be small on the 2D image domain. Effectiveness of such a scheme is grounded on how our humans perceive 2D images. Indeed, when adding noise of small magnitude to a benign image, human perception is overwhelmed by the appearance patterns contained in the original image, rather than by the added high-frequency but small-magnitude noise that has no semantic patterns; adversarial imperceptibility is achieved consequently.

More recently, this general idea of adversarial attack is employed to 3D data of point clouds [11], [21], [22], [23]. They learn to either perturb individual points contained in a point cloud or attach additional points to it, in order to make the resulting point cloud be misclassified by a point set classifier of interest [4], [5], [6]. To achieve the objective of imperceptibility, these methods follow works of adversarial 2D images, and constrain the adversarial point cloud such that it is close to the benign one under certain distance metrics of point set. This seems a straightforward technical extension at a first glance. Unfortunately, adversarial point clouds obtained by these methods typically contain point outliers, and these point outliers are particularly noticeable when humans perceive the underlying surface represented by the point cloud, as shown in Figure 1 — one can intuitively think of outliers of a point cloud as those away from the underlying surface. We analyze in this work the inefficacy of directly applying the methodology of generating adversarial 2D images to generation of 3D point clouds, and argue that the inefficacy may be attributed to the sharp difference between our human perception of 2D images and that of 3D shapes. Psychophysical evidence shows that humans perceive object surface shapes from combined sources of motion, texture, shading, boundary contour, etc [24], where object boundary contours are par-

Y. Wen, J. Lin, K. Chen, and K. Jia are with the School of Electronic and Information Engineering, C. L. Philip Chen is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China, emails: wen.yuxin@mail.scut.edu.cn, lin.jiehong@mail.scut.edu.cn, chenk@scut.edu.cn, philip.chen@ieee.org, kuijia@scut.edu.cn.
Corresponding author: Kui Jia.

This work is supported in part by the National Natural Science Foundation of China (Grant No.: 61771201, 61902131), the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (Grant No.: 2017ZT07X183), and the Fundamental Research Funds for the Central Universities (Grant No.: 2019MS022).

ticularly effective to visually tell surface shapes [25], [26]. As approximate surface representations, point cloud data contain no texture and shading, and our shape perception of poind clouds is mostly from their point-wise depths and boundary contours. Consequently, point outliers produced by existing methods [11], [22] would cause a spiky boundary perception of the surfaces and thus draw our humans' attention. In addition, these methods produce adversarial point clouds that tend to be defended easily, by removing the generated point outliers via simple statistical techniques [27].

To address the issues analyzed above, we are motivated by the fact that an adversarial point cloud, as a discrete approximation of object surface, should satisfy (discrete versions of) the general surface properties of *smoothness* and *fairness* [28], which concern with the continuity and variation of (partial) derivatives of a parametric surface function, in particular with the curvatures of local surface patches. To achieve the objective of imperceptibility, we propose a new solution of *geometry-aware objectives*, whose design is to make the adversarial point cloud bear the aforementioned general surface properties, while being close to the benign point cloud under distance metrics of point set. Technically, our proposed geometry-aware objectives enhance classical distance terms of point set with a term that promotes consistency of local curvatures between the adversarial and benign point clouds. In this work, we consider the setting of targeted attack [11], [19] and use a misclassification loss that is able to achieve higher malicious levels when compared with the traditional margin-based ones [11], [19]. Regularizing the more aggressive misclassification loss with our proposed geometry-aware objectives gives our proposed method of *Geometry-Aware Adversarial Attack (GeoA^3)*, which is expected to produce more adversarial, arguably less defendable, point clouds without introducing noticeable modifications.

While our main focus of this paper is to learn to generate adversarial point clouds, which would cause safety-critical issues practically by guiding, for example, a LiDAR spoofer [29] to fool the 3D sensor, it is more desirable to make the underlying surfaces represented by the point clouds be adversarial; otherwise, a benign surface re-sampling would possibly defend the adversarial attack. As an attempt towards generation of adversarial surfaces via generation of adversarial point clouds, we present in this work a simple but effective solving algorithm termed *Iterative Tangent Jittering (IterTanJit)*. When optimizing the objective of *GeoA^3*, IterTanJit introduces per iteration point-wise jittering of 3D coordinates on the tangent planes associated with individual points contained in the intermediate update of adversarial point cloud; by accumulating iteratively, IterTanJit pursues optimization directions of adversarial updates that account more for surface deformation.

We quantitatively evaluate adversarial point clouds generated by different methods in terms of attack success rate, under the state-of-the-art defense [27], and geometric regularity; for the later evaluation we also propose a new measure that is based on the same desired surface property of low curvature. For qualitative evaluation, we conduct subjective studies by collecting human preferences from Amazon Mechanical Turk. Evaluation of surface-level adversarial effects is conducted by first reconstructing object meshes from the obtained adversarial point clouds, and then evaluating the remained adversarial effects via re-sampling point clouds from the reconstructed mesh surfaces. Physical attack is conducted similarly by 3D printing the mesh reconstructions and then re-scanning the printed objects to have the point clouds to be evaluated. We present comprehensive experiments which show the advantages of our proposed methods over existing ones.

1.1 Related Works

We briefly review existing works that are closely related to the present one. We organize the review into the following three lines of research.

Adversarial Attack of 2D Images – The existence of adversarial examples w.r.t. deep image classification networks is first suggested in [8]. Subsequent research proposes various methods to generate adversarial examples that are less likely to be perceived by our humans, yet still able to attack the classification models. Representative methods include Fast Gradient Sign Method (FGSM) [18], C&W attack [19], and Projected Gradient Descent (PGD) [20]. They typically search for pixel-level noise of small magnitude on the domains of benign images, by optimizing misclassification loss functions defined on the classification models, such that superimposing the obtained noisy images onto the respective benign ones produces the adversarial results. More specifically, Goodfellow *et al.* [18] find that moving a benign image towards the decision boundary of a classification model via a single step of gradient update is enough to generate the adversarial image. The C&W attack proposed in [19] includes both the misclassification loss and a loss constraining the magnitudes of adversarial noise into the optimization objective, and empirical results show that adversarial images generated by C&W attack are both aggressive and difficult to be perceived. PGD [20] is a multi-step optimization method that can be regarded as an improved version of FGSM. These methods adopt l_p-norms to constrain the magnitudes of adversarial noise, which is shown to be effective to satisfy the adversarial criterion of imperceptibility. Apart from l_p-norms, other measures such as Perceptual Adversarial Similarity Score (PASS) [32] are proposed to account more for the mechanism that humans perceive 2D images, e.g., by measuring the regional illumination, contrast, and structural similarity between the adversarial images and benign ones. Unfortunately, due to the sharp difference between our human perception of 2D images and that of 3D shapes (cf. the discussion in Section 2.1), both l_p-norms and PASS are less relevant for generation of adversarial point clouds that are expected to be imperceptible to our humans.

Adversarial Generation of 3D Point Clouds – Following the studies on adversarial 2D images, there is a recent surge of interest studying attacking point set classification models [4], [5], [6] with adversarial point clouds [11], [21], [22], [23], [33], [34]. Given a benign point cloud, these methods generate adversarial ones either by perturbing the contained individual points [11], [21], [22], [23], by removing some of the points [22], [33], [34], or by attaching additional points to the given point cloud [11], [22]. Intuitively, the

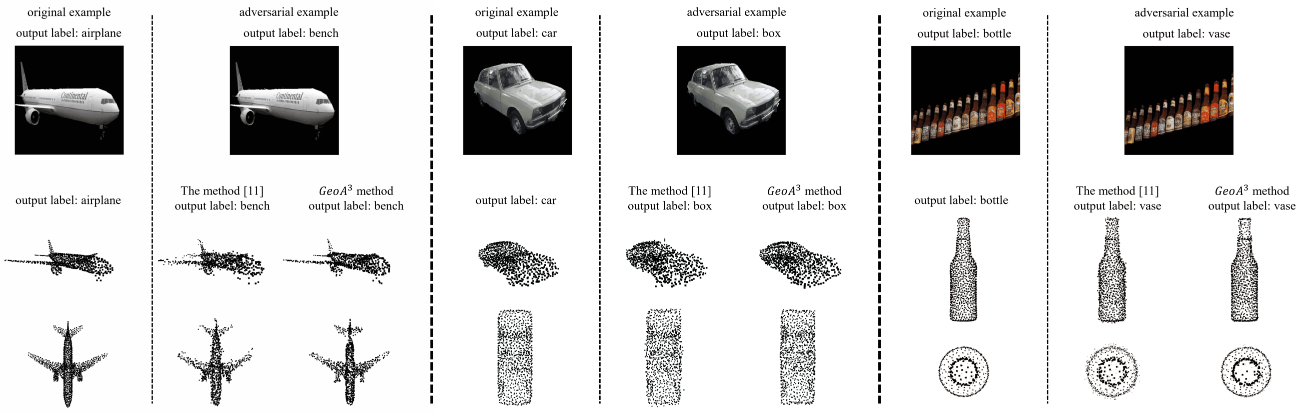


Fig. 1: Adversarial examples of 2D images and 3D point clouds corresponding to the same object categories from the PASCAL3D+ dataset [30]. Adversarial images are obtained by using C&W attack [19] against the Inception-V3 [31] model of image classification. Adversarial point clouds are obtained respectively by using the method [11] and our proposed $GeoA^3$ against PointNet model [4].

strengths of adversarial attack depend on the numbers of points that are free to be adjusted via optimization in the generation process; consequently, point perturbation would arguably be the most effective way while point detachment would be the least effective one — indeed, point detachment could be more of a way to study the point-wise saliency for classification [33]. Technically, existing methods of adversarial point clouds follow their 2D image counterparts; they optimize point-wise coordinate offsets w.r.t. misclassification loss functions defined on point set classification models, where the offset searching is constrained either by l_p-norms or by classical point set distance functions. For example, Xiang *et al.* [11] adopt the framework of C&W attack [19], by respectively using l_2-norm to constrain point-wise perturbations and Chamfer or Hausdorff distances for point attachment; Liu *et al.* [21] follow FGSM [18] by using l_2-norm or its variants as the constraints. In spite of the attacking success achieved by existing methods, however, they tend to generate adversarial point clouds that contain clearly visible outliers. The generation of point outliers both violates the adversarial criterion of imperceptibility and makes the defense easier (e.g., via simple techniques of outlier removal [27]). It is worth noting that a simple attempt to achieve imperceptibility is made in [21] by iteratively projecting the perturbed points back onto the mesh where the benign point cloud is sampled from; unfortunately, attacking success rates become very low by adopting this attempt. A recent work from Tsai *et al.* [35] is motivated to address the issue of point outliers. They propose into the C&W framework a perturbation-constraining regularization which combines a global Chamfer distance and a local term that encourages the compactness of local neighborhoods in the obtained adversarial point cloud (via constraining the averaged distances among points in each neighborhood). In addition, they propose a scheme to guide the gradient update directions of adversarial offsets towards those of mesh-level deformations. However, adversarial results generated by [35] appear to have rugged surfaces, which are easy to be perceived by our humans. In contrast, the geometry-aware objectives proposed in the present work are intuitive and clean, and are well grounded on the surface properties of smoothness and fairness. Comparative experiments show that our results are advantageous over those from [35] in terms of both geometric regularity and attacking success rate.

Deep Learning Point Cloud/Surface Reconstruction – Our proposed geometry-aware generation of adversarial point clouds is also related to the recent methods that learn deep networks to generate 3D surface shapes in the forms of either point cloud [36], [37] or mesh [38], [39], [40]. Among these methods, Fan *et al.* [36] make the first attempt to reconstruct a point cloud object surface from as few as a single image. AtlasNet [38] and subsequent improvements [39], [40] extend for learning to generate object surface meshes by learning to deform the vertices of input, initial meshes. While some learning terms defined in [39], [40] are inspired by smooth surface properties as well, the same source of inspiration would suggest different formulations and learning objectives respectively useful in the contexts of surface reconstruction or generation of adversarial point clouds. More discussions on the technical differences are given in Section 2.2.

1.2 Contributions

Our technical contributions are summarized as follows.

- Motivated by the different human perception of 3D shapes from that of 2D images, we propose a new method, termed *Geometry-Aware Adversarial Attack* (*$GeoA^3$*), for generation of adversarial point clouds whose changes from the benign ones are expected to be imperceptible to humans. $GeoA^3$ is built on a new design of *geometry-aware objectives*, which enables the resulting adversarial point clouds to bear (discrete versions of) the general surface properties of smoothness and fairness.
- The proposed geometry-aware objectives afford a more aggressive attack learning, for which we propose a new misclassification loss of targeted attack that supports continuous pursuing of higher-

level malicious signals. Regularizing the proposed targeted attack loss with geometry-aware objectives is expected to produce more adversarial, arguably less defendable, point clouds without introducing noticeable modifications.

- We also present a simple but effective solving algorithm, termed Iterative Tangent Jittering (IterTanJit), to have the practically desirable surface-level adversarial effects in the generated adversarial point clouds. IterTanJit iteratively accumulates optimization directions of adversarial updates that account more for surface deformation; the expectation is that certain adversarial effects remain after re-sampling point clouds from the surface meshes reconstructed from adversarial point clouds.
- We quantitatively and qualitatively evaluate adversarial point clouds and their remained adversarial effects in the reconstructed mesh models and 3D printed physical models. In addition to attack success rate, we propose a new quantitative measure of geometric regularity that is based on the same desired surface property of low curvature. Qualitative evaluation is also conducted by collecting human preferences from Amazon Mechanical Turk. Comparative results in comprehensive experiments confirm the advantages of our proposed methods over existing ones.

Data, source codes, and pre-trained models are made public at https://github.com/Yuxin-Wen/GeoA3.

2 Adversarial Point Clouds of Object Surface Shapes

Our problem setting assumes the availability of a collection of point clouds in the input space $\mathcal{X}$, of which any point cloud $\mathcal{P} \in \mathcal{X}$ is an approximate shape representation of its underlying object surface $\mathcal{S}$ of a certain category. Each $\mathcal{P}$ contains an orderless set of n points $\{\boldsymbol{p}_i\}_{i=1}^n$, with the corresponding label $y \in \mathcal{Y}$ of object category, where any $\boldsymbol{p} = [x, y, z]^\top \in \mathbb{R}^3$ denotes the coordinates in the Euclidean space. In this work, we focus on machine learning models of 3D point set classification [4], [5], [6], which learn a classifier $f : \mathcal{X} \rightarrow \mathcal{Y}$ and expect $f(\mathcal{P}) = y$ for any input $\mathcal{P}$ with the true label y. Given a learned f, our objective is to obtain from $\mathcal{P}$ an *adversarial* point cloud $\mathcal{P}'$ by perturbing its individual points $\{\boldsymbol{p}_i\}_{i=1}^n$; the obtained $\mathcal{P}'$ would be misclassified by f. Similar to adversarial examples of 2D images [8], the adversarial objective suggests that $\mathcal{P}'$ should be as close to $\mathcal{P}$ as possible under certain distance metrics of point set, such that the change from $\mathcal{P}$ to $\mathcal{P}'$ is *imperceptible* to humans. In the subsequent description, with a slight abuse of notation, we also use $\boldsymbol{x} \in \mathcal{X}$ to represent a signal of either 2D image or 3D point cloud, while using $\mathcal{P} \in \mathcal{X}$ particularly for point cloud data, which are self-clear in the context.

2.1 The (Im)perceptibility of Adversarial Point Clouds

Studies on adversarial examples stem from robust analysis of deep networks, particularly for convolutional networks that are trained to classify 2D images. It is discovered in [8] that for images correctly classified by a network, superimposing certain noise of small magnitude to them can fool the same network; these images are adversarial since magnitudes of the noise are so small such that they are less likely to be perceived by our humans. Most of subsequent research [18], [19], [20], [41] formalizes this problem by proposing various algorithms to search *pixel-wisely independent* noise under the constraints of small l_p-norms on the image domain, whose objectives can be generally written as

$$\min_{\boldsymbol{x}'} C_{Mis}(\boldsymbol{x}') \ \text{ s.t. } \ \|\boldsymbol{x}' - \boldsymbol{x}\|_p \leq \epsilon, \tag{1}$$

where $C_{Mis}(\boldsymbol{x}')$ is a loss term promoting misclassification of $\boldsymbol{x}'$, and ϵ is a small constant. With the constraints, appearance patterns of the resulting images are still dominated by the original ones.

While our intended study on adversarial point clouds generally follows the same methodology, there exists a sharp difference between our human perception of 2D images and that of 3D shapes; psychophysical evidence shows that humans perceive object surface shapes from combined variables of motion, texture, shading, boundary contour, etc [24]. Indeed, for adversarial images, the high-frequency information provided by the added pixel-wisely independent noise is overwhelmed by the appearance patterns contained in the original, clean images; consequently, the noise addition is imperceptible to our humans. As approximate surface representations, point cloud data contain no texture and shading, our shape perception of point clouds is mostly from their point-wise depths and boundary contours. In fact, studies have shown that contours themselves are particularly effective to visually tell surface shapes, and adding additional surface information of shading, texture, and motion gives only small improvements in shape judgment [25], [26]. Consequently, point perturbations would possibly produce point outliers, which make a spiky boundary perception of the surface and thus draw our humans' attention. Our human sensitivity to spiky surface is also supported from study on the aesthetics of object shapes; in [42], experiments find that when visiting an art gallery, visitors prefer shapes with gentle curves over those with sharp points. To have a more intuitive understanding of the difference between our human perception of 2D images and 3D shapes, we show in Figure 1 adversarial examples of 2D images and 3D point clouds corresponding to the same object categories from the PASCAL3D+ dataset [30]. We can hardly tell from Figure 1 the difference between adversarial images and benign ones, but rather easily perceive the manipulation of adversarial point clouds obtained by a representative existing method [11]. In contrast, our proposed $GeoA^3$ largely improves the imperceptibility, while still being able to attack classification successfully — in fact, we will show in Section 5 that our method achieves even higher attacking success rates than the method [11] does. We note that the above surface perception issue is largely overlooked in existing works of adversarial point clouds [11], [22]. In addition to this issue, outlier points to an object surface are also easier to be removed, causing the generated adversaries to be easily defended [27].

To address the aforementioned issues when preparing adversarial point clouds, we are motivated from the fact that as a discrete approximation of an object surface, a point cloud $\mathcal{P}$ satisfies (discrete versions of) the general surface properties of *smoothness* and *fairness* [28], which concern with the continuity and variation of (partial) derivatives of a parametric surface function; technically, the properties concern with the curvatures of local surface patches. Since the adversarial point cloud is expected to be imperceptible to humans, we argue that $\mathcal{P}'$ should satisfy these surface properties as well, with similar degrees of smoothness/fairness to local surface patches of $\mathcal{P}$, while being close to $\mathcal{P}$ when measured under certain distance metrics of point set; otherwise, humans would notice either the global, possibly topological changes of part configuration of the object surface, or those of local surface details. Our analysis leads to technical solutions of *geometry-aware objectives* to generate adversarial point clouds, as presented shortly.

2.2 The Proposed Geometry-Aware Objectives

Analysis in Section 2.1 inspires us to modify a point cloud $\mathcal{P}$ to have $\mathcal{P}'$ in a geometry-aware manner. This can be technically achieved by objectives that constrain the magnitudes of modification under distance metrics of point set, while taking into account the local surface smoothness of the resulting $\mathcal{P}'$, in particular prevention of generating point outliers. In this section, we present these proposed geometry-aware objectives; combining them with an additional misclassification loss gives our proposed method of adversarial point cloud generation, which is to be presented in Section 3.

Chamfer Distance – Given two point sets $\mathcal{P}$ and $\mathcal{P}'$ respectively of n and n' points, the Chamfer distance computes

$$\begin{aligned} C_{\texttt{Chamfer}}(\mathcal{P}', \mathcal{P}) = \frac{1}{n'} \sum_{\boldsymbol{p}' \in \mathcal{P}'} \min_{\boldsymbol{p} \in \mathcal{P}} \|\boldsymbol{p}' - \boldsymbol{p}\|_2^2 \\ + \frac{1}{n} \sum_{\boldsymbol{p} \in \mathcal{P}} \min_{\boldsymbol{p}' \in \mathcal{P}'} \|\boldsymbol{p} - \boldsymbol{p}'\|_2^2, \end{aligned} \tag{2}$$

which is symmetric w.r.t. $\mathcal{P}$ and $\mathcal{P}'$. Although the Chamfer distance (2) is not a strict distance metric, since the triangle inequality does not hold, it is popularly used in the recent literature of learning based 3D shape generation [36], [43], [44]. It measures the distance between the two point sets by averaging over the individual deviations of any $\boldsymbol{p} \in \mathcal{P}$ from $\mathcal{P}'$ and those of any $\boldsymbol{p}' \in \mathcal{P}'$ from $\mathcal{P}$. We note that Chamfer distance is less effective in prevention of outlier points when generating $\mathcal{P}'$ from $\mathcal{P}$, since a small portion of outliers in $\mathcal{P}'$ increases the distance (2) negligibly. This shortcoming of Chamfer distance motivates us to additionally use the following Hausdorff distance.

Hausdorff Distance – For the point sets $\mathcal{P}$ and $\mathcal{P}'$, we consider in this work a non-symmetric Hausdorff distance that concerns with the resulting $\mathcal{P}'$ only, which computes

$$C_{\texttt{Hausdorff}}(\mathcal{P}', \mathcal{P}) = \max_{\boldsymbol{p}' \in \mathcal{P}'} \min_{\boldsymbol{p} \in \mathcal{P}} \|\boldsymbol{p}' - \boldsymbol{p}\|_2^2. \tag{3}$$

As (3) indicates, the Hausdorff distance finds the largest one among the smallest distances of individual $\boldsymbol{p}' \in \mathcal{P}'$ from $\mathcal{P}$. It is thus sensitive in case that outliers are generated in $\mathcal{P}'$.

Computation of distances (2) and (3) involves individual points contained in $\mathcal{P}$ and $\mathcal{P}'$, but not the local surface geometries centered on them; consequently, it is possible that less smooth surface change, including generation of spiky points, would visibly appear in the resulting $\mathcal{P}'$, even though $\mathcal{P}'$ could be close to $\mathcal{P}$ when measured by (2) and/or (3), causing failure to achieve the imperceptible objective of adversarial modification. We introduce the following objective to reduce the less smooth surface modification.

Consistency of Local Curvatures – Our way to achieve imperceptible modification of adversarial point cloud can be generally described as ensuring the *local consistency of curvatures* between the surface of $\mathcal{P}$ and that of $\mathcal{P}'$, where local consistency means that for a spatially closest pair of surface points respectively from $\mathcal{P}$ and $\mathcal{P}'$, their magnitudes of curvature are similar. Since computations in this work are conducted on the discrete point clouds, we rely on the following discrete notions of point-wise curvature.

For any point $\boldsymbol{p}' \in \mathcal{P}'$, we find its closest point $\boldsymbol{p} \in \mathcal{P}$ by $\boldsymbol{p} = \arg\min_{\boldsymbol{p} \in \mathcal{P}} \|\boldsymbol{p}' - \boldsymbol{p}\|_2$. There exist local point neighborhoods $\mathcal{N}'_{\boldsymbol{p}'} \subset \mathcal{P}'$ and $\mathcal{N}_{\boldsymbol{p}} \subset \mathcal{P}$ respectively associated with $\boldsymbol{p}'$ and $\boldsymbol{p}$, which are obtained in this work by searching k nearest neighbors, suggesting $|\mathcal{N}'_{\boldsymbol{p}'}| = |\mathcal{N}_{\boldsymbol{p}}| = k$. To capture the local geometry of $\mathcal{N}_{\boldsymbol{p}}$, we rely on the following discrete notion

$$\kappa(\boldsymbol{p}; \mathcal{P}) = \frac{1}{k} \sum_{\boldsymbol{q} \in \mathcal{N}_{\boldsymbol{p}}} \left| \langle (\boldsymbol{q} - \boldsymbol{p}) / \|\boldsymbol{q} - \boldsymbol{p}\|_2, \boldsymbol{n}_{\boldsymbol{p}} \rangle \right|, \tag{4}$$

where $\boldsymbol{n}_{\boldsymbol{p}}$ denotes the unit normal vector of the surface at $\boldsymbol{p}$. We follow [45] and compute $\boldsymbol{n}_{\boldsymbol{p}}$ from $\mathcal{N}_{\boldsymbol{p}}$ as follows: we first generate a 3×3 positive semidefinite covariance matrix

$$\boldsymbol{C} = \sum_{\boldsymbol{q} \in \mathcal{N}_{\boldsymbol{p}}} (\boldsymbol{q} - \boldsymbol{p}) \otimes (\boldsymbol{q} - \boldsymbol{p}), \tag{5}$$

where $\otimes$ denotes the outer product operation; we then apply eigen-decomposition to $\boldsymbol{C}$, obtaining $\boldsymbol{n}_{\boldsymbol{p}}$ as the eigenvector corresponding to the smallest eigenvalue of $\boldsymbol{C}$; the first two eigenvectors define the surface tangent plane $\mathcal{T}(\mathcal{N}_{\boldsymbol{p}})$ at $\boldsymbol{p}$.

The term (4) intuitively measures the averaged angles between the normal vector and the vector defined by pointing $\boldsymbol{p}$ towards each $\boldsymbol{q}$ of its neighboring points. Indeed, since the normal vector $\boldsymbol{n}_{\boldsymbol{p}}$ is orthogonal to the tangent plane $\mathcal{T}(\mathcal{N}_{\boldsymbol{p}})$ of the surface at $\boldsymbol{p}$, each inner product in (4) characterizes how the normals vary directionally in the local neighborhood $\mathcal{N}_{\boldsymbol{p}}$, thus approximately measuring the local, directional curvature, and an average of $|\mathcal{N}_{\boldsymbol{p}}|$ inner products in (4) approximately measures the local, mean curvature. We compute $\kappa'(\boldsymbol{p}'; \mathcal{P}')$ in the same way as (4), with a subtle difference that instead of computing $\boldsymbol{n}'_{\boldsymbol{p}'}$ from $\mathcal{N}'_{\boldsymbol{p}'}$, we directly use $\boldsymbol{n}_{\boldsymbol{p}}$, *i.e.*, the unit normal vector of the point in $\mathcal{P}$ that is closest to $\boldsymbol{p}'$, as a surrogate of $\boldsymbol{n}'_{\boldsymbol{p}'}$, since normal vectors of $\mathcal{P}$ can be pre-computed and efficiently retrieved during the modification process.

Given $\kappa'(\boldsymbol{p}'; \mathcal{P}', \mathcal{P})$ and $\kappa(\boldsymbol{p}; \mathcal{P})$, we use the following objective to encourage the consistency of local geometries between any $\boldsymbol{p}' \in \mathcal{P}'$ and its closest point $\boldsymbol{p} \in \mathcal{P}$

$$\begin{aligned} C_{\texttt{Curvature}}(\mathcal{P}', \mathcal{P}) = \frac{1}{n'} \sum_{\boldsymbol{p}' \in \mathcal{P}'} \|\kappa'(\boldsymbol{p}'; \mathcal{P}', \mathcal{P}) - \kappa(\boldsymbol{p}; \mathcal{P})\|_2^2 \\ \text{s.t. } \boldsymbol{p} = \arg\min_{\boldsymbol{p} \in \mathcal{P}} \|\boldsymbol{p}' - \boldsymbol{p}\|_2, \end{aligned} \tag{6}$$

where we write $\kappa'(\boldsymbol{p}'; \mathcal{P}', \mathcal{P})$ since the normal vector involved in its computation is from the corresponding one of $\mathcal{P}$. Note that terms similar to (4) are also used in [39], [40] for single-view surface reconstruction. Our use of the term (4) in (6) is to encourage the consistency of local surface geometries between $\mathcal{P}'$ and $\mathcal{P}$, rather than to directly minimize (4) as in [39], [40].

The Combined Geometry-Aware Objective – We use the following combined objective to learn to perturb individual points of $\mathcal{P}$ to obtain $\mathcal{P}'$

$$C_{Geometry}(\mathcal{P}',\mathcal{P}) = C_{\texttt{Chamfer}}(\mathcal{P}', \mathcal{P}) + \lambda_1 \cdot C_{\texttt{Hausdorff}}(\mathcal{P}', \mathcal{P}) + \lambda_2 \cdot C_{\texttt{Curvature}}(\mathcal{P}', \mathcal{P}), \quad (7)$$

where λ_1 and λ_2 are penalty parameters whose default values are set as $\lambda_1 = 0.1$ and $\lambda_2 = 1$, which work well in all our experiments; a sensitivity analysis on how their settings affect empirical performance is also presented in Section 5. The combined objective $C_{Geometry}(\mathcal{P}', \mathcal{P})$ (7), shortened as $C_{Geo}(\mathcal{P}', \mathcal{P})$, will be used a regularizer to penalize a misclassification loss, as specified in (10).

2.3 Evaluation of Geometric Regularity for Point Cloud Representation of Object Surface

For a given point cloud $\mathcal{P}$ representing an object surface, it is in general difficult to measure its geometric regularity in a quantitative manner. In this work, we are inspired by the desired surface property of *fairness* [28], and introduce the following discrete and approximate measure to quantify the regularity of $\mathcal{P}$

$$R(\mathcal{P}) = \max_{\boldsymbol{p} \in \mathcal{P}} \frac{1}{k} \sum_{\boldsymbol{q} \in \mathcal{N}_{\boldsymbol{p}}} D(\boldsymbol{q}, \mathcal{T}(\mathcal{N}_{\boldsymbol{p}})), \quad (8)$$

where $D(\cdot)$ computes the l_2-norm distance between any neighboring point $\boldsymbol{q} \in \mathcal{N}_{\boldsymbol{p}}$ and its projection onto the tangent plane $\mathcal{T}(\mathcal{N}_{\boldsymbol{p}})$ computed from the neighborhood $\mathcal{N}_{\boldsymbol{p}}$. Note that a surface is generally considered as fair if its curvatures are globally minimized. The proposed measure (8) is easily computable, and functions as an approximate surrogate of the globally maximum curvature defined on the discrete point cloud, since a surface with high values of local curvature gives a high value of $R(\mathcal{P})$. Thus, the lower value of $R(\mathcal{P})$ is, the more regular $\mathcal{P}$ is. The proposed $R(\mathcal{P})$ is also relevant to the measure of perceptual aesthetics [46], [47], since both of them are motivating from the same desired surface property of fairness. In Section 5, we show that (8) is quite relevant in capturing regularities of adversarial point clouds, by collecting human preference over those generated by different methods from Amazon Mechanical Turk.

3 Generation of Adversarial Point Clouds

Assume a model $f : \mathcal{X} \rightarrow \mathcal{Y}$ that classifies benign signals. Generation of adversarial signals aims to obtain from a benign $\boldsymbol{x}$ a crafted malicious $\boldsymbol{x}'$ such that $\boldsymbol{x}'$ would be misclassified by the model $f(\cdot)$, where the modification from $\boldsymbol{x}$ to $\boldsymbol{x}'$ is expected to be imperceptible to our humans. In the literature of adversarial 2D image generation, there exist generally two settings respectively termed as untargeted attack [18], [48], [49] and targeted attack [8], [32], [50]. Assume the true label of $\boldsymbol{x}$ be $y \in \mathcal{Y}$, and $\boldsymbol{x}$ is correctly classified by $f(\cdot)$, i.e., $f(\boldsymbol{x}) = y$. Untargeted attack generates an adversarial signal $\boldsymbol{x}'$ such that $f(\boldsymbol{x}') \neq y$. Targeted attack generates $\boldsymbol{x}'$ such that $f(\boldsymbol{x}') = \hat{y}$, with $\hat{y} \in \mathcal{Y}$ but $\hat{y} \neq y$. It is obvious that targeted attack is a more involved task setting than the untargeted one, by crafting specified malicious signals. In this work, we focus on the setting of targeted attack for point cloud data, which generates $\mathcal{P}'$ such that $\mathcal{P}'$ is classified as a specified class $\hat{y} \neq y$.

Technically, we adapt the state-of-the-art framework of C&W attack [19] to achieve the goal. Let the classification model $f(\cdot)$ be realized by a function $\boldsymbol{g} : \mathcal{X} \rightarrow \mathbb{R}^{|\mathcal{Y}|}$, and we have $f(\boldsymbol{x}) = \arg\max_{i \in \mathcal{Y}} g_i(\boldsymbol{x})$, where $g_i(\cdot)$ takes the i^{th} element of $\boldsymbol{g}(\cdot)$. When implementing $f(\cdot)$ as a deep classification network, $\boldsymbol{g}(\cdot)$ outputs the network logits, *i.e.*, output of the network before the final softmax. C&W attack commonly uses a margin based loss for the misclassification term in the general adversarial objective (1), which gives $C_{Mis}(\boldsymbol{x}') = \max\{\max_{i \neq \hat{y}} g_i(\boldsymbol{x}') - g_{\hat{y}}(\boldsymbol{x}'), 0\}$. This margin based $C_{Mis}(\boldsymbol{x}')$ would stop pursuing more malicious signals once $\max_{i \neq \hat{y}} g_i(\boldsymbol{x}') - g_{\hat{y}}(\boldsymbol{x}') \leq 0$. In this work, we use the following misclassification loss to increase the malicious levels possibly achieved by targeted attack of point clouds

$$C_{Mis}(\mathcal{P}') = -\log\left(\exp(g_{\hat{y}}(\mathcal{P}')) / \sum_{i=1}^{|\mathcal{Y}|} \exp(g_i(\mathcal{P}'))\right), \quad (9)$$

where $\hat{y}$ is the specified class of targeted attack. Regularizing $C_{Mis}(\mathcal{P}')$ with our proposed geometry-aware regularizer (7) gives our proposed method of Geometry-Aware Adversarial Attack ($GeoA^3$) for learning adversarial point cloud $\mathcal{P}'$ from $\mathcal{P}$

$$\min_{\mathcal{P}'} C_{Adv}(\mathcal{P}', \mathcal{P}) = C_{Mis}(\mathcal{P}') + \beta \cdot C_{Geo}(\mathcal{P}', \mathcal{P}), \quad (10)$$

where β is a penalty parameter controlling the overall level of geometry-aware regularization, whose setting follows [19] and is automatically adjusted via binary search. Minimization of (10) can be simply achieved by Stochastic Gradient Descent (SGD) (or its variants [51]), which gives the following rule to update $\mathcal{P}'_{t+1}$ from $\mathcal{P}'_t$

$$\mathcal{P}'_{t+1} \leftarrow \mathcal{P}'_t - \eta \cdot \nabla C_{Adv}(\mathcal{P}'_t, \mathcal{P}), \quad (11)$$

where η is the learning rate. We note that without our proposed (7), the more aggressive misclassification loss (9) would produce $\mathcal{P}'$ whose modification from $\mathcal{P}$ could be clearly perceived. It is our combined use of (9) and geometry-aware regularizer (7) that enables pursuing more adversarial, arguably less defendable, point clouds without introducing noticeable modifications. In practice, we generate an adversarial $\mathcal{P}'$ by minimizing (10) to perturb individual points $\{\boldsymbol{p}_i\}_{i=1}^n$ contained in $\mathcal{P}$, which technically means an iterative optimization of adversarial point-wise coordinate offsets. We conduct thorough empirical studies in Section 5 to verify our proposed $GeoA^3$ objective.

The input $\mathcal{P}$ is obtained by sampling a discrete set of points from a certain object surface. Consequently, adversarial effects achieved by point perturbation may be attributed to coupled factors of *surface shape deformation* and *point cloud* *re-sampling*, since the resulting $\mathcal{P}'$ may represent a discrete sampling from a deformed surface shape, a re-sampling from the same original shape, or a mixture of them. We study this issue shortly in an attempt towards generation of adversarial surface shapes.

4 Towards Generation of Adversarial Surface Shapes

While our main focus of this paper is to learn to generate from a given $\mathcal{P}$ an adversarial $\mathcal{P}'$ w.r.t. a model $f(\cdot)$, which would practically cause safety-critical issues by using $\mathcal{P}'$ to guide a LiDAR spoofer [29] to fool the 3D sensor, it is more desirable to make the underlying surface $\mathcal{S}'$ represented by $\mathcal{P}'$ adversarial to the surface $\mathcal{S}$ represented by $\mathcal{P}$. Indeed, when $\mathcal{P}'$ only represents an adversarial sampling of the original $\mathcal{S}$, it would be easily defended by a benign re-sampling. Technically, this desirable objective is to obtain a $\mathcal{P}'$ such that when re-sampling a $\mathcal{Q}'$ from a surface reconstructed by $\mathcal{P}'$ (*e.g.*, via meshing from $\mathcal{P}'$), $\mathcal{Q}'$ still has a certain degree of adversarial effect to attack the model $f(\cdot)$.

To this end, we propose in this work a simple algorithm termed *Iterative Tangent Jittering (IterTanJit)*. When optimizing our proposed objective (10) via SGD, IterTanJit introduces per iteration point-wise jittering of 3D coordinates on the tangent planes associated with individual points contained in the intermediate update of the adversarial $\mathcal{P}'$. Specifically, for any $\boldsymbol{p}'_t \in \mathcal{P}'_t$ of the intermediate update at iteration t, we compute its associated tangent plane $\mathcal{T}(\mathcal{N}_{\boldsymbol{p}'_t})$ from the neighborhood $\mathcal{N}_{\boldsymbol{p}'_t}$. Note that $\mathcal{T}(\mathcal{N}_{\boldsymbol{p}'_t})$ is spanned by the two leading eigenvectors $\boldsymbol{v}_1(\mathcal{N}_{\boldsymbol{p}'_t})$ and $\boldsymbol{v}_2(\mathcal{N}_{\boldsymbol{p}'_t})$ of a covariance matrix constructed by (5), as described in Section 2.2. To have a jittering of $\boldsymbol{p}'_t$ on its tangent plane, we sample a scaling pair (s_1, s_2) from a Gaussian distribution with a standard deviation of σ, resulting in

$$\begin{aligned} \boldsymbol{j}'_t &= s_1 \cdot \boldsymbol{v}_1(\mathcal{N}_{\boldsymbol{p}'_t}) + s_2 \cdot \boldsymbol{v}_2(\mathcal{N}_{\boldsymbol{p}'_t}) \\ &\text{s.t. } \; s_1, s_2 \sim \mathrm{Normal}(0, \sigma). \end{aligned} \quad (12)$$

Denote $\{\boldsymbol{j}'_t\}$, corresponding to all the points of $\mathcal{P}'_t$, collectively as $\mathcal{J}'_t$, IterTanJit updates (10) by

$$\mathcal{P}'_{t+1} \leftarrow \mathcal{P}'_t - \eta \cdot \nabla C_{Adv}(\mathcal{P}'_t + \mathcal{J}'_t, \mathcal{P}). \quad (13)$$

Rationale of the proposed (13) is that at each iteration t, IterTanJit treats $\mathcal{P}'_t + \mathcal{J}'_t$ as a randomized (approximate) re-sampling from the underlying surface $\mathcal{S}'_t$ represented by $\mathcal{P}'_t$, such that the direction of adversarial update accounts more for surface deformation; the accumulated expectation by conducting (13) iteratively is to obtain a final $\mathcal{P}' = \mathcal{P}'_T$ whose adversarial effect comes more from a deformation of the input shape $\mathcal{S}$. To improve efficiency of the proposed IterTanJit, we only update the neighborhood $\mathcal{N}_{\boldsymbol{p}'_t}$ per $\boldsymbol{p}'_t$ every certain number of iterations.

Practical Implementation and Physical Attack - Practically, we employ screened Poisson surface reconstruction [52] to reconstruct a surface mesh $\mathcal{S}'$ from a given $\mathcal{P}'$, and use a strategy of farthest point sampling [5] to re-sample a $\mathcal{Q}'$ from the reconstructed $\mathcal{S}'$. For physical attack, we do 3D printing using $\mathcal{S}'$ and then scan the printed object to obtain $\mathcal{Q}'$. The respectively obtained $\mathcal{Q}'$ is used for evaluation of attack performance.

5 Experiments

Dataset – We use object instances from ModelNet40 [53] to evaluate our proposed methods. The dataset consists of $12,311$ CAD models belonging to 40 semantic categories. To have a point cloud, we follow [5] and sample points uniformly from each CAD model and then normalize them into a unit ball. We use the official data splits of ModelNet40 for point set classification [4], [5], which give $9,843$ instances to train classifiers that are to be attacked. For testing, we follow [11] and randomly select 25 instances for each of 10 object categories in the ModelNet40 testing set, which can be well classified by the classifiers of interest. The 10 object categories include *airplane, bed, bookshelf, bottle, chair, monitor, sofa, table, toilet,* and *vase*.

Evaluation Metrics – To quantitatively compare the adversarial results generated by different methods, we use both the measure of attack success rate and that of geometric regularity. The former measures, among the input, benign point clouds, the ratio of their adversarial results that successfully fool a classifier; note that all the benign point clouds are classified correctly by the classifier. We use the proposed $R(\mathcal{P}')$ (8) to measure the geometric regularity of any adversarial $\mathcal{P}'$; a geometrically regular $\mathcal{P}'$ gives a lower value of $R(\mathcal{P}')$. A better-performing method is expected to generate adversarial results that achieve higher attack success rates and lower values of $R(\mathcal{P}')$, simultaneously. We report attack success rates under a state-of-the-art defense method of Statistical Outlier Removal (SOR) [27], which works by statistically dropping certain points from any adversarial $\mathcal{P}'$; more specifically, if the distance between a point and its neighbors are much larger than such distance averaged over the whole $\mathcal{P}'$, this point will be considered as an outlier and will be dropped by SOR. Geometric regularity is reported by averaging $R(\mathcal{P}')$ over adversarial results of all instances. We also conduct subjective studies for a qualitative evaluation by collecting human preferences from Amazon Mechanical Turk.

Models and Implementation Details – We use PointNet [4], PointNet++ [5], and DGCNN [6] as our classifiers of interest, against which comparative methods fire attacks. Without mentioning otherwise, all the methods follow a white-box, targeted attack protocol via point perturbation. We use Adam [51] to optimize the objective (10) of our proposed $GeoA^3$, where the learning rate and momentum are respectively set as 0.01 and 0.9. We set $k = 16$ to define local point neighborhoods. We use the default values of $\lambda_1 = 0.1$ and $\lambda_2 = 1.0$ for penalty parameters in (7). The penalty β in (10) is initialized as $2,500$ and automatically adjusted via binary search, which follows [19].

5.1 Ablation Studies and Sensitivity Analysis

To test the efficacy of our proposed $GeoA^3$ (10), we first conduct ablation studies by removing individual terms in the geometry-aware regularizer (7) and compare the results quantitatively and qualitatively. Table 1 shows that across a range of dropping ratios via the state-of-the-art defense SOR [27], attack success rates drop by removing any one of the three terms in (7), with the term of Hausdorff distance playing the most important role and that of consistency of

TABLE 1: Ablation studies on our proposed $GeoA^3$ (10). Each input, benign point cloud contains $1,024$ points. Performance is measured in terms of both the attack success rate (%) and geometric regularity $R(\mathcal{P}')$ (8). Attack success rates are reported by dropping a range of ratios of points from their adversarial results using the state-of-the-art SOR defense method [27]. All experiments are conduced using PointNet [4] as the model of classifier.

Method	Attack success rate (%) defense by dropping different ratios of points via SOR [27]							Geometric regularity $R(\mathcal{P}')$
	0%	1%	2%	5%	10%	15%	20%	
$GeoA^3$ with all three geometry-aware terms	100	**83.47**	**70.56**	**52.61**	**31.58**	**18.62**	**11.71**	**0.0968**
w/o Chamfer Distance	100	82.17	68.14	50.08	30.24	15.14	10.01	0.1020
w/o Hausdorff Distance	100	24.51	18.89	11.68	6.60	3.97	3.12	0.1588
w/o Consistency of Local Curvatures	100	53.85	35.90	14.88	5.57	2.81	2.00	0.1055

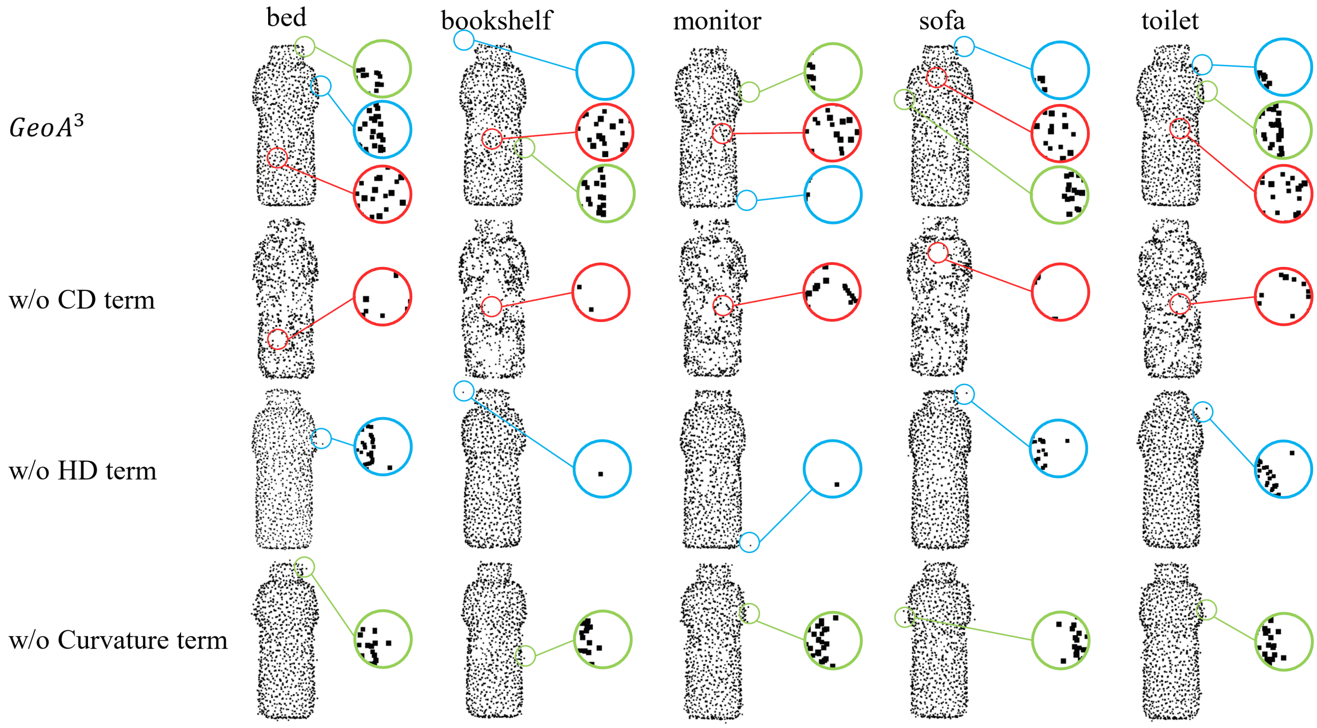


Fig. 2: Example results of ablation study by removing individual terms from our proposed geometry-aware regularizer (7), where CD stands for the term of Chamfer distance, HD for that of Hausdorff distance, and Curvature for that of consistency of local curvatures. Each input, benign point cloud contains $1,024$ points, to which adversarial attacks are applied. Results are from a *bottle* instance that is attacked against PointNet [4], targeting at the categories of *bed*, *bookshelf*, *monitor*, *sofa*, and *toilet*. All the shown examples conduct the attacks successfully. Lens of different colors are used to highlight the differences among the comparative methods. Comparative results from instances of other categories are of similar quality.

TABLE 2: Adversarial results when respectively sampling different numbers of points from each CAD model as the working point clouds. Performance is measured in terms of both the attack success rate (%) and geometric regularity $R(\mathcal{P}')$ (8). Attack success rates are reported by dropping a range of ratios of points from their adversarial results using the state-of-the-art SOR defense method [27]. All experiments are conduced using PointNet [4] as the model of classifier.

Number of points	Attack success rate (%) defense by dropping different ratios of points via SOR [27]							Geometric regularity $R(\mathcal{P}')$
	0%	1%	2%	5%	10%	15%	20%	
$1,024$	100	83.47	70.56	52.61	31.58	18.62	11.71	0.0968
$2,048$	100	81.02	70.44	48.36	28.04	16.58	10.62	0.0988
$4,096$	100	80.80	71.11	47.73	22.31	18.04	9.87	0.0997

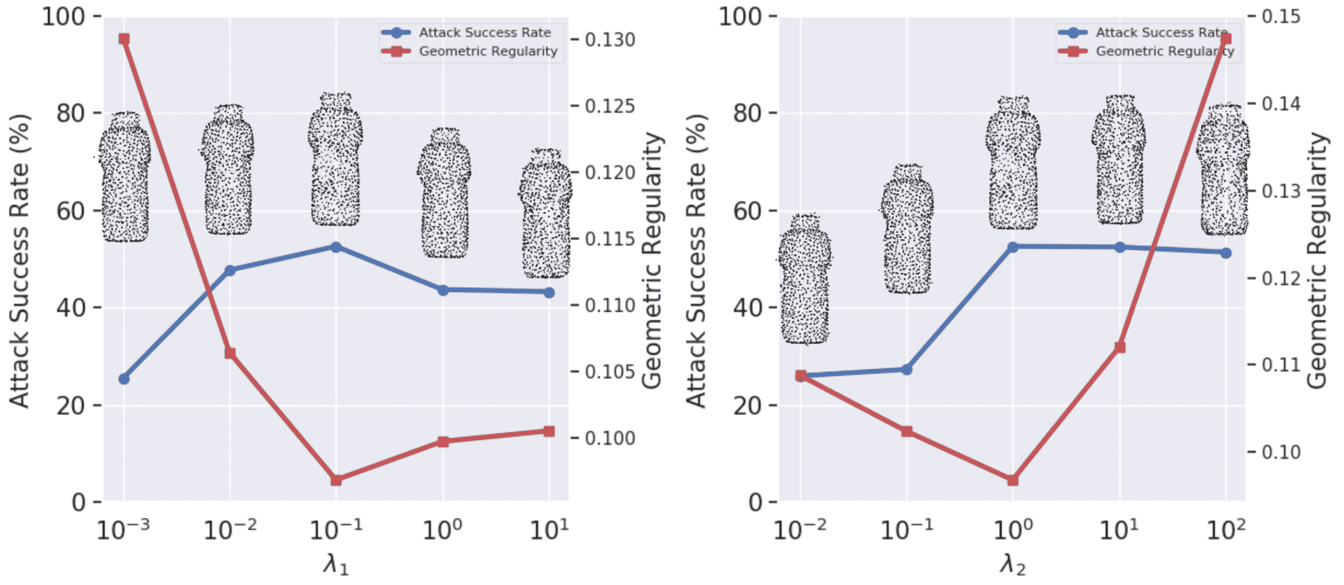


Fig. 3: Sensitivity analysis of our method w.r.t. the values of λ_1 and λ_2 in the geometry-aware regularizer (7). We plot both the attack success rate (%), under the SOR defense [27] of 5% point dropping, and geometric regularity (8) by varying the values of either λ_1 (left) or λ_2 (right) around their respective default values of $\lambda_1 = 0.1$ and $\lambda_2 = 1$. Experiments are conducted on point clouds of $1,024$ points using PointNet [4] as the model of classifier. For a better comparison, an example result from a *bottle* instance is also accompanied for each setting of λ_1 and λ_2 values.

TABLE 3: Results of our method when attacking different models of point set classifiers including PointNet [4], PointNet++ [5], and DGCNN [6]. Each input, benign point cloud contains $1,024$ points. Performance is measured in terms of both the attack success rate (%), under different ratios of point dropping via the state-of-the-art SOR defense [27], and geometric regularity $R(\mathcal{P}')$ (8).

Model	Attack success rate (%) defense by dropping different ratios of points via SOR [27]							Geometric regularity $R(\mathcal{P}')$
	0%	1%	2%	5%	10%	15%	20%	
PointNet [4]	100	83.47	70.56	52.61	31.58	18.62	11.71	0.0968
PointNet++ [5]	100	72.09	56.46	31.73	14.57	7.48	3.78	0.1037
DGCNN [6]	100	64.28	50.12	26.47	10.08	2.99	2.67	0.1368

local curvatures following. The observations are consistent for the measure of geometric regularity. To qualitatively understand how the three terms play roles for generating visually less perceptible adversarial point clouds, we show in Figure 2 example results, which tells that removing any of three terms will produce adversarial results containing either less regular surfaces or clearly visible outliers. Results from instances of other categories are of similar quality. These comparisons confirm the combined advantage of our proposed geometry-aware regularizer (7) for generating adversarial point clouds that are both less defendable and less perceptible, simultaneously.

While most of the experiments are conducted on point clouds of $1,024$ points, it is interesting to investigate whether the adversarial effects vary w.r.t. different numbers of points per instance. We conduct such experiments by respectively sampling $1,024$, $2,048$, and $4,096$ points from each CAD model as our working point clouds. We correspondingly set their sizes of local neighborhoods as $k = 16$, 32, and 64. Table 2 shows that under both the measures of attack success rate and geometric regularity, our results are relatively stable w.r.t. different numbers of points. We choose point clouds of $1,024$ points as the default setting.

To investigate how the penalty parameters λ_1 and λ_2 in the geometry-aware regularizer (7) affect adversarial efficacy, we conduct experiments by varying each of them around their default values, while keeping the other one fixed. Plottings in Figure 3 show that attack success rates are relatively stable w.r.t. different values of λ_1 and λ_2, and our default values of $\lambda_1 = 0.1$ and $\lambda_2 = 1$ give the optimal results in terms of geometric regularity.

We finally investigate the efficacy of our proposed $GeoA^3$ to attack different models of point set classifiers. We use the representative PointNet++ [5] and DGCNN [6] whose classification performance on the benchmark ModelNet40 [53] is better than that of PointNet [4]. Table 3 shows that the attacking performance declines with the improved design of point set classifiers, and the corresponding

TABLE 4: Comparative results of different methods for adversarial point clouds. Each input, benign point cloud contains $1,024$ points. Performance is measured in terms of both the attack success rate (%) and geometric regularity $R(\mathcal{P}')$ (8). For different methods, attack success rates are reported by dropping a range of ratios of points from their adversarial results using the state-of-the-art SOR defense method [27]. All experiments are conduced using PointNet [4] as the model of classifier.

Method	Attack success rate (%)							Geometric regularity $R(\mathcal{P}')$
	defense by dropping different ratios of points via SOR [27]							
	0%	1%	2%	5%	10%	15%	20%	
Xiang's method [11]	100	1.42	1.24	0.98	0.80	0.67	0.49	0.1772
Liu's method [21]	100	0.63	0.40	0.36	0.09	0.09	0.04	0.1658
Tsai's method [35]	100	23.84	17.74	14.50	11.12	9.01	8.12	0.1602
Degenerate $GeoA^3$	100	3.21	1.60	1.02	0.62	0.36	0.31	0.1956
$GeoA^3$	100	**83.47**	**70.56**	**52.61**	**31.58**	**18.62**	**11.71**	**0.0968**

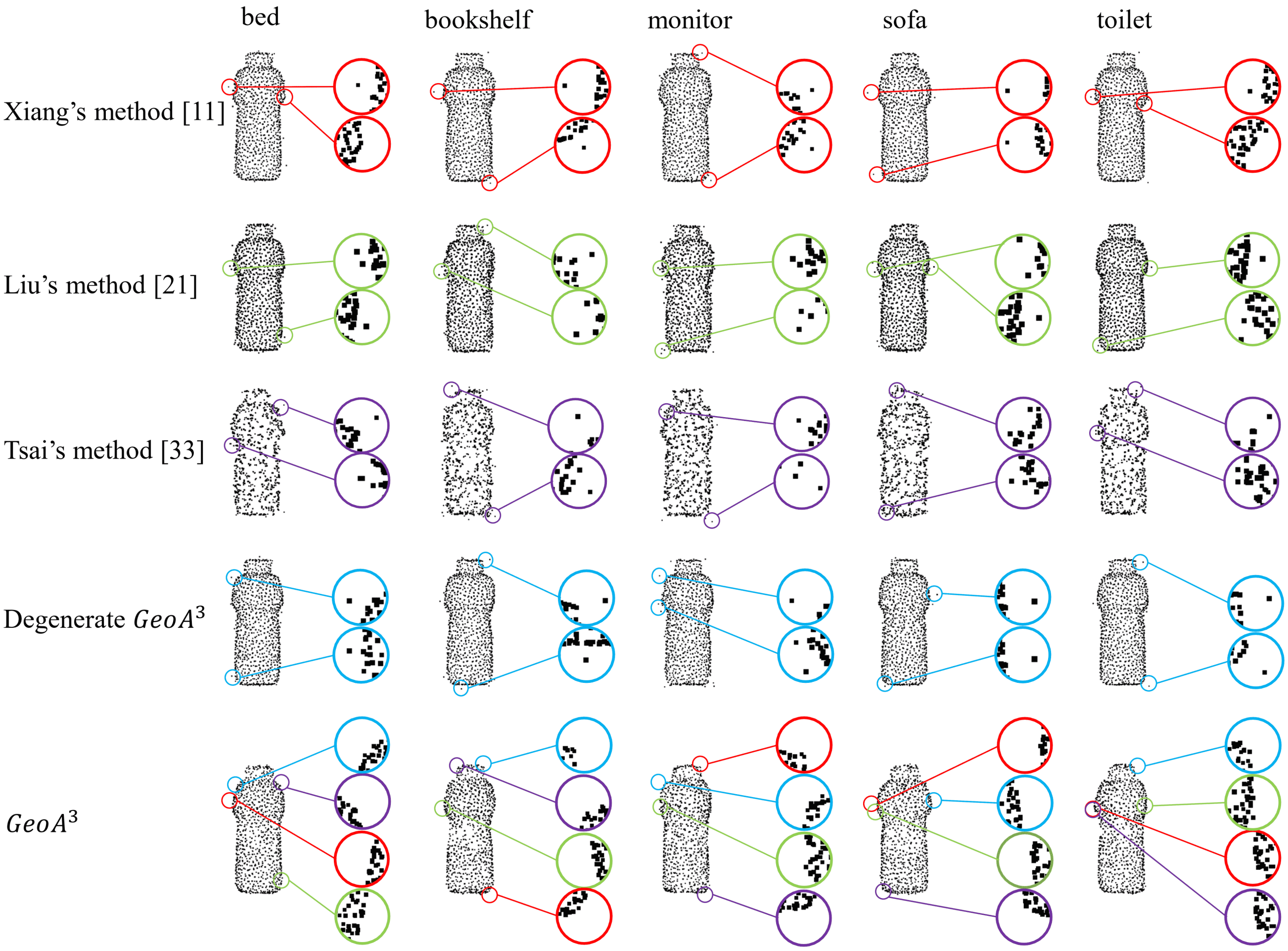
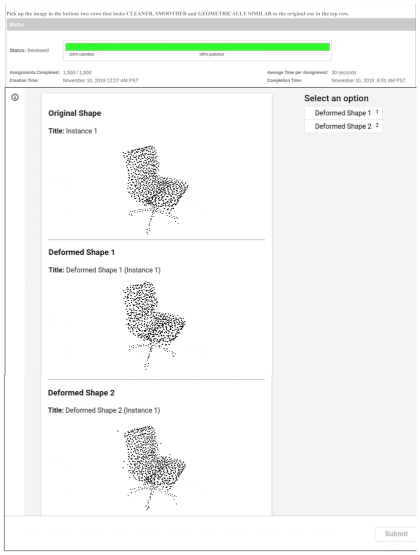


Fig. 4: Qualitative comparisons among adversarial results generated by different methods. Each point cloud contains $1,024$ points. Results are from a *bottle* instance that is attacked against PointNet [4], targeting at the categories of *bed*, *bookshelf*, *monitor*, *sofa*, and *toilet*. All the shown example results can successfully make the targeted attacks. The respective competing methods are compared with our $GeoA^3$ by highlighting the local differences using lens of varying colors. Results of different methods from other instances and targeting categories are of similar comparative quality.



Pick up the image in the bottom two rows that looks CLEANER, SMOOTHER and GEOMETRICALLY SIMILAR to the original one in the top row.

Status

Status: Reviewed

100% submitted 100% published

Assignments Completed: 1,500 / 1,500

Creation Time: November 10, 2019 12:27 AM PST

Average Time per Assignment: 30 seconds

Completion Time: November 10, 2019 6:31 AM PST

Original Shape

Title: Instance 1

Deformed Shape 1

Title: Deformed Shape 1 (Instance 1)

Deformed Shape 2

Title: Deformed Shape 2 (Instance 1)

Fig. 5: An interface we use to conduct subjective user studies by uploading onto Amazon Mechanical Turk a triple of point cloud snapshots including the benign one, the adversarial one generated by Xiang's method [11], and the adversarial one generated by our $GeoA^3$. All the uploaded results attack PointNet [4] successively.

geometric regularities drop as well. Such a phenomenon suggests that more effective attacking methods need to be designed specially to attack advanced models of point set classifiers, which is different from adversarial attacking of 2D images [9] where advanced image classifiers tend to be more vulnerable to adversarial examples. In the subsequent experiments, we use PointNet as the model of classifier to compare with existing methods for generation of adversarial point clouds.

5.2 Comparative Results of Adversarial Point Clouds

In this section, we compare our proposed $GeoA^3$ with existing methods [11], [21], [35] that generate adversarial point clouds under the setting of white-box, targeted attack. In Xiang's method [11], adversarial point perturbation follows the framework of C&W attack [19], by using a margin-based misclassification loss regularized by point-wise l_2-norms constraining the magnitudes of point perturbation. Liu's method [21] adopts a variant of the basic iterative method in [41], under a constraint of l_2-norm distance between the entire, benign point cloud and the adversarial one. Tsai's method [35] also follows the framework of C&W attack, whose constraint combines a global Chamfer distance and a local term that encourages the compactness of local neighborhoods in the generated adversarial point cloud. Results of these methods are obtained either by using their released codes, when available, or by reproducing their methods; in both cases, we tune their respective hyper-parameters as the optimal ones. In addition to existing methods, we also compare with a degenerate version of our method, dubbed Degenerate $GeoA^3$, which replaces our proposed geometry-aware (7) with constraints used in existing methods, *i.e.*, point-wise l_2-norm, while keeping the use of our more aggressive misclassification loss (9). Comparing with Degenerate $GeoA^3$ thus further highlights the importance of our proposed (7) for generation of adversarial point clouds. All the comparative experiments are conducted using PointNet [4] as the model of classifier.

In Table 4, we compare different methods under the measures of attack success rate, again under the state-of-the-art SOR defense, and geometric regularity $R(\mathcal{P}')$ (8). Under both of the two measures, our proposed $GeoA^3$ performs much better, across a range of dropping ratios via SOR, than both the existing methods and Degenerate $GeoA^3$ do. The comparisons confirm the combined efficacy of using our more aggressive misclassification loss (9) and geometry-aware regularizer (7) to simultaneously achieve the strongest adversarial attacking and least visual perception of the perturbations. The later advantage of our method is illustrated in Figure 4 where we show adversarial results of a *bottle* instance that is attacked, targeting at 5 different categories, by different methods (the same instance and targeting categories as in Figure 2). Clearly, $GeoA^3$ gives adversarial results whose visual differences from the benign ones can arguably be least perceived by humans; results from competing methods contain either point outliers and/or local geometric irregularities.

Subjective Evaluation - A subjective user study is implemented on Amazon Mechanical Turk (AMT) in order to evaluate visually imperceptible quality of adversarial results. Specifically, for each object instance, we upload onto AMT a triple of point cloud snapshots including the benign one, the adversarial one generated by Xiang's method [11], and the adversarial one generated by our $GeoA^3$; all the uploaded adversarial results attack the PointNet classifier successively. Online participants are asked to compare to discriminate which one of the two adversarial point clouds are visually more similar to the benign one. To implement the subjective comparison properly, we do the following: the order of showing the two results is randomized, each participant is limited to contribute at most 30 trials, and each adversarial result can be shown up to 50 different participants. In total, we collect $1,500$ trials from 128 participants; 82.06% of the trials consider adversarial results from our $GeoA^3$ visually closer to the benign ones, when compared against those from Xiang's method [11]. An AMT interface for our conducted subjective study is shown in Figure 5.

We finally show in Figure 6 more example results of adversarial point clouds from our method in an adversarial attacking matrix, where an example instance of each category is attacked against PointNet [4] targeting at all the other 9 categories.

5.3 Evaluation of Adversarial Surface Shapes and Physical Attacks

As discussed in Section 4, it could be practically less defendable when adversarial effect of a generated point cloud $\mathcal{P}'$ comes more from a deformation of surface shape, which essentially defines a much challenging task of surface-level adversarial attack. In Section 4, we also propose a simple optimizing algorithm of IterTanJit towards generation of adversarial surface shapes. To evaluate its efficacy and compare with the existing Tsai's method [35], we follow the

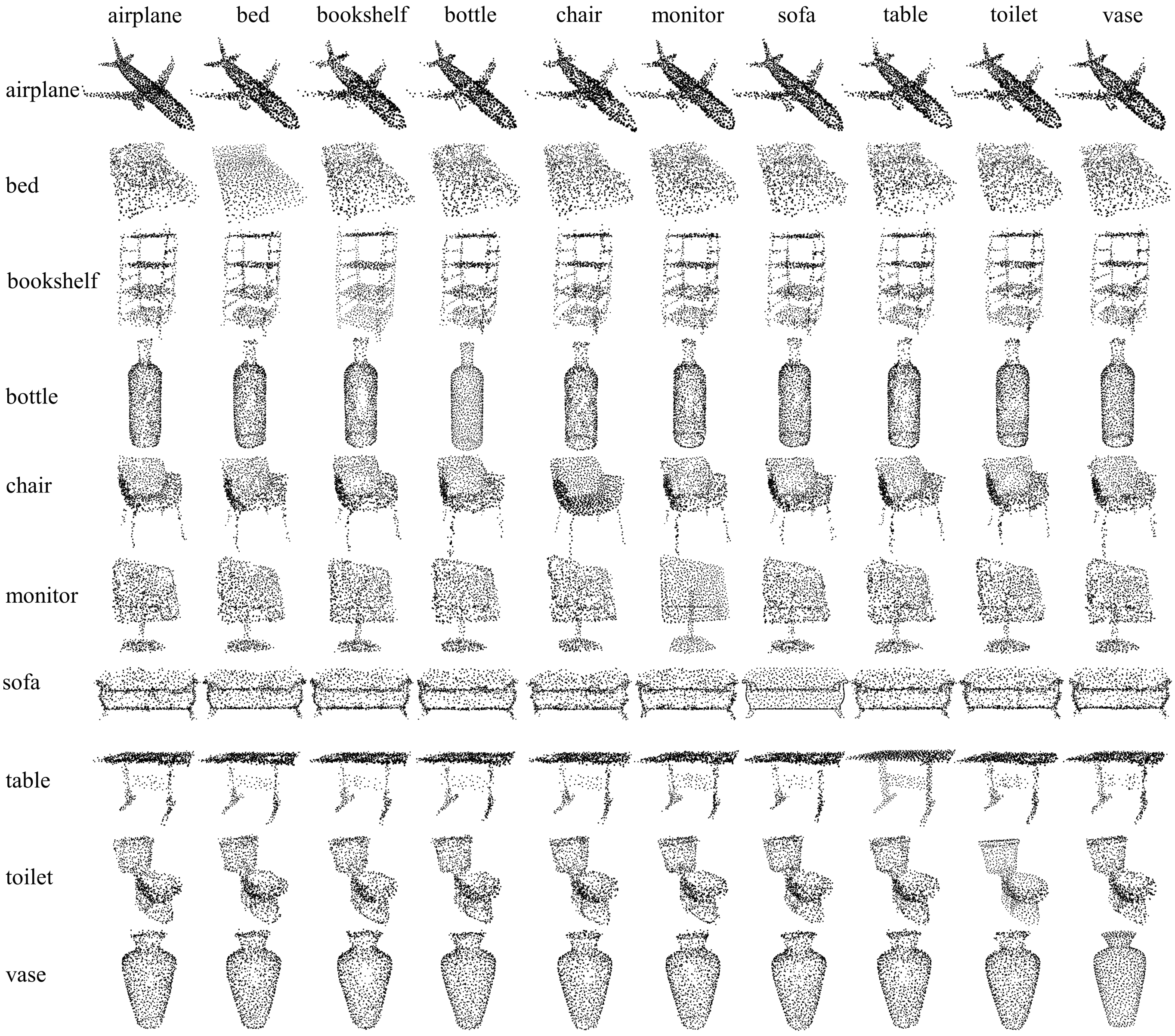


Fig. 6: Adversarial examples from our proposed $GeoA^3$. Results are organized in a matrix form where each off-diagonal entry presents the adversarial point cloud obtained by attacking an example instance from a certain category against PointNet [4], targeting at one of the remaining 9 categories, and the diagonal entry presents the input, benign point cloud.

TABLE 5: Evaluation of surface-level adversarial effects for results generated by different methods. For each adversarial result $\mathcal{P}'$, we use screened Poisson surface reconstruction [52] to get its mesh $\mathcal{S}'$, from which farthest point sampling [5] is used to re-sample a $\mathcal{Q}'$. Performance is measured in terms of both the attack success rate (%) and geometric regularity (8) on $\mathcal{P}'$ and $\mathcal{Q}'$. All experiments are conduced using PointNet [4] as the model of classifier.

Method	Attack success rate (%) on $\mathcal{P}'$ / Geometric regularity $R(\mathcal{P}')$	Attack success rate (%) on $\mathcal{Q}'$ after meshing and re-sampling / Geometric regularity $R(\mathcal{Q}')$										
		avg.	airplane	bed	bookshelf	bottle	chair	monitor	sofa	table	toilet	vase
Tsai's method [35]	100/0.1605	32.6/0.0865	0.0	16.0	16.0	86.0	8.0	16.0	4.0	92.0	4.0	84.0
$GeoA^3$	100/**0.0852**	14.0/**0.0791**	4.0	16.0	8.0	16.0	4.0	12.0	4.0	4.0	12.0	60.0
$GeoA^3$-IterTanJit	100/0.1024	**34.4**/0.0829	0.0	20.0	12.0	72.0	28.0	12.0	12.0	92.0	12.0	84.0

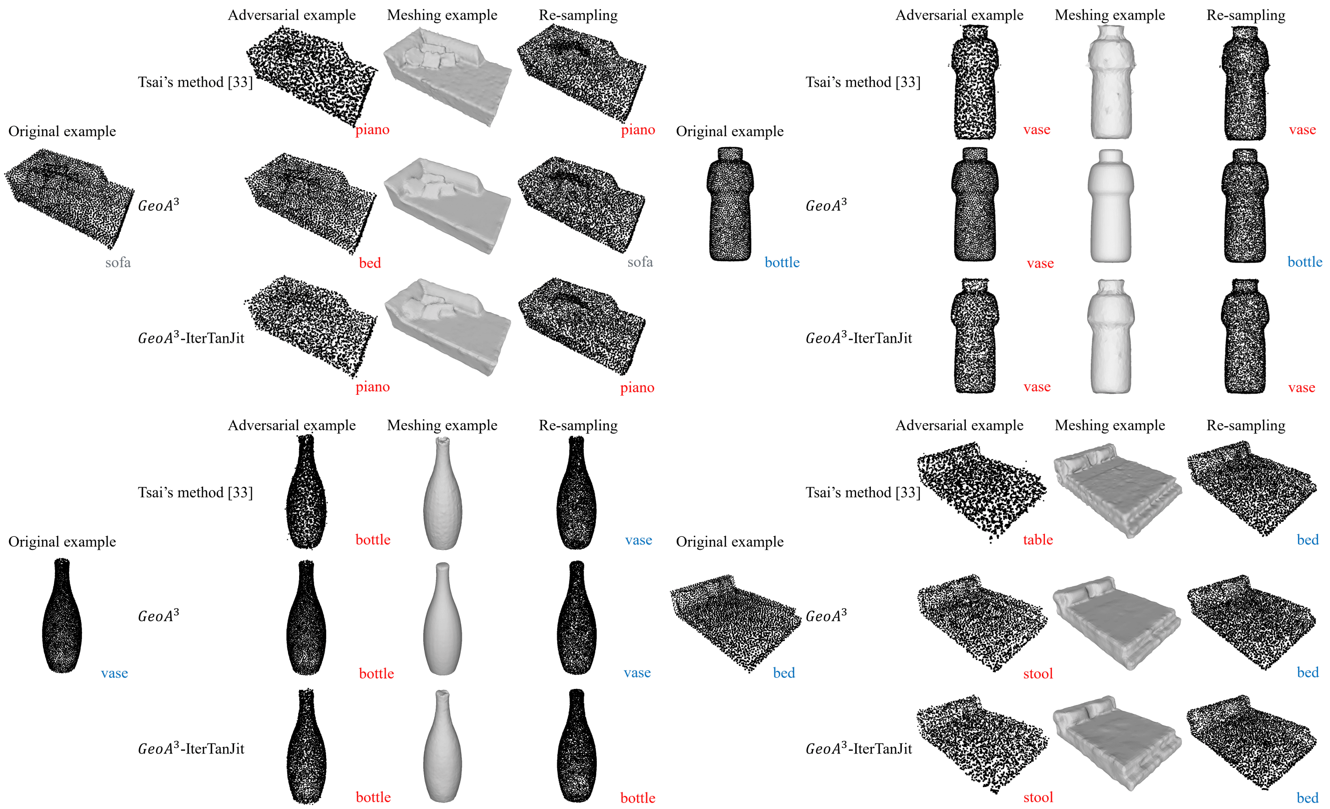


Fig. 7: Example results of adversarial point clouds generated by different methods, and their corresponding meshing and re-sampling results. Accompanying each point cloud is the category predicted by the PointNet [4] model of classifier; a red category indicates a successful attack.

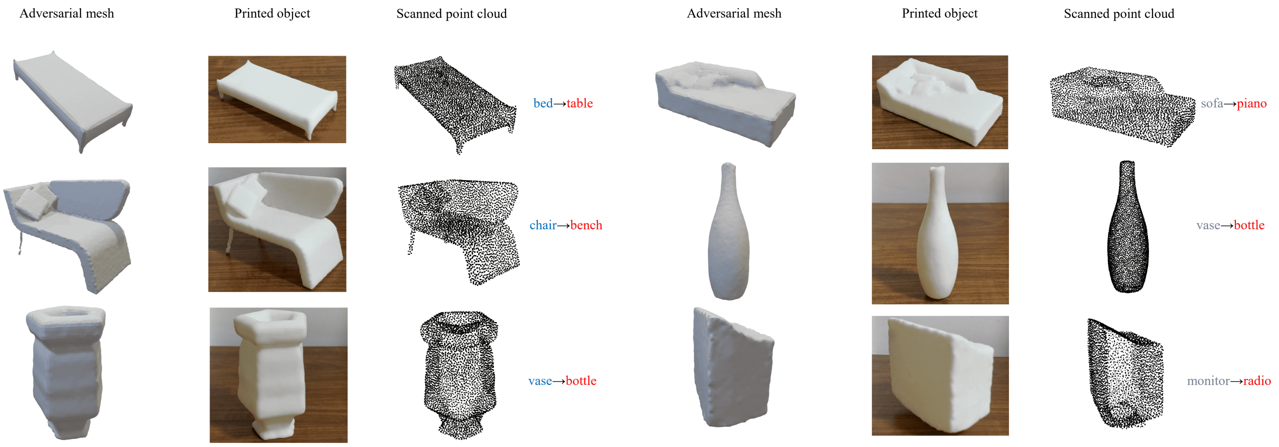


Fig. 8: Example meshes, their printed objects, and scanned point clouds. These results are obtained by first 3D-printing the mesh $\mathcal{S}'$ constructed from an adversarial $\mathcal{P}'$ generated by our proposed $GeoA^3$-IterTanJit, and then scanning the printed object to have the point cloud $\mathcal{Q}'$. UnionTech Lite600HD 3D printer (materials of 9400 resin) is used for this purpose. Accompanying each example is the category predicted by the PointNet [4] model of classifier; a blue category indicates the original label and a red category indicates a successful attack.

relatively easier setting of untargeted attack used in [35]. Our objective of untargeted attack is a variant of (10), by modifying its misclassification loss term (9) as

$$C_{Mis}(\mathcal{P}') = \log\left(\exp(g_y(\mathcal{P}'))/\sum_{i=1}^{|\mathcal{Y}|}\exp(g_i(\mathcal{P}'))\right),$$

where y is the ground-truth label of input point cloud. We randomly sample 25 instances from each of the considered 10 object categories as the working CAD models. For each CAD model, we uniformly sample $4,096$ points to have the input, benign $\mathcal{P}$, whose adversarial result $\mathcal{P}'$ is obtained by point-wise perturbation through optimization of the above modified version of (10) using IterTanJit. We set $\sigma = 0.02$ in (12) to sample point-wise jittering. After obtaining an adversarial $\mathcal{P}'$, we use screened Poisson surface reconstruction [52] to get its mesh $\mathcal{S}'$, from which we either use farthest point sampling [5] to re-sample a point cloud $\mathcal{Q}'$, or do 3D printing using $\mathcal{S}'$ and then scan the printed object to obtain $\mathcal{Q}'$. The respectively obtained $\mathcal{Q}'$ is used for evaluation of attacking performance.

Table 5 compares our methods with the only existing method [35] that aims for better surface-level adversarial effects when generating adversarial point clouds. For any $\mathcal{P}'$ generated by Tsai's method [35], we do the same procedure of meshing and re-sampling to have its re-sampled $\mathcal{Q}'$. While directly optimizing the untargeted attack variant of (10) using SGD (i.e., $GeoA^3$ in Table 5) performs poorly in terms of measuring attack success rate on $\mathcal{Q}'$, optimization via our proposed algorithm of IterTanJit (i.e., $GeoA^3$-IterTanJit in Table 5) greatly improves the attacking performance and outperforms Tsai's method as well. Note that our better attacking performance is achieved at a less violation of the imperceptibility criterion measured by both of the geometric regularities $R(\mathcal{P}')$ and $R(\mathcal{Q}')$. Figure 7 gives example results of the adversarial $\mathcal{P}'$, meshing $\mathcal{S}'$, and re-sampled $\mathcal{Q}'$ respectively from different methods, where one may perceive less perturbations in the results generated by our methods. Results in Table 5 suggest that achieving surface-level attacks via generation of adversarial point clouds is indeed a challenging task, and our contributed algorithm takes only a small step towards the desired goal.

Among the 250 testing instances used in this section, we do 3D printing for 15 of them which are among the successful attacking cases reported in Table 5. We do 3D printing using their respective $\mathcal{S}'$, and then scan each printed object to have the corresponding $\mathcal{Q}'$, which is used for evaluation of adversarial attack. 10 out of the 15 instances can still attack the PointNet classifier successfully after 3D printing and scanning, showing that our proposed $GeoA^3$-IterTanJit has a certain degree of robustness to preserve adversarial effects. Example meshes, their printed objects, and scanned point clouds are shown in Figure 8.

6 Conclusions

In this paper, we study learning to generate adversarial point clouds in order to attack deep models of point set classifiers. We focus on the key characterization of imperceptibility to humans that adversarial point clouds should have, which is largely overlooked in existing methods. We analyze the different mechanisms that humans perceive 2D images and 3D shapes, and propose a new method of $GeoA^3$ whose objective combines a misclassification loss of targeted attack and a new design of geometry-aware regularizer. Our proposed regularizer favors solutions with the desired surface properties of smoothness and fairness, while the targeted attack loss supports continuous pursuing of more malicious signals. The combined effect enables $GeoA^3$ to generate adversarial results that are arguably less defendable and of the key adversarial characterization of being imperceptible to humans. Comparative results confirm the advantages of our $GeoA^3$ in terms of both quantitative and qualitative measures. To generate practically more desirable adversarial surfaces, we make an attempt in this paper and propose a simple algorithm of IterTanJit to optimize $GeoA^3$. Solving $GeoA^3$ via IterTanJit can better preserve surface-level adversarial effects when re-sampling point clouds from the surface meshes reconstructed from the obtained adversarial point clouds. Experiments of both synthetic and physical attacks show the efficacy of our contributed algorithm. However, surface-level adversarial attack is still at a much lower successful rate. In future research, we are interested in improving the attacking performance either by advanced methods of adversarial point cloud generation or by directly learning to generate adversarial meshes.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[2] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[3] K. Jia, S. Li, Y. Wen, T. Liu, and D. Tao, "Orthogonal deep neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2019.

[4] C. R. Qi, H. Su, K. Mo, and L. 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.

[5] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, 2017, pp. 5099–5108.

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

[7] L. Tang, K. Chen, C. Wu, Y. Hong, K. Jia, and Z. Yang, "Improving semantic analysis on point clouds via auxiliary supervision of local geometric priors," *arXiv preprint arXiv:2001.04803*, 2020.

[8] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

[9] D. Su, H. Zhang, H. Chen, J. Yi, P.-Y. Chen, and Y. Gao, "Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 631–648.

[10] A. Fawzi, H. Fawzi, and O. Fawzi, "Adversarial vulnerability for any classifier," in *Advances in Neural Information Processing Systems*, 2018, pp. 1178–1187.

[11] C. Xiang, C. R. Qi, and B. Li, "Generating 3d adversarial point clouds," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9136–9144.

[12] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proceedings of the 2016 acm sigsac conference on computer and communications security*, 2016, pp. 1528–1540.

[13] Y. Cao, C. Xiao, D. Yang, J. Fang, R. Yang, M. Liu, and B. Li, "Adversarial objects against lidar-based autonomous driving systems," *arXiv preprint arXiv:1907.05418*, 2019.

[14] R. R. Wiyatno and A. Xu, "Physical adversarial textures that fool visual object tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 4822–4831.

[15] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," in *Advances in Neural Information Processing Systems*, 2019, pp. 125–136.

[16] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan, "Theoretically principled trade-off between robustness and accuracy," *arXiv preprint arXiv:1901.08573*, 2019.

[17] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Madry, "Adversarially robust generalization requires more data," in *Advances in Neural Information Processing Systems*, 2018, pp. 5014–5026.

[18] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[19] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 39–57.

[20] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018.

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

[22] J. Yang, Q. Zhang, R. Fang, B. Ni, J. Liu, and Q. Tian, "Adversarial attack and defense on point sets," *arXiv preprint arXiv:1902.10899*, 2019.

[23] D. Liu, R. Yu, and H. Su, "Adversarial point perturbations on 3d objects," *arXiv preprint arXiv:1908.06062*, 2019.

[24] J. S. Lappin, J. F. Norman, and F. Phillips, "Fechner, information, and shape perception," *Attention, Perception, and Psychophysics*, no. 73, pp. 2353–2378, 2011.

[25] J. Koenderink, A. Doorn, C. Christou, and J. Lappin, "Shape constancy in pictorial relief," *Perception, 25*, vol. 15, pp. 5–164, 1996.

[26] J. F. Norman, A. N. Bartholomew, and C. L. Burton, "Aging preserves the ability to perceive 3d object shape from static but not deforming boundary contours," *Acta Psychologica*, vol. 129, pp. 198–207, 2008.

[27] 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 *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2019, pp. 1961–1970.

[28] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Lévy, *Polygon mesh processing*. AK Peters/CRC Press, 2010.

[29] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu, and Z. M. Mao, "Adversarial sensor attack on lidar-based perception in autonomous driving," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 2267–2281.

[30] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond pascal: A benchmark for 3d object detection in the wild," in *IEEE winter conference on applications of computer vision*. IEEE, 2014, pp. 75–82.

[31] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[32] A. Rozsa, E. M. Rudd, and T. E. Boult, "Adversarial diversity and hard positive generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 25–32.

[33] T. Zheng, C. Chen, J. Yuan, B. Li, and K. Ren, "Pointcloud saliency maps," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2019, pp. 1598–1606.

[34] M. Wicker and M. Kwiatkowska, "Robustness of 3d deep learning in an adversarial setting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 767–11 775.

[35] T. Tsai, K. Yang, T.-Y. Ho, and Y. Jin, "Robust adversarial objects against deep learning models," in *AAAI*, 2020.

[36] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3d object reconstruction from a single image," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 605–613.

[37] K. Wang, K. Chen, and K. Jia, "Deep cascade generation on point sets," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 7 2019, pp. 3726–3732.

[38] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "A papier-mache approach to learning 3d surface generation," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018.

[39] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, "Pixel2mesh: Generating 3d mesh models from single rgb images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 52–67.

[40] J. Tang, X. Han, J. Pan, K. Jia, and X. Tong, "A skeleton-bridged deep learning approach for generating meshes of complex topologies from single rgb images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4541–4550.

[41] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *5th International Conference on Learning Representations, ICLR 2017*, 2017.

[42] M. Gambino, "Do our brains find certain shapes more attractive than others?" *Smithsonian.com*, 2013.

[43] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Point cloud auto-encoder via deep grid deformation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 206–215.

[44] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "Atlasnet: A papier-mâché approach to learning 3d surface generation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 216–224.

[45] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, 1992, pp. 71–78.

[46] K. Miura and R. Gobithaasan, "Aesthetic curves and surfaces in computer aided geometric design," *International Journal of Automation Technology*, vol. 8, pp. 304–316, 05 2014.

[47] K. Dev, M. Lau, and L. Liu, "A perceptual aesthetics measure for 3d shapes," *arXiv preprint arXiv:1608.04953*, 2016.

[48] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2574–2582.

[49] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.

[50] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European Symposium on Security and Privacy (EuroS P)*, March 2016, pp. 372–387.

[51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[52] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 32, no. 3, pp. 1–13, 2013.

[53] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. 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.