Isometric 3D Adversarial Examples in the Physical World

Yibo Miao[1], **Yinpeng Dong**[2,3†], **Jun Zhu**[2,3,4,5], **Xiao-Shan Gao**[1†]
[1] KLMM, UCAS, Academy of Mathematics and Systems Science,
Chinese Academy of Sciences, Beijing 100190, China
[2] Dept. of Comp. Sci. & Tech., Institute for AI, Tsinghua-Bosch Joint ML Center,
THBI Lab, BNRist Center, Tsinghua University, Beijing 100084, China
[3] RealAI [4] Peng Cheng Laboratory [5] Pazhou Laboratory (Huangpu), Guangzhou, China
yibomiao21@163.com, {dongyinpeng, dcszj}@tsinghua.edu.cn, xgao@mmrc.iss.ac.cn

Abstract

3D deep learning models are shown to be as vulnerable to adversarial examples as 2D models. However, existing attack methods are still far from stealthy and suffer from severe performance degradation in the physical world. Although 3D data is highly structured, it is difficult to bound the perturbations with simple metrics in the Euclidean space. In this paper, we propose a novel ϵ-isometric (ϵ-ISO) attack to generate natural and robust 3D adversarial examples in the physical world by considering the geometric properties of 3D objects and the invariance to physical transformations. For naturalness, we constrain the adversarial example to be ϵ-isometric to the original one by adopting the Gaussian curvature as a surrogate metric guaranteed by a theoretical analysis. For invariance to physical transformations, we propose a maxima over transformation (MaxOT) method that actively searches for the most harmful transformations rather than random ones to make the generated adversarial example more robust in the physical world. Experiments on typical point cloud recognition models validate that our approach can significantly improve the attack success rate and naturalness of the generated 3D adversarial examples than the state-of-the-art attack methods.

1 Introduction

Deep neural networks (DNNs) have achieved unprecedented performance on numerous tasks, including 2D image classification [33, 25, 27] and 3D point cloud recognition [49, 50, 69]. However, DNNs are vulnerable to adversarial examples [60, 20] — inputs crafted by adding imperceptible perturbations to original examples that can cause misclassification of the victim model. Adversarial examples are prevalent in various domains beyond images, including texts [28], speeches [85] and 3D objects [74]. As deep 3D point cloud recognition has been adopted in safety-critical applications, such as autonomous driving [5, 86], robotics [66, 95], medical image processing [61], it has garnered increasing attention to studying the adversarial robustness of 3D point cloud recognition models [6].

However, the existing adversarial attacks on point cloud recognition models are still far from stealthy and suffer from drastic performance degeneration in the physical world. There is usually a trade-off between the stealthiness and the real-world attacking performance, making it challenging to achieve the best of both worlds. Early methods [80, 74, 43] adopt gradient-based attacks to add, remove, and modify points, but they are limited to digital-world attacks. The KNN attack [64] and the $GeoA^3$ attack [70] constrain the smoothness of the adversarial point clouds and reconstruct adversarial

†Corresponding authors.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Figure 1: An illustration of adversarial objects crafted by KNN attack [64], $GeoA^3$ attack [70], Mesh Attack [89] and our ϵ-ISO attack against the PointNet model: KNN attack and $GeoA^3$ attack can produce unnatural adversarial objects (and often low success rates); Mesh Attack can generate a lot of distortions; while ϵ-ISO attack improves the naturalness of the 3D adversarial sample and ensures the consistency between the intrinsic geometric properties of the adversarial and original 3D objects [8].

meshes from the point clouds that can be 3D-printed in the physical world. Although these works demonstrate successful physical attacks, point cloud reconstruction introduces large noises and errors, resulting in low attack success rates and unnaturalness of the adversarial objects in the physical world. Mesh Attack [89] is recently proposed to perturb the mesh representation of 3D objects, which improves the success rate but often creates large distortions that can be easily detected by humans as anomalies, as illustrated in Fig. 1. Overall, it is difficult to achieve both the naturalness and effectiveness of 3D adversarial attacks in the physical world, which we think is largely due to the lack of an appropriate metric to characterize the naturalness of 3D data.

To address these issues, we propose an ϵ-isometric (**ϵ-ISO**) attack method to generate natural and robust 3D adversarial examples in the physical world against point cloud recognition models. The ϵ-ISO attack improves the naturalness of the 3D adversarial example by constraining it to be ϵ-isometric (see Definition 1) to the original one, which guarantees the consistency between the intrinsic geometric properties of two 3D objects [8]. We theoretically demonstrate that Gaussian curvature (see Definition 2) can be used to provide a sufficient condition to ensure that two surfaces are ϵ-isometric. Due to the computable and differentiable nature of Gaussian curvature, we adopt it as a new regularization loss to practically generate natural 3D adversarial examples. To improve the robustness of 3D adversarial examples under physical transformations, we further propose a maxima over transformation (**MaxOT**) method that actively searches for the most harmful transformations rather than random ones [1] for optimization. Armed with Bayesian optimization that provides better initialization of the transformations, MaxOT is able to find a set of diverse worst-case transformations, leading to improved performance of the 3D adversarial examples in the physical world.

We conduct extensive experiments to evaluate the performance of our method on attacking typical point cloud recognition models [49, 50, 69]. Results demonstrate that, in comparison with the alternative state-of-the-art attack methods [64, 70, 89], ϵ-ISO attack achieves higher success rates, while making the generated adversarial examples more natural and robust under physical transformations. A physical-world experiment is conducted by 3D-printing the adversarial meshes and re-scanning the objects for evaluation, which also validates the effectiveness of our method.

2 Related work

Deep learning on 3D point clouds. Deep 3D point cloud recognition [49, 21, 75, 77, 82, 51, 62] has emerged in recent years with various applications in many fields, such as 3D object classification [56, 38, 84, 87], 3D scene segmentation [22, 68, 78, 26], and 3D object detection in autonomous driving [98, 79]. One of the pioneering works is PointNet [49], which directly applies a multilayer perceptron to learn point features and aggregates them in an efficient way using a max-pool module. PointNet++ [50] and a large number of later works [13, 44, 81] are built on PointNet to further capture fine-grained local structure information from the neighborhood of each point. Recently, some works have focused on designing special convolutions on 3D domains [2, 40, 45, 63] or developing graph neural networks [18, 53, 54, 69] to improve point cloud recognition.

3D adversarial attacks. Following the previous studies on adversarial machine learning in the 2D image domain [60, 35, 4, 10, 19, 83, 11, 12], many works [74, 3, 42] apply adversarial attacks to the

3D point cloud domain. Xiang et al. [74] proposed point generation attacks by adding a limited number of synthetic points to the point cloud. Recently, more studies [71, 94] use gradient-based attack methods to identify key points from the point cloud for deletion. More point perturbation attacks [24, 46, 92, 9] learn to perturb the xyz coordinates of each point through a C&W framework [4] based on metrics defined in the Euclidean space. Zhao et al. [93] attack by the isometric transformations in the Euclidean space such as rotation. It is worth noting that we consider isometric mappings between surfaces, which is essentially different from [93]. Later works [36, 88] further apply iterative gradient methods to achieve more advanced adversarial perturbations. Besides, other works consider generative models [97], 3D data attacks [65, 58], adversarial robustness [90, 59], attacks against LIDAR [37, 30], autonomous driving [57, 91], backdoor attacks [39, 76], etc., in the 3D domain. However, the existing attacks on 3D point cloud recognition are still far from stealthy and the only three methods that consider the physical-world attacks [64, 70, 89] are not very effective. In this paper, we surpass the performance of previous methods by proposing a novel ϵ-isometric (ϵ-ISO) attack method to generate natural and robust 3D adversarial examples in the physical world.

3 Methodology

We now formally present **ϵ-ISO attack**. We first present the general problem formulation, and then describe how ϵ-ISO attack enhances the imperceptibility and robustness of the generated 3D adversarial samples, respectively.

3.1 Problem formulation

To generate 3D adversarial objects in the physical world, it is more straightforward to perturb the mesh representation of 3D objects rather than point clouds [89] since the reconstruction process can incur large errors [64, 70]. A mesh $\mathcal{M} = (\mathcal{V}, \mathcal{F})$ is an approximate shape representation of its underlying surface, where $\mathcal{V} := \{v_i\}_{i=1}^{n_v}$ is the set of n_v vertices of xyz coordinates, and $\mathcal{F} := \{z_i\}_{i=1}^{n_f}$ is the set of n_f triangle faces represented by the indices of vertices. We let S denote a differentiable sampling process such that $\mathcal{P} := S(\mathcal{M}) \in \mathcal{X}$ is the point cloud obtained by randomly sampling on the mesh surface, where $\mathcal{X}$ is the set of all point clouds. We let $y \in \mathcal{Y}$ denote the corresponding ground-truth label of $\mathcal{M}$ as well as $\mathcal{P}$.

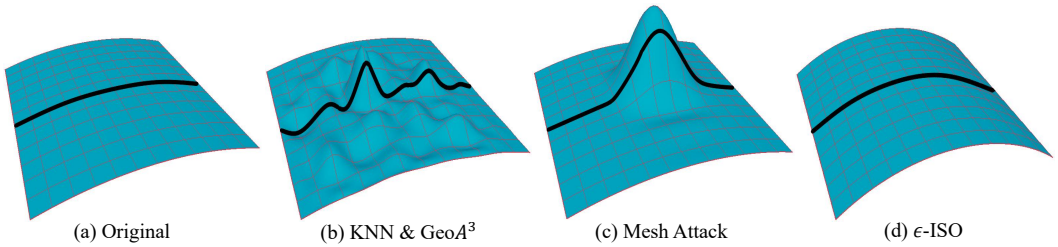
In this paper, we focus on the challenging *targeted attacks* against deep 3D point cloud classification models [49, 50, 69]. Given a point cloud classifier $f : \mathcal{X} \to \mathcal{Y}$, the goal of the attack is to generate an adversarial mesh $\mathcal{M}_{adv} = (\mathcal{V}_{adv}, \mathcal{F})$ for the original one $\mathcal{M}$ with vertex perturbations such that the sampled point cloud $\mathcal{P}_{adv} := S(\mathcal{M}_{adv})$ will be misclassified by f as the target class $y^*(\neq y)$. In general, the perturbation should be small to make the adversarial mesh $\mathcal{M}_{adv}$ inconspicuous under human inspection. Thus, the optimization problem of generating the adversarial mesh can be generally formulated as

$$\min_{\mathcal{M}_{adv}} \mathcal{L}_f \left(S \left(\mathcal{M}_{adv} \right), y^* \right) + \beta \cdot \mathcal{R} \left(\mathcal{M}_{adv}, \mathcal{M} \right), \tag{1}$$

where $\mathcal{L}_f$ is the loss that facilitates the misclassification of $\mathcal{P}_{adv}$ to y^*, $\mathcal{R}$ is the regularization term that minimizes a perceptibility distance between $\mathcal{M}_{adv}$ and $\mathcal{M}$, and β is a balancing hyperparameter between these two losses. In this paper, we try to develop a stealthy and robust attack method by proposing a new regularization loss $\mathcal{R}$ based on Gaussian curvature with theoretical guarantees to remain the naturalness as well as a new attacking loss $\mathcal{L}_f$ to enhance the robustness of the generated 3D adversarial objects under physical transformations. $\mathcal{R}$ and $\mathcal{L}_f$ will be introduced in the following.

3.2 ϵ-ISO attack

Most of the existing 3D adversarial attacks only consider the constraints $\mathcal{R}$ defined in the Euclidean space [15, 82, 23]. The generated adversarial examples have noticeable point outliers that cause spikes to appear on the object's surface, thus losing the naturalness. Moreover, the outliers are more easily removed and defended against. The main reason is that the existing methods do not consider the geometric properties of the 3D objects. In differential geometry, isometric mapping guarantees the consistency of the intrinsic geometric features of two objects [8]. Therefore, we propose a constraint loss $\mathcal{R}$ based on ϵ-isometric mapping to restrict the naturalness of 3D adversarial objects. We first give the definition of ϵ-isometric below.



(a) Original (b) KNN & GeoA^3 (c) Mesh Attack (d) ϵ-ISO

Figure 2: An illustration of ϵ-isometric attack. (a): Original mesh. (b) and (c): Adversarial meshes generated by KNN & $GeoA^3$ attack and Mesh Attack, respectively. They consider only the constraints defined in the Euclidean space, and the curve lengths (shown as the black curves) of the generated adversarial examples differ significantly from those of the original samples, which do not satisfy ϵ-isometric and lose naturalness. (d): Adversarial mesh generated by ϵ-ISO. We consider the geometric features of 3D objects and constrain the 3D adversarial example to be ϵ-isometric to the original one, such that the curve lengths of the generated adversarial samples vary little and have naturalness.

Definition 1. *Let S and $\tilde{S}$ denote two surfaces of $\mathbb{R}^3$. A diffeomorphism $\varphi : S \to \tilde{S}$ is called an ϵ-isometric mapping if there exists a constant n such that it takes any local curve C in S to a curve $\tilde{C} = \varphi(C)$ in $\tilde{S}$ satisfying $(1 - n\epsilon)s(C) < s(\tilde{C}) < (1 + n\epsilon)s(C)$ where $s(\cdot)$ is the length. The surfaces S and $\tilde{S}$ are then said to be ϵ-isometric.*

As shown in Fig. 2, Fig. 2(a) is the original mesh, Fig. 2(b) is the adversarial mesh generated by KNN attack and $GeoA^3$ attack, and Fig. 2(c) is the adversarial mesh generated by Mesh Attack. These three methods only consider the constraints defined in the Euclidean space, and the curve lengths of the generated adversarial samples differ greatly from those of the original samples, which are not ϵ-isometric and lose naturalness. Fig. 2(d) is the adversarial mesh generated by our proposed ϵ-ISO attack. We consider the geometric features of 3D objects to generate natural adversarial examples by constraining them to be ϵ-isometric to the original examples (i.e., the curve length of the resulting adversarial examples varies very little). However, it is intractable to directly optimize the adversarial mesh to be ϵ-isometric as the original one. Therefore, we give the definition of the Gaussian curvature of the surface from [8].

Definition 2. *Let S be a surface of $\mathbb{R}^3$ parameterized by $\boldsymbol{r} := \boldsymbol{r}(u, v) = [x(u, v), y(u, v), z(u, v)]$, where $(u, v) \in \mathbb{R}^2$. We let $\boldsymbol{r}_u, \boldsymbol{r}_v$ denote the partial derivatives of $\boldsymbol{r}$ w.r.t. u and v, $\boldsymbol{r}_{uu}, \boldsymbol{r}_{uv}, \boldsymbol{r}_{vv}$ denote the second partial derivatives of $\boldsymbol{r}$, and $\wedge, \langle\cdot,\cdot\rangle$ denote the outer product and inner product, respectively. The parametrization thus defines unit normal vectors $\boldsymbol{n} := \frac{\boldsymbol{r}_u \wedge \boldsymbol{r}_v}{|\boldsymbol{r}_u \wedge \boldsymbol{r}_v|}$ of the surface S. We denote the eigenvalues of the coefficient matrix of the Weingarten map $\begin{bmatrix} L & M \\ M & N \end{bmatrix} \begin{bmatrix} E & F \\ F & G \end{bmatrix}^{-1}$ as k_1 and k_2, where $E = \langle \boldsymbol{r}_u, \boldsymbol{r}_u \rangle$, $F = \langle \boldsymbol{r}_u, \boldsymbol{r}_v \rangle$ and $G = \langle \boldsymbol{r}_v, \boldsymbol{r}_v \rangle$ are coefficients of the first fundamental form and $L = \langle \boldsymbol{r}_{uu}, \boldsymbol{n} \rangle$, $M = \langle \boldsymbol{r}_{uv}, \boldsymbol{n} \rangle$ and $N = \langle \boldsymbol{r}_{vv}, \boldsymbol{n} \rangle$ are coefficients of the second fundamental form. The Gaussian curvature is defined as $K = k_1 k_2 = \frac{LN - M^2}{EG - F^2}$.*

Remark 1. The Gaussian curvature intrinsically measures the bending degree of the surface reflected by the Gaussian mapping. Let the area element of the surface S be $dA = \langle \boldsymbol{r}_u \wedge \boldsymbol{r}_v, \boldsymbol{n} \rangle \, dudv$ and the area element under the Gaussian mapping $g : \begin{array}{l} S \in \mathbb{R}^3 \to S^2 \\ \boldsymbol{r}(u, v) \to \boldsymbol{n}(u, v) \end{array}$ be $dA' = \langle \boldsymbol{n}_u \wedge \boldsymbol{n}_v, \boldsymbol{n} \rangle \, dudv$. From $\boldsymbol{n}_u \wedge \boldsymbol{n}_v = K \boldsymbol{r}_u \wedge \boldsymbol{r}_v$ (proof in Appendix D), we obtain

$$\lim_{D \to P} \frac{\text{Area } (g(D))}{\text{Area } (D)} = \lim_{D \to P} \frac{\int_{g(D)} dA'}{\int_D dA} = \lim_{D \to P} \frac{\int_D K dA}{\int_D dA} = K(P). \tag{2}$$

Eq. (2) illustrates that the geometric meaning of Gaussian curvature is the ratio of the area of the domain at the point P on the surface S and the area of the domain at the corresponding point under the Gaussian mapping, i.e., the bending degree of the surface reflected by the Gaussian mapping.

Based on Definitions 1 and 2, we have the following theorem.

Theorem 1 (proof in Appendix A)**.** *Let S and $\tilde{S}$ denote two surfaces of $\mathbb{R}^3$; $\varphi : S \to \tilde{S}$ denote a diffeomorphism that takes a point v in S to point $v' = \varphi(v)$ in $\tilde{S}$; and $K(\cdot)$ be the Gaussian curvature of the points. If $|K(v) - K(v')| < \epsilon$ for any point v, then the surfaces S and $\tilde{S}$ are ϵ-isometric.*

Theorem 1 indicates that to make two surfaces ϵ-isometric, one can constrain their Gaussian curvatures. Since the Gaussian curvature is computable and differentiable w.r.t. vertices, we adopt it to constrain the naturalness of 3D adversarial meshes as

$$\mathcal{R}_{Gauss}\left(\mathcal{M}_{adv}, \mathcal{M}\right) = \frac{1}{n_v} \sum_{v \in \mathcal{V}, v' = \varphi(v) \in \mathcal{V}_{adv}} \left\| K\left(v'\right) - K(v) \right\|_2^2, \tag{3}$$

where $\varphi(\cdot)$ is the corresponding mapping between vertices in $\mathcal{V}$ and $\mathcal{V}_{adv}$. We follow the Gauss-Bonnet formula [7] to calculate the Gaussian curvature of the vertices as

$$K(v) = \frac{2\pi - \sum_{i \in N(v)} \theta_i(v)}{A(v)}, \tag{4}$$

where $A(\cdot)$ is the area of the vertex neighborhood, i.e., the area of the polygonal region joined by the consecutive midpoints of triangles incident on the vertex of interest, $N(v)$ is the set of faces containing v, and $\theta_i(v)$ is the interior angle of the face at vertex v. Note that the value of $\sum_{i \in N(v)} \theta_i(v)$ for a plane is 2π and the Gaussian curvature is 0. The more curved the surface, the smaller the value of $\sum_{i \in N(v)} \theta_i(v)$ and the larger the Gaussian curvature.

In addition, we prevent the generated adversarial meshes from self-intersecting by using the Laplace loss [16], denoted as $\mathcal{R}_{Lap}\left(\mathcal{M}_{adv}\right)$, which represents the distance between a vertex and its nearest neighbor's center of mass, and the edge length loss [67], denoted as $\mathcal{R}_{edge}\left(\mathcal{M}_{adv}\right)$, which represents the smoothness of the surface. Thus, the overall regularization term can be expressed as:

$$\mathcal{R}\left(\mathcal{M}_{adv}, \mathcal{M}\right) = \lambda_1 \cdot \mathcal{R}_{Gauss}\left(\mathcal{M}_{adv}, \mathcal{M}\right) + \lambda_2 \cdot \mathcal{R}_{Lap}\left(\mathcal{M}_{adv}\right) + \lambda_3 \cdot \mathcal{R}_{edge}\left(\mathcal{M}_{adv}\right), \tag{5}$$

where λ_1, λ_2 and λ_3 are balancing hyperparameters.

3.3 Improving the robustness under physical transformations

Besides concerning the naturalness of 3D adversarial examples, we further enhance their robustness under physical transformations, such as 3D rotations, affine projections, cutouts, etc. A common method is the expectation over transformation (EOT) algorithm [1], which optimizes the adversarial example over the distribution of different transformations. However, it is still challenging to maintain the attacking performance under various physical transformations. As shown in the experiments, after using the EOT algorithm, there are still some transformations that the generated adversarial examples are not robust to, leading to a reduction of the attack success rate.

To address this issue, our key insight is to consider the worst-case transformations rather than their expectation, since if the adversarial examples are resistant to the most harmful physical transformations, they can also resist much weaker transformations, inspired by adversarial training [47]. Therefore, we propose a **maxima over transformation (MaxOT)** algorithm to actively search for physical transformations that maximize the misclassification loss. The loss function $\mathcal{L}_f$ is thus formulated as:

$$\mathcal{L}_f(S(\mathcal{M}_{adv}), y^*) = \max_{T^* \subset T} \mathbb{E}_{t \in T^*} \mathcal{L}_{ce}\left(t(S(\mathcal{M}_{adv})), y^*\right), \tag{6}$$

where T contains all possible transformations, T^* is a subset of transformations in T, and $\mathcal{L}_{ce}$ is the cross-entropy loss. Note that in Eq. (6) we consider a subset of transformations T^* rather than a single one because the loss landscape w.r.t. transformations is largely non-convex and contains many local maxima [14]. Thus we aim to find a set of diverse worst-case transformations. By integrating Eq. (6) into Eq. (1), it forms a minimax optimization problem, where the inner maximization aims to find physical transformations that maximize the cross-entropy loss, while the outer minimization aims to optimize the adversarial example with the worst-case transformations.

3.3.1 Bayesian optimization

To solve problem (6), we search for the worst-case transformations one by one. Given an initialized transformation, we perform gradient-based optimization to update the transformation parameters (e.g., angles for rotations). However, randomly selecting initialized transformations is ineffective since the random initialization may drop into regions of weak transformations, which limits the exploration of the space of all transformations. To address this issue, we propose to adopt the Bayesian optimization [17, 55] to better break the dilemma between exploration and exploitation to find more appropriate initialized transformations.

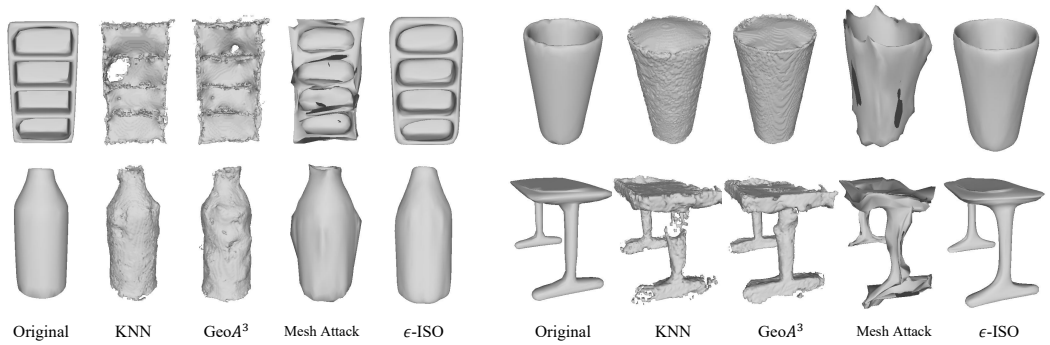


Figure 3: Adversarial objects obtained by KNN attack, $GeoA^3$ attack, Mesh Attack, and our ϵ-ISO attack against the PointNet model. The KNN and $GeoA^3$ attack produce adversarial samples with dense irregular deformations. The Mesh Attack produces distortions and self-intersecting surfaces (i.e., black surfaces). None of them guarantees naturalness, while ϵ-ISO attack is more natural.

Bayesian optimization is an efficient method for solving global optimization problems consisting of two key components: a surrogate model, such as a Gaussian process (GP) [52] or a Bayesian neural network (BNN) [32], which models the unknown objective; and an acquisition function $\alpha(\cdot)$, which is maximized by balancing exploitation and exploration to recommend the next query location. We choose GP as a surrogate model, which provides a Bayesian posterior distribution to describe the unknown function $\mathcal{L}_{ce}\left(t(S(\mathcal{M}_{adv})), y^*\right)$. We use the expected improvement (EI) [29, 48] acquisition function, which measures the expected improvement of each point with respect to the current best value. Then, the unknown function is sampled by maximizing the acquisition function to better explore the space of all transformations by selecting the next query point in the region where the prediction is high and the model is very uncertain.

As shown in the overall algorithm (in Appendix B), we update the Bayesian posterior distribution of the objective f using the observations obtained from the previous iterations in the gradient descent process. Then, we maximize the EI acquisition function $\alpha_{EI}\left(t; D_n\right)$ to find the initial transformation. From the initialization, we optimize the transformation parameters by gradient-based method to solve problem (6). This process is repeated for the number of transformations in the MaxOT algorithm.

4 Experiments

4.1 Experimental setup

Dataset. We use the ModelNet40 [72] dataset in our experiments. This dataset contains 12,311 CAD models with 40 common object semantic categories in the real world. We use the official split [49, 50], where 9,843 examples are used for training and the remaining 2,468 examples are used for testing. We follow [89] and get closed manifolds. For adversarial attacks, we follow [70] and take all the instances of the 40 categories that are well classified in the ModelNet40 testing set.

Victim models. Following [70, 89, 41], we select three commonly used point cloud classification networks as the victim models, i.e., PointNet [49], PointNet++ [50], and DGCNN [69].

Evaluation metrics. To quantitatively evaluate the effectiveness of our proposed ϵ-ISO attack, we measure the attack success rate (ASR). Besides, to measure the imperceptibility of different attack methods, we use the Chamfer distance $\mathcal{D}_c$ [15] and Gaussian curvature distance $\mathcal{D}_g$ (i.e., Eq. (3)) as evaluation metrics. We also report the attack success rates on several existing defenses [80, 96, 73] and under the physical world to further validate the superiority of our ϵ-ISO attack.

Implementation details. We use Adam [31] to optimize the objective of our proposed ϵ-ISO attack. We use a fixed learning schedule of 250 iterations, where the learning rate and momentum are respectively set as 0.01 and 0.9. We assign the weighting parameters $\lambda_1 = 1.0$, $\lambda_2 = 0.2$ and $\lambda_3 = 0.8$. The balancing parameter β is initialized as 1,500 and automatically adjusted by conducting 10 runs of binary search following [4]. We uniformly sample 1,024 points from the adversarial mesh.

Table 1: Quantitative results of KNN attack, $GeoA^3$ attack, Mesh Attack, and our proposed ϵ-ISO attack against different models. Our proposed ϵ-ISO attack outperforms all existing methods in terms of attack success rate (ASR) and imperceptibility. We adopt the Chamfer distance $\mathcal{D}_c$ [15] and Gaussian curvature distance $\mathcal{D}_g$ as the evaluation metrics.

Model	PointNet			PointNet++			DGCNN		
	ASR	$\mathcal{D}_c$	$\mathcal{D}_g$	ASR	$\mathcal{D}_c$	$\mathcal{D}_g$	ASR	$\mathcal{D}_c$	$\mathcal{D}_g$
KNN	14.78%	0.0034	0.0096	6.24%	**0.0027**	0.0122	4.17%	0.0036	0.0125
$GeoA^3$	19.65%	0.0066	0.0037	11.20%	0.0192	0.0031	8.24%	0.0172	0.0042
Mesh Attack	88.10%	0.0054	0.0048	94.79%	0.0045	0.0046	66.79%	0.0051	0.0055
ϵ-ISO	**98.45%**	**0.0031**	**0.0004**	**99.58%**	0.0040	**0.0004**	**84.16%**	**0.0032**	**0.0009**

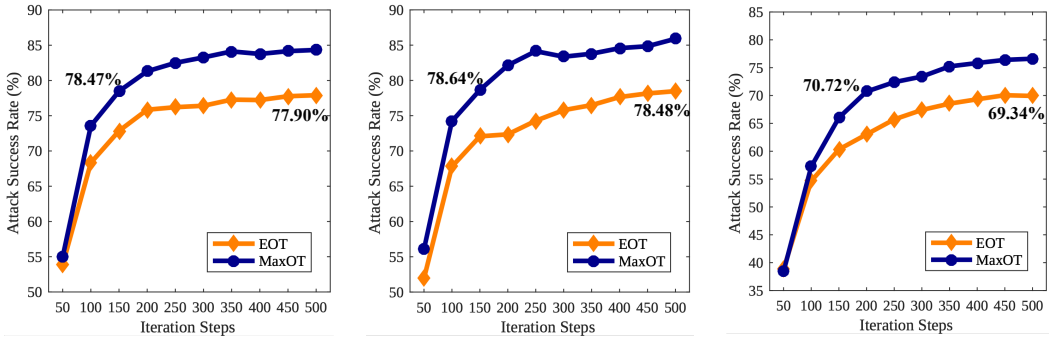


Figure 4: The attack success rate w.r.t number of iterations curves of EOT and MaxOT against PointNet, PointNet++, and DGCNN. Our proposed MaxOT algorithm can lead to a higher attack success rate with a small number of iterations than that of EOT with a large number of iterations.

4.2 Pseudo physical attack

In this section, we compare our proposed ϵ-ISO attack with existing methods [64, 70, 89] that generate adversarial objects under the setting of white-box targeted attack, including the KNN attack [64], $GeoA^3$ attack [70], and Mesh Attack [89]. In Table 1, we compare different methods under the measures of attack success rate (ASR), Chamfer distance $\mathcal{D}_c$ [15] and Gaussian curvature distance $\mathcal{D}_g$. Our proposed ϵ-ISO attack outperforms all existing methods in terms of attack success rate and is almost optimal in terms of Chamfer distance $\mathcal{D}_c$ and Gaussian curvature distance $\mathcal{D}_g$, the evaluation metrics for measuring imperceptibility. These comparisons confirm the effectiveness of our proposed regularization term based on Gaussian curvature, to simultaneously achieve the strongest adversarial attack and the most natural perturbations.

Fig. 3 shows the adversarial objects generated by different methods. The KNN attack and $GeoA^3$ attack produce dense irregular deformations on the mesh surface, losing smoothness and regularity. The Mesh Attack creates severe deformations and spurs, which are easily perceived by humans visually. In addition, the severe irregularities and distortions generated by the Mesh Attack can lead to self-intersecting phenomena in the mesh of 3D objects, i.e., the black surfaces in Fig. 3. These self-intersecting surfaces can prohibit 3D-printing the adversarial objects in the physical world. By considering geometric features of the object instead of using metrics in the Euclidean space, our ϵ-ISO attack produces adversarial samples without dense deformations, bursts, distortions, and self-intersection phenomena, while are stealthy and natural.

4.3 Robustness under physical transformation

To further verify the superior robustness of our proposed MaxOT algorithm under transformations in the physical world, we compare MaxOT to EOT. In Table 2, we compare the different methods under rotations to calculate the attack success rate, Chamfer distance $\mathcal{D}_c$ and Gaussian curvature distance $\mathcal{D}_g$. Our proposed MaxOT algorithm outperforms the EOT algorithm in terms of attack success rate. This comparison confirms our insight that the resistance to the most harmful physical transformations is better than random ones.

Table 2: Quantitative results of attacking different models using EOT and MaxOT. Our proposed MaxOT algorithm outperforms the EOT algorithm in terms of attack success rate.

Model	PointNet			PointNet++			DGCNN		
	ASR	$\mathcal{D}_c$	$\mathcal{D}_g$	ASR	$\mathcal{D}_c$	$\mathcal{D}_g$	ASR	$\mathcal{D}_c$	$\mathcal{D}_g$
EOT	76.20%	0.0074	0.0009	74.28%	0.0094	0.0007	65.72%	0.0068	0.0041
MaxOT	**82.50%**	0.0074	0.0009	**84.14%**	0.0094	0.0006	**72.40%**	0.0067	0.0039

Table 3: The attack success rate of various methods against PointNet under different defense methods. Our proposed ϵ-ISO attack performs better under stronger defenses, and the attack success rate under DUP-Net and IF-Defence is higher than all other attacks.

	no defence	SRS	SOR	DUP-Net	IF-Defence
KNN	32.15%	24.34%	13.36%	11.25%	7.46%
$GeoA^3$	40.87%	3.34%	33.22%	11.97%	1.04%
Mesh Attack	93.39%	**87.38%**	**89.93%**	78.28%	49.06%
ϵ-ISO	**99.81%**	79.85%	85.42%	**78.55%**	**60.51%**

Table 4: The success rate of attacking the PointNet model in the physical world using the EOT algorithm and our proposed MaxOT algorithm.

	PointNet
EOT	72.36%
MaxOT	**80.12%**

In Fig. 4, we give the success rate w.r.t. iterations curves for EOT and MaxOT of different victim models. Our proposed MaxOT algorithm is always ahead of the EOT algorithm at different number of iterations. Moreover, we note that our proposed MaxOT algorithm can lead to higher attack success rate with a small number of iterations than that of EOT with a large number of iterations. Therefore, our proposed MaxOT algorithm is more efficient.

4.4 Performance under defense

We further verify the effectiveness of our ϵ-ISO attack under the existing defense methods. To evaluate the adversarial robustness of various attacks, we use the following defense methods: Simple Random Sampling (SRS) [80], Statistical Outlier Removal (SOR) [96], DUP-Net defense [96], and IF-Defense [73]. We give the attack success rate of various attacks under each defense method in Table 3. Under simple defenses such as SRS and SOR, the attack success rate of Mesh Attack is higher than our ϵ-ISO attack.

However, under more advanced and effective defenses such as IF-Defence, the attack success rate of our ϵ-ISO attack is higher than all other attacks. This is because KNN attack and $GeoA^3$ attack generate adversarial objects through dense local deformation, generating outliers that are easily detected by the defense methods. The Mesh Attack generates adversarial objects through large anomalous deformation, resulting in fewer local outliers that have advantages under simple statistical defenses such as SRS and SOR, but cannot effectively attack the IF-Defence with better defense performance. Our ϵ-ISO attack produces adversarial samples without local outliers or anomalous deformations, with better attack performance, especially under IF-Defence.

4.5 Physical attack

We randomly select 100 instances in Table 2, 50 of which are generated by the MaxOT algorithm and the other 50 are the corresponding instances generated by the EOT algorithm. The selected meshes are printed by the Stratsys J850 Prime 3D printer and scanned by the EinScan-SE 3D scanner. The attack success rate in the physical world is shown in Table 4, where our algorithm almost completely maintains the effect of the pseudo physical world attack. None of the previous algorithms could maintain the pseudo physical world attack well in the physical world because the dense deformation and self-intersection phenomena would interfere with the scanning and prevent the scanner from correctly scanning the surface of the adversarial samples generated by their algorithms. In contrast, the surface of the adversarial sample produced by our ϵ-ISO attack is smooth and natural and can be scanned correctly by the scanner, preserving the adversarial effect. In addition, the improvement on the attack success rate of our proposed MaxOT algorithm against the EOT algorithm is also reflected in the physical world. Our proposed MaxOT algorithm is necessary to improve the robustness of the adversarial sample under the transformation of the physical world. The adversarial meshes, 3D printed physical meshes, and scanned point clouds are shown in Fig. 5.

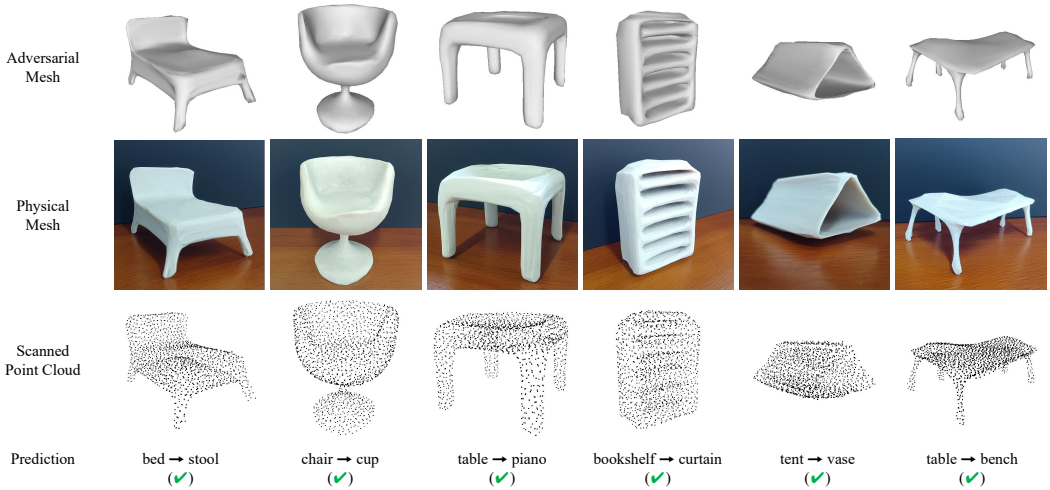


Figure 5: Visualization of the ϵ-ISO attack in the physical world. The adversarial mesh is randomly selected from Table 2. After 3D-printing and scanning, the point cloud obtained from the scanning is used to attack the PointNet. A green check mark indicates a successful attack. The adversarial samples we generated are natural and stealthy, and because of their smoothness and naturalness, can be scanned correctly by the 3D-scanner, maintaining the adversarial effect in the physical world.

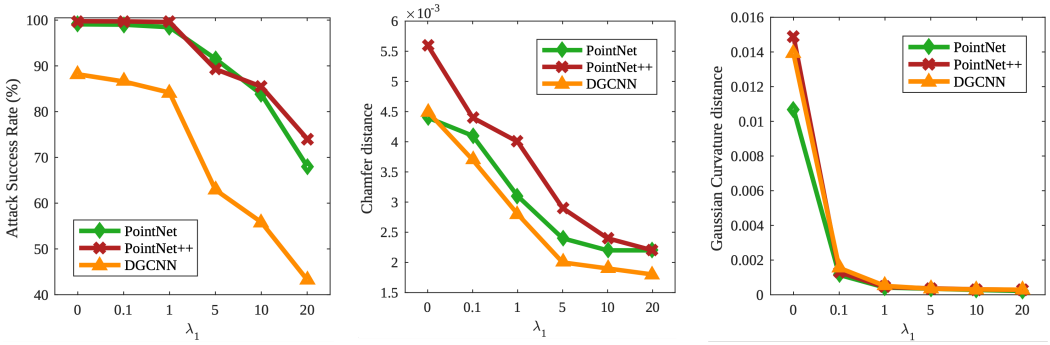


Figure 6: The effects of the penalty parameter λ_1 of the Gaussian curvature consistency regularization term. When λ_1 is large, the attack success rate decreases rapidly. When λ_1 is small, the naturalness of the adversarial objects is worse. The default $\lambda_1 = 1$ gives the best result for balancing the attack success rate and stealthiness.

4.6 Ablation studies

To investigate the effect of the penalty parameter λ_1 of the Gaussian curvature consistency regularization term on the attack success rate and imperceptibility metrics, we adjust the value of λ_1 to perform quantitative and qualitative experiments. The plots in Fig. 6 show that when λ_1 is tuned high, the attack success rate decreases rapidly. When λ_1 is adjusted lower, the values of the evaluation metrics increase and deteriorate rapidly. The qualitative visualization results in Appendix C demonstrate the irregularity of the 3D objects as λ_1 is turned down. Our default value of $\lambda_1 = 1$ gives the best result for balancing the attack success rate and stealthiness. To investigate the effectiveness of Bayesian optimization to find the initial transformations, we test the attack success rate for randomly selecting the initial transformation in the MaxOT algorithm. As shown in Appendix C, the improvement of using Bayesian optimization over random initialization in the MaxOT algorithm is significant. This confirms the importance of Bayesian optimization methods in our approach. More ablation studies can be found in Appendix C.

5 Conclusion

In this paper, we propose a novel ϵ-isometric (ϵ-ISO) attack method to generate natural and robust 3D adversarial examples in the physical world. We improve the naturalness of 3D adversarial examples by constraining it to be ϵ-isometric to the original one. Through a theoretical analysis, we adopt

the Gaussian curvature as the surrogate metric. We further propose a maxima over transformation (MaxOT) algorithm armed with Bayesian optimization that actively searches for the most harmful transformations rather than random ones to make the generated adversarial example more robust in the physical world. Extensive Experiments on typical point cloud recognition models validate the effectiveness of our approach in terms of attack success rate and naturalness compared to the state-of-the-art methods. A potential negative societal impact of our work in that the malicious adversaries may adopt our method to generate 3D adversarial objects in the physical world, which can cause severe security/safety consequences for real-world applications. Thus it is imperative to develop more robust 3D recognition models, which we leave to future work.

Acknowledgement

Yibo Miao and Xiao-Shan Gao were supported by NKRDP grant No.2018YFA0704705 and NSFC grant No.12288201. Yinpeng Dong and Jun Zhu were supported by the National Key Research and Development Program of China (2020AAA0106000, 2020AAA0104304, 2020AAA0106302), NSFC Projects (Nos. 62061136001, 62076145, 62076147, U19B2034, U1811461, U19A2081, 61972224), Beijing NSF Project (No. JQ19016), BNRist (BNR2022RC01006), Tsinghua Institute for Guo Qiang, and the High Performance Computing Center, Tsinghua University. Y. Dong was also supported by the China National Postdoctoral Program for Innovative Talents and Shuimu Tsinghua Scholar Program. J. Zhu was also supported by the XPlorer Prize.

References

[1] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293. PMLR, 2018.

[2] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point convolutional neural networks by extension operators. *arXiv preprint arXiv:1803.10091*, 2018.

[3] Yulong Cao, Chaowei Xiao, Dawei Yang, Jing Fang, Ruigang Yang, Mingyan Liu, and Bo Li. Adversarial objects against lidar-based autonomous driving systems. *arXiv preprint arXiv:1907.05418*, 2019.

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

[5] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017.

[6] Wenda Chu, Linyi Li, and Bo Li. Tpc: Transformation-specific smoothing for point cloud models. *arXiv preprint arXiv:2201.12733*, 2022.

[7] David Cohen-Steiner and Jean-Marie Morvan. Restricted delaunay triangulations and normal cycle. In *Proceedings of the nineteenth annual symposium on Computational geometry*, pages 312–321, 2003.

[8] Manfredo P Do Carmo. *Differential geometry of curves and surfaces: revised and updated second edition*. Courier Dover Publications, 2016.

[9] Xiaoyi Dong, Dongdong Chen, Hang Zhou, Gang Hua, Weiming Zhang, and Nenghai Yu. Self-robust 3d point recognition via gather-vector guidance. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11513–11521. IEEE, 2020.

[10] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[11] Yinpeng Dong, Zhijie Deng, Tianyu Pang, Jun Zhu, and Hang Su. Adversarial distributional training for robust deep learning. *Advances in Neural Information Processing Systems*, 33:8270–8283, 2020.

[12] Yinpeng Dong, Ke Xu, Xiao Yang, Tianyu Pang, Zhijie Deng, Hang Su, and Jun Zhu. Exploring memorization in adversarial training. *arXiv preprint arXiv:2106.01606*, 2021.

[13] Yueqi Duan, Yu Zheng, Jiwen Lu, Jie Zhou, and Qi Tian. Structural relational reasoning of point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 949–958, 2019.

[14] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In *International Conference on Machine Learning*, pages 1802–1811, 2019.

[15] Haoqiang Fan, Hao Su, and Leonidas 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*, pages 605–613, 2017.

[16] David A Field. Laplacian smoothing and delaunay triangulations. *Communications in applied numerical methods*, 4(6):709–712, 1988.

[17] Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[18] Xiang Gao, Wei Hu, and Guo-Jun Qi. Graphter: Unsupervised learning of graph transformation equivariant representations via auto-encoding node-wise transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7163–7172, 2020.

[19] Justin Gilmer, Ryan P Adams, Ian Goodfellow, David Andersen, and George E Dahl. Motivating the rules of the game for adversarial example research. *arXiv preprint arXiv:1807.06732*, 2018.

[20] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

[21] Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. Revisiting point cloud shape classification with a simple and effective baseline. In *International Conference on Machine Learning*, pages 3809–3820. PMLR, 2021.

[22] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9224–9232, 2018.

[23] T Groueix, M Fisher, VG Kim, BC Russell, and M Aubry. Atlasnet: A papier-mâché approach to learning 3d surface generation. arxiv 2018. *arXiv preprint arXiv:1802.05384*, 1802.

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

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[26] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Learning semantic segmentation of large-scale point clouds with random sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[27] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[28] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025, 2020.

[29] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

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

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

[32] Igor Kononenko. Bayesian neural networks. *Biological Cybernetics*, 61(5):361–370, 1989.

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2012.

[34] Wolfgang Kühnel. *Differential geometry*, volume 77. American Mathematical Soc., 2015.

[35] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *International Conference on Learning Representations (ICLR) Workshops*, 2017.

[36] Itai Lang, Uriel Kotlicki, and Shai Avidan. Geometric adversarial attacks and defenses on 3d point clouds. In *2021 International Conference on 3D Vision (3DV)*, pages 1196–1205. IEEE, 2021.

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

[38] Huan Lei, Naveed Akhtar, and Ajmal Mian. Spherical kernel for efficient graph convolution on 3d point clouds. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3664–3680, 2020.

[39] Xinke Li, Zhirui Chen, Yue Zhao, Zekun Tong, Yabang Zhao, Andrew Lim, and Joey Tianyi Zhou. Pointba: Towards backdoor attacks in 3d point cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16492–16501, 2021.

[40] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31, 2018.

[41] Daizong Liu and Wei Hu. Imperceptible transfer attack and defense on 3d point cloud classification. *arXiv preprint arXiv:2111.10990*, 2021.

[42] Daizong Liu, Wei Hu, and Xin Li. Point cloud attacks in graph spectral domain: When 3d geometry meets graph signal processing. *arXiv preprint arXiv:2207.13326*, 2022.

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

[44] Yongcheng Liu, Bin Fan, Gaofeng Meng, Jiwen Lu, Shiming Xiang, and Chunhong Pan. Densepoint: Learning densely contextual representation for efficient point cloud processing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5239–5248, 2019.

[45] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8895–8904, 2019.

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

[47] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

[48] Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. The application of bayesian methods for seeking the extremum. *Towards global optimization*, 2(117-129):2, 1978.

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

[50] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[51] Yongming Rao, Jiwen Lu, and Jie Zhou. Global-local bidirectional reasoning for unsupervised representation learning of 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5376–5385, 2020.

[52] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.

[53] Yiru Shen, Chen Feng, Yaoqing Yang, and Dong Tian. Mining point cloud local structures by kernel correlation and graph pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4548–4557, 2018.

[54] Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3693–3702, 2017.

[55] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

[56] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015.

[57] Jiachen Sun, Yulong Cao, Qi Alfred Chen, and Z Morley Mao. Towards robust {LiDAR-based} perception in autonomous driving: General black-box adversarial sensor attack and countermeasures. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 877–894, 2020.

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

[59] Jiachen Sun, Yulong Cao, Christopher B Choy, Zhiding Yu, Anima Anandkumar, Zhuoqing Morley Mao, and Chaowei Xiao. Adversarially robust 3d point cloud recognition using self-supervisions. *Advances in Neural Information Processing Systems*, 34:15498–15512, 2021.

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

[61] Abdel Aziz Taha and Allan Hanbury. Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool. *BMC medical imaging*, 15(1):1–28, 2015.

[62] Gusi Te, Wei Hu, Amin Zheng, and Zongming Guo. Rgcnn: Regularized graph cnn for point cloud segmentation. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 746–754, 2018.

[63] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6411–6420, 2019.

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

[65] James Tu, Mengye Ren, Sivabalan Manivasagam, Ming Liang, Bin Yang, Richard Du, Frank Cheng, and Raquel Urtasun. Physically realizable adversarial examples for lidar object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13716–13725, 2020.

[66] Jacob Varley, Chad DeChant, Adam Richardson, Joaquín Ruales, and Peter Allen. Shape completion enabled robotic grasping. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 2442–2447. IEEE, 2017.

[67] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 52–67, 2018.

[68] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2569–2578, 2018.

[69] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019.

[70] Yuxin Wen, Jiehong Lin, Ke Chen, CL Philip Chen, and Kui Jia. Geometry-aware generation of adversarial point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

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

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

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

[74] Chong Xiang, Charles R Qi, and Bo Li. Generating 3d adversarial point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9136–9144, 2019.

[75] Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 915–924, 2021.

[76] Zhen Xiang, David J Miller, Siheng Chen, Xi Li, and George Kesidis. A backdoor attack against 3d point cloud classifiers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7597–7607, 2021.

[77] Mutian Xu, Runyu Ding, Hengshuang Zhao, and Xiaojuan Qi. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3173–3182, 2021.

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

[79] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning object bounding boxes for 3d instance segmentation on point clouds. *Advances in neural information processing systems*, 32, 2019.

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

[81] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. Modeling point clouds with self-attention and gumbel subset sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3323–3332, 2019.

[82] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 206–215, 2018.

[83] Lijia Yu, Yihan Wang, and Xiao-Shan Gao. Adversarial parameter attack on deep neural networks. *arXiv preprint arXiv:2203.10502*, 2022.

[84] Tan Yu, Jingjing Meng, and Junsong Yuan. Multi-view harmonized bilinear network for 3d object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 186–194, 2018.

[85] Xuejing Yuan, Yuxuan Chen, Yue Zhao, Yunhui Long, Xiaokang Liu, Kai Chen, Shengzhi Zhang, Heqing Huang, Xiaofeng Wang, and Carl A Gunter. {CommanderSong}: A systematic approach for practical adversarial voice recognition. In *27th USENIX security symposium (USENIX security 18)*, pages 49–64, 2018.

[86] Xiangyu Yue, Bichen Wu, Sanjit A Seshia, Kurt Keutzer, and Alberto L Sangiovanni-Vincentelli. A lidar point cloud generator: from a virtual world to autonomous driving. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pages 458–464, 2018.

[87] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

[88] Maciej Zamorski, Maciej Zięba, Piotr Klukowski, Rafał Nowak, Karol Kurach, Wojciech Stokowiec, and Tomasz Trzciński. Adversarial autoencoders for compact representations of 3d point clouds. *Computer Vision and Image Understanding*, 193:102921, 2020.

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

[90] Jinlai Zhang, Lyujie Chen, Bo Ouyang, Binbin Liu, Jihong Zhu, Yujin Chen, Yanmei Meng, and Danfeng Wu. Pointcutmix: Regularization strategy for point cloud classification. *Neurocomputing*, 505:58–67, 2022.

[91] Jinlai Zhang, Weiming Li, Shuang Liang, Hao Wang, and Jihong Zhu. Adversarial samples for deep monocular 6d object pose estimation. *arXiv preprint arXiv:2203.00302*, 2022.

[92] Yu Zhang, Gongbo Liang, Tawfiq Salem, and Nathan Jacobs. Defense-pointnet: Protecting pointnet against adversarial attacks. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 5654–5660. IEEE, 2019.

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

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

[95] Boxuan Zhong, He Huang, and Edgar Lobaton. Reliable vision-based grasping target recognition for upper limb prostheses. *IEEE Transactions on Cybernetics*, 2020.

[96] Hang Zhou, Kejiang Chen, Weiming Zhang, Han Fang, Wenbo Zhou, and Nenghai Yu. Dup-net: Denoiser and upsampler network for 3d adversarial point clouds defense. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1961–1970, 2019.

[97] Hang Zhou, Dongdong Chen, Jing Liao, Kejiang Chen, Xiaoyi Dong, Kunlin Liu, Weiming Zhang, Gang Hua, and Nenghai Yu. Lg-gan: Label guided adversarial network for flexible targeted attack of point cloud based deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10356–10365, 2020.

[98] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Wei Li, Yuexin Ma, Hongsheng Li, Ruigang Yang, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar-based perception. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] We do not develop more robust 3D models in this work, which we leave to future work.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Sec. 5
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix A.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The code is included in the supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Sec. 4.1.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Proof

Theorem 1. *Let S and $\tilde{S}$ denote two surfaces of $\mathbb{R}^3$; $\varphi : S \to \tilde{S}$ denote a diffeomorphism that takes a point v in S to point $v' = \varphi(v)$ in $\tilde{S}$; and $K(\cdot)$ be the Gaussian curvature of the points. If $|K(v) - K(v')| < \epsilon$ for any point v, then the surfaces S and $\tilde{S}$ are ϵ-isometric.*

Proof. We will prove Theorem 1 in geodesic polar coordinates [8], which are defined as follows.

Definition 3. *Choose in the tangent plane $T_v(S)$, a system of polar coordinates (ρ, θ) where ρ is the polar radius and θ,$0 < \theta < 2\pi$, is the polar angle, the pole of which is the origin 0 of $T_v(S)$. Set $\exp_v(l) = L$, where $\exp_v$ is the exponential map at v and l is the closed half-line which corresponds to $\theta = 0$. Since $\exp_p : T_v(S) - l \to S - L$ is a diffeomorphism, we may parametrize the points of $S - L$ by the coordinates (ρ, θ), which are called geodesic polar coordinates.*

Lemma 1 (Theorem 4.27 in [34])**.** *Let (ρ, θ) be geodesic polar coordinates. Then the coefficients $E = E(\rho, \theta)$ and $F = F(\rho, \theta)$ of the first fundamental form satisfy the conditions $E = 1, F = 0$.*

Remark 1. *The geometric meaning of the fact that $F = 0$ is that in a normal neighborhood the family of geodesic circles is orthogonal to the family of radial geodesics. This fact is known as the Gauss lemma.*

We can see that the first fundamental form $ds^2 = d\rho^2 + Gd\theta^2$ in a polar system by Lemma 1. By applying the Gauss formula [8],

$$-\frac{1}{\sqrt{EG}}\left\{\left(\frac{(\sqrt{E})_v}{\sqrt{G}}\right)_v + \left(\frac{(\sqrt{G})_u}{\sqrt{E}}\right)_u\right\} = \frac{LN - M^2}{EG} = K. \tag{A.1}$$

Remark 2. *The first fundamental form is the expression of how the surface S inherits the natural inner product of $\mathbb{R}^3$. Geometrically, the first fundamental form allows us to make measurements on the surface (lengths of curves, angles of tangent vectors, areas of regions) without referring back to the ambient space $\mathbb{R}^3$ where the surface lies. The second fundamental form describes the shape of the surface in the ambient space $\mathbb{R}^3$. The Gaussian curvature can be defined by the coefficients of the first fundamental form and the coefficients of the second fundamental form. The Gauss formula and the Mainardi-Codazzi equations reveal the relations between the first and second fundamental forms of a surface. Gauss formula expresses the Gaussian curvature as a function of the coefficients of the first fundamental form and its derivatives, which is also known as Gauss' Theorema Egregium.*

the Gaussian curvature K can be written

$$K = -\frac{(\sqrt{G})_{\rho\rho}}{\sqrt{G}}. \tag{A.2}$$

We state the following lemma to calculate G.

Lemma 2 (Theorem 4.6.3 in [8])**.** *Let (ρ, θ) be geodesic polar coordinates. Then the coefficients $G = G(\rho, \theta)$ of the first fundamental form satisfy the conditions $\lim_{\rho\to 0} G = 0, \lim_{\rho\to 0}(\sqrt{G})_\rho = 1$.*

If $K > 0$, $(\sqrt{G})_{\rho\rho} + K\sqrt{G} = 0$. The general solution is given by

$$\sqrt{G} = f(\theta)\cos(\sqrt{K}\rho) + g(\theta)\sin(\sqrt{K}\rho). \tag{A.3}$$

From the conclusion in Lemma 2, we know $f(\theta) = 0$ and $g(\theta) = \frac{1}{\sqrt{K}}$. We finally have

$$ds^2 = d\rho^2 + \frac{1}{K}\sin^2(\sqrt{K}\rho) \cdot d\theta^2. \tag{A.4}$$

If $K = 0$, $(\sqrt{G})_{\rho\rho} = 0$. Thus,

$$(\sqrt{G})_\rho = f(\theta). \tag{A.5}$$

From the conclusion in Lemma 2, we know $f(\theta) = 1$ and

$$\sqrt{G} = \rho + g(\theta). \tag{A.6}$$

By reusing Lemma 2, we know $g(\theta) = 0$ and $\sqrt{G} = \rho$. We finally have

$$ds^2 = d\rho^2 + \rho^2 \cdot d\theta^2. \tag{A.7}$$

If $K < 0$, $(\sqrt{G})_{\rho\rho} + K\sqrt{G} = 0$. The general solution is given by

$$\sqrt{G} = f(\theta)\cosh(\sqrt{-K}\rho) + g(\theta)\sinh(\sqrt{-K}\rho). \tag{A.8}$$

From the conclusion in Lemma 2, we know $f(\theta) = 0$ and $g(\theta) = \frac{1}{\sqrt{-K}}$. We finally have

$$ds^2 = d\rho^2 + \frac{1}{-K}\sinh^2(\sqrt{-K}\rho) \cdot d\theta^2. \tag{A.9}$$

Thus we can derive

$$\begin{aligned}
&\left|\tfrac{1}{K(v)}\sin^2(\sqrt{K(v)}\rho) - \tfrac{1}{K(v')}\sin^2(\sqrt{K(v')}\rho)\right| \\
&= \left|\tfrac{1}{K(v)K(v')}\left[K(v')\sin^2(\sqrt{K(v)}\rho) - K(v)\sin^2(\sqrt{K(v')}\rho)\right]\right| \\
&= \left|\tfrac{1}{K(v)K(v')}\left[K(v')\left(K(v)\rho^2 - \tfrac{K(v)^2\rho^4}{3}\right) - K(v)\left(K(v')\rho^2 - \tfrac{K(v')^2\rho 4}{3}\right)\right]\right| \\
&= \left|\tfrac{1}{K(v)K(v')} \cdot K(v)K(v') \cdot \tfrac{\rho^4}{3}(K(v') - K(v))\right| < \tfrac{\rho^4}{3}\epsilon.
\end{aligned} \tag{A.10}$$

$$\left|\frac{1}{K(v)}\sin^2(\sqrt{K(v)}\rho) - \rho^2\right| = \left|\frac{1}{K(v)}\left(K(v)\rho^2 - \frac{K(v)^2\rho^4}{3}\right) - \rho^2\right| = \left|-\frac{p4}{3}(K(v) - 0)\right| < \frac{p^4}{3}\epsilon. \tag{A.11}$$

$$\begin{aligned}
&\left|-\tfrac{1}{K(v)}\sinh^2(\sqrt{-K(v)}\rho) + \tfrac{1}{K(v')}\sinh^2(\sqrt{-K(v')}\rho)\right| \\
&= \left|\tfrac{1}{K(v)K(v')}\left[-K(v')\sinh^2(\sqrt{-K(v)}\rho) + K(v)\sinh^2(\sqrt{-K(v')}\rho)\right]\right| \\
&= \left|\tfrac{1}{K(v)K(v')}\left[-K(v')\left(-K(v)\rho^2 + \tfrac{K(v)^2\rho^4}{3}\right) + K(v)\left(-K(v')\rho^2 + \tfrac{K(v')^2\rho^4}{3}\right)\right]\right| \\
&= \left|\tfrac{1}{K(v)K(v')} \cdot K(v)K(v') \cdot \tfrac{\rho^4}{3}(K(v') - K(v))\right| < \tfrac{\rho^4}{3}\epsilon.
\end{aligned} \tag{A.12}$$

$$\left|-\frac{1}{K(v)}\sinh^2(\sqrt{-K(v)}\rho) - \rho^2\right| = \left|-\frac{1}{K(v)}\left(-K(v)\rho^2 + \frac{K(v)^2\rho^4}{3}\right) - \rho^2\right| < \frac{p^4}{3}\epsilon. \tag{A.13}$$

$$\begin{aligned}
&\left|\tfrac{1}{K(v)}\sin^2(\sqrt{K(v)}\rho) + \tfrac{1}{K(v')}\sinh^2(\sqrt{-K(v')}\rho)\right| \\
&=\left|\tfrac{1}{K(v)K(v')}\left[K(v')\sin^2(\sqrt{K(v)}\rho) + K(v)\sinh^2(\sqrt{-K(v')}\rho)\right.\right| \\
&= \left|\tfrac{1}{K(v)K(v')}\left[K(v')\left(K(v)\rho^2 - \tfrac{K(v)^2\rho^4}{3}\right) + K(v)\left(-K(v')\rho^2 + \tfrac{K(v')^2\rho^4}{3}\right)\right]\right| \\
&= \left|\tfrac{1}{K(v)K(v')} \cdot K(v)K(v') \cdot \tfrac{\rho^4}{3}(K(v') - K(v))\right| < \tfrac{\rho^4}{3}\epsilon.
\end{aligned} \tag{A.14}$$

Thus from the condition $|K(v) - K(v')| < \epsilon$ we have

$$|ds^2 - d\tilde{s}^2| < \frac{\rho^4}{3} \cdot d\theta^2 \cdot \epsilon, \tag{A.15}$$

where ds is the element of arc length on a curved surface.

Let $s(C) = n \cdot ds$ and $s(\tilde{C}) = n \cdot d\tilde{s}$, then we have

$$|s(\tilde{C}) - s(C)| = n|ds - d\tilde{s}| < s(C) \cdot \frac{d\theta^2}{ds^2 + dsd\hat{s}} \cdot \frac{\rho^4}{3}\epsilon \tag{A.16}$$

This equation consequently results in

$$\left(1 - \frac{d\theta^2}{ds^2 + dsd\hat{s}} \cdot \frac{\rho^4}{3} \cdot \epsilon\right) s(C) < s(\tilde{C}) < \left(1 + \frac{d\theta^2}{ds^2 + dsd\hat{s}} \cdot \frac{\rho^4}{3} \cdot \epsilon\right) s(C) \tag{A.17}$$

Thus the surfaces S and $\tilde{S}$ are ϵ-isometric. □

B Algorithm

We provide the algorithm to solve problem (6) using Bayesian Optimization in Alg. 1. We choose GP as a surrogate model, which provides a Bayesian posterior distribution to describe the objective function $f(t; D_n) \sim GP(\mu(D_n), \Sigma(D_n, D_n))$, where $f(t; D_n)$ is the modeling of the unknown function $\mathcal{L}_{ce}(t(S(\mathcal{M}_{adv})), y^*)$ and $D_n = \{t_i, \mathcal{L}_{ce}(t_i(S(\mathcal{M}_{adv})), y^*)\}_{i=1}^n$ is the n samples observed so far. μ and Σ are the mean and covariance functions, respectively. We use the expected improvement (EI) [29, 48] acquisition function $\alpha_{EI}(t; D_n) = E_{f(t;D_n)\sim GP(\mu(D_n),\Sigma(D_n,D_n))}\left[\max\left(f(t) - f_{Dn}^+, 0\right)\right]$, which measures the expected improvement of each point with respect to the current best value, where $f_{Dn}^+ = \max_{i\leq n} f(t_i)$ is the best value observed so far. Then, the target function f is sampled by $\arg\max_t \alpha_{EI}(t; D_n)$ to better explore the space of all transformations by selecting the next query point t_{n+1} in the region where the prediction is high and the model is very uncertain.

Algorithm 1 Solve problem (6) using Bayesian Optimization

Require: Observation data D_K, surrogate model $GP(\mu(\cdot), \Sigma(\cdot,\cdot))$, objective function $\mathcal{L}_{ce}(t(S(\mathcal{M}_{adv})), y^*)$, the number of transformations in the MaxOT algorithm S, the number of gradient descent iterations K, and learning rate η_t.

$D_K = \{t_i, \mathcal{L}_{ce}(t_i(S(\mathcal{M}_{adv})), y^*)\}_{i=1}^K$
for $s = 1$ **to** S **do**
  Update the surrogate model:

$$f(t; D_{sK}) \sim GP(\mu(D_{sK}), \Sigma(D_{sK}, D_{sK}));$$

  Calculate the acquisition function:

$$\alpha_{EI}(t; D_{sK}) \leftarrow E_{f(t;D_{sK})\sim GP(\mu(D_{sK}),\Sigma(D_{sK},D_{sK}))}\left[\max\left(f(t) - f_{D_{sK}}^+, 0\right)\right];$$

  Select $t_{sK+1} \leftarrow \arg\max_t \alpha_{EI}(t; D_{sK})$;
  for $k = 1$ **to** K **do**
    Update t_{sK+1+k} with gradient ascent:

$$t_{sK+1+k} \leftarrow t_{sK+k} - \eta_t \cdot \nabla_{t_{sK+k}} \mathcal{L}_{ce}(t_{sK+k}(S(\mathcal{M}_{adv})), y^*);$$

    if $k = K$ **then**
      $t_s^* \leftarrow t_{sK+1+k}$;
    end if
  end for
  $\overline{D}_s \leftarrow \{t_{sK+1+k}, \mathcal{L}_{ce}(t_{sK+1+k}(S(\mathcal{M}_{adv})), y^*)\}_{k=0}^{K-1}$;
  $D_{(s+1)K} \leftarrow D_{sK} \cup \overline{D}_s$;
end for
return $t_s^*, s = 1, \cdots, S$

C Supplementary experimental results

We provide more experimental results in this section. All of the experiments are conducted on NVIDIA 3080 Ti GPUs. The source code is submitted as part of the supplementary material, and will be released after the review process.

C.1 The effects of the penalty parameter λ_1

We study the effects of the penalty parameter λ_1 of the Gaussian curvature consistency regularization term in Sec. 4.6. We adjust the value of λ_1 to perform quantitative and qualitative experiments. The qualitative visualization results in Fig. C.1 demonstrate the irregularity of the 3D objects as λ_1 is turned down. Our default value of $\lambda_1 = 1$ guarantees stealthiness.

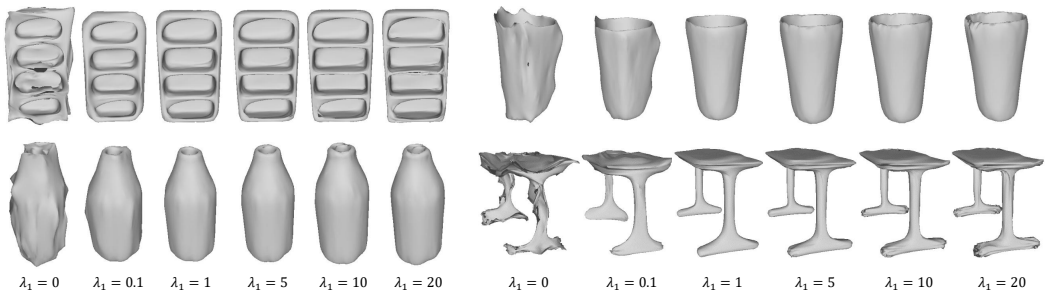


Figure C.1: Qualitative visualization results of the effects of the penalty parameter λ_1. When λ_1 is turned down, the naturalness of the adversarial objects is worse. Our default value of $\lambda_1 = 1$ guarantees stealthiness.

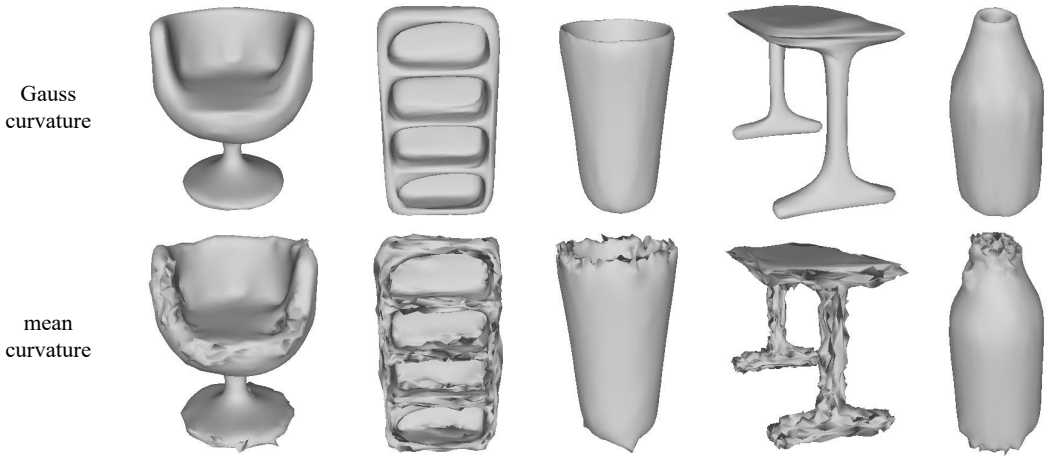


Figure C.2: Qualitative visualization results of Gaussian curvature and mean curvature. The mean curvature consistency regularization term does not guarantee naturalness and smoothness, while Gaussian curvature does.

Table C.1: Quantitative results of attacking different models using EOT, MaxOT w/o BO and MaxOT. The improvement of using Bayesian optimization over random initialization in the MaxOT algorithm is significant.

Model	PointNet			PointNet++			DGCNN		
	ASR	$\mathcal{D}_c$	$\mathcal{D}_g$	ASR	$\mathcal{D}_c$	$\mathcal{D}_g$	ASR	$\mathcal{D}_c$	$\mathcal{D}_g$
EOT	76.20%	0.0074	0.0009	74.28%	0.0094	0.0007	65.72%	0.0068	0.0041
MaxOT w/o BO	80.59%	0.0075	0.0009	81.09%	0.0094	0.0007	70.25%	0.0067	0.0039
MaxOT	82.50%	0.0074	0.0009	84.14%	0.0094	0.0006	72.40%	0.0067	0.0039

C.2 The effectiveness of Bayesian optimization

To investigate the effectiveness of Bayesian optimization to find the initial transformations, we test the attack success rate for randomly selecting the initial transformation in the MaxOT algorithm. As shown in Table C.1, the improvement of using Bayesian optimization over random initialization in the MaxOT algorithm is significant. This confirms the importance of Bayesian optimization methods in our approach.

C.3 The effectiveness of Gaussian curvature

To investigate the effectiveness of Gaussian curvature, we test the mean curvature [8] as a substitute for Gaussian curvature. As shown in Fig. C.2, the mean curvature consistency regularization term does not guarantee naturalness and smoothness. This is because the Gaussian curvature consistency regularization term theoretically provides a sufficient condition (see Theorem 1) to ensure that two surfaces are ϵ-isometric, while the mean curvature consistency regularization term is not.

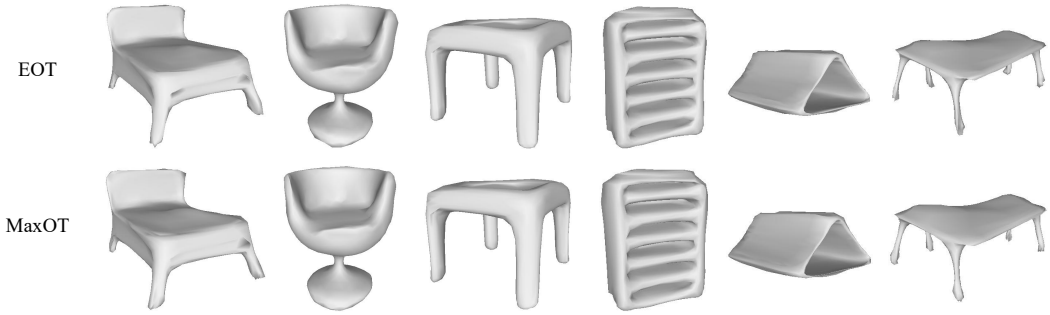


Figure C.3: Qualitative visualization results of MaxOT and EOT. MaxOT algorithm and EOT algorithm generate adversarial objects with the same degree of naturalness.

Table C.2: The effect of the penalty parameter λ_2. When λ_2 is large, the attack success rate decreases rapidly. Our default value of $\lambda_2 = 0.2$ give the best result for balancing the attack success rate and stealthiness.

Model	$\lambda_2 = 0$	$\lambda_2 = 0.02$	$\lambda_2 = 0.2$	$\lambda_2 = 1$	$\lambda_2 = 2$	$\lambda_2 = 4$
PointNet	98.72%	98.67%	98.45%	94.22%	78.03%	69.45%
PointNet++	99.69%	99.66%	99.58%	89.25%	79.82%	72.42%
DGCNN	85.13%	84.82%	84.16%	69.76%	60.48%	52.46%

Table C.3: The effect of the penalty parameter λ_3. When λ_3 is large, the attack success rate decreases rapidly. Our default value of $\lambda_3 = 0.8$ give the best result for balancing the attack success rate and stealthiness.

Model	$\lambda_3 = 0$	$\lambda_3 = 0.08$	$\lambda_3 = 0.8$	$\lambda_3 = 4$	$\lambda_3 = 8$	$\lambda_3 = 16$
PointNet	99.02%	98.86%	98.45%	89.29%	78.42%	63.01%
PointNet++	99.73%	99.70%	99.58%	86.56%	80.47%	71.96%
DGCNN	87.28%	86.89%	84.16%	65.92%	56.92%	42.35%

C.4 Qualitative visualization results of MaxOT and EOT

We compare MaxOT to EOT in Sec. 4.3. The qualitative visualization results of MaxOT and EOT in Fig. C.3 demonstrate the same degree of naturalness, while our proposed MaxOT algorithm outperforms the EOT algorithm in terms of attack success rate and efficiency.

C.5 The effects of the penalty parameter λ_2 and λ_3

To investigate the effect of the penalty parameter λ_2 and λ_3, we adjust the value of λ_2 and λ_3 to perform quantitative and qualitative experiments.

As shown in Table C.2 and Table C.3, when λ_2 or λ_3 is tuned high, the attack success rate decreases rapidly. As shown in Fig. C.4, when λ_2 is adjusted lower, 3D objects show local unevenness and minor self-intersection. When λ_3 is adjusted lower, 3D objects show large areas of self-intersection. Our default value of $\lambda_2 = 0.2$ and $\lambda_3 = 0.8$ give the best result for balancing the attack success rate and stealthiness.

C.6 Evaluation on the Transferability

We conduct experiments on the transfer-based attacks. We generate 3D adversarial examples against one white-box model and evaluate the black-box attack success rates on the other black-box victim models. As shown in Table C.4, our ϵ-ISO attack has much higher success rates than the baselines. This is because our ϵ-ISO retains the geometric properties of the 3D objects well, without local outliers or anomalous deformations. Thus the crafted adversarial examples more transferable across different models.

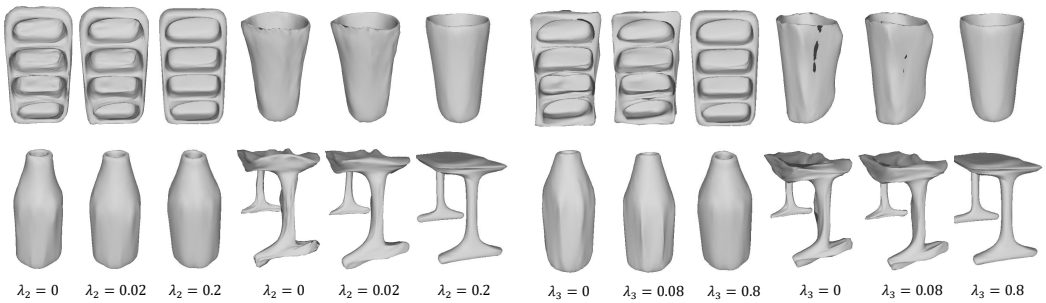


Figure C.4: Qualitative visualization results of the effects of the penalty parameter λ_2 and λ_3. When λ_2 is adjusted lower, 3D objects show local unevenness and minor self-intersection. When λ_3 is adjusted lower, 3D objects show large areas of self-intersection. Our default value of $\lambda_2 = 0.2$ and $\lambda_3 = 0.8$ guarantees stealthiness.

Table C.4: The transfer-based attack success rates on three models by various attacks. Our ϵ-ISO attack has much higher success rates than the baselines.

White-box Target Model	Attacks	Black-box Victim Model		
		PointNet	PointNet++	DGCNN
PointNet	KNN	-	11.1%	10.7%
	$GeoA^3$	-	11.5%	2.5%
	ϵ-ISO	-	47.6%	35.8%
PointNet++	KNN	6.4%	-	7.9%
	$GeoA^3$	9.4%	-	19.7%
	ϵ-ISO	32.9%	-	51.2%
DGCNN	KNN	7.2%	32.2%	-
	$GeoA^3$	12.4%	24.2%	-
	ϵ-ISO	55.4%	62.7%	-

D Proof of $\boldsymbol{n}_u \wedge \boldsymbol{n}_v = K\boldsymbol{r}_u \wedge \boldsymbol{r}_v$

The Weingarten map $\mathcal{W}$ is defined by

$$\mathcal{W}: \begin{array}{ll} T_P S & \to T_P S \\ \boldsymbol{v} = \lambda \boldsymbol{r}_u + \mu \boldsymbol{r}_v & \to \mathcal{W}(\boldsymbol{v}) = -\left(\lambda \boldsymbol{n}_u + \mu \boldsymbol{n}_v\right) \end{array} \tag{D.1}$$

Thus, $\mathcal{W}(\boldsymbol{r}_u) = -\boldsymbol{n}_u, \mathcal{W}(\boldsymbol{r}_v) = -\boldsymbol{n}_v$. The coefficient matrix of the Weingarten map $\mathcal{W}$ is $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$, where $\mathcal{W}(\boldsymbol{r}_u) = -\boldsymbol{n}_u = a\boldsymbol{r}_u + b\boldsymbol{r}_v, \mathcal{W}(\boldsymbol{r}_v) = -\boldsymbol{n}_v = c\boldsymbol{r}_u + d\boldsymbol{r}_v$. Take the dot product of each of these equations with $\boldsymbol{r}_u$ and $\boldsymbol{r}_v$. This gives

$$\begin{array}{l} \langle -\boldsymbol{n}_u, \boldsymbol{r}_u \rangle = a \langle \boldsymbol{r}_u, \boldsymbol{r}_u \rangle + b \langle \boldsymbol{r}_v, \boldsymbol{r}_u \rangle \\ \langle -\boldsymbol{n}_u, \boldsymbol{r}_v \rangle = a \langle \boldsymbol{r}_u, \boldsymbol{r}_v \rangle + b \langle \boldsymbol{r}_v, \boldsymbol{r}_v \rangle \\ \langle -\boldsymbol{n}_v, \boldsymbol{r}_u \rangle = c \langle \boldsymbol{r}_u, \boldsymbol{r}_u \rangle + d \langle \boldsymbol{r}_v, \boldsymbol{r}_u \rangle \\ \langle -\boldsymbol{n}_v, \boldsymbol{r}_v \rangle = c \langle \boldsymbol{r}_u, \boldsymbol{r}_v \rangle + d \langle \boldsymbol{r}_v, \boldsymbol{r}_v \rangle \end{array} \tag{D.2}$$

Since $\boldsymbol{r}_u$ and $\boldsymbol{r}_v$ are tangent vectors to the surface, $\langle \boldsymbol{r}_u, \boldsymbol{n} \rangle = 0, \langle \boldsymbol{r}_v, \boldsymbol{n} \rangle = 0$. Differentiating these equations with respect to u and v gives

$$\begin{array}{l} \langle \boldsymbol{r}_{uu}, \boldsymbol{n} \rangle = \langle \boldsymbol{r}_u, -\boldsymbol{n}_u \rangle \\ \langle \boldsymbol{r}_{uv}, \boldsymbol{n} \rangle = \langle \boldsymbol{r}_u, -\boldsymbol{n}_v \rangle = \langle \boldsymbol{r}_v, -\boldsymbol{n}_u \rangle \\ \langle \boldsymbol{r}_{vv}, \boldsymbol{n} \rangle = \langle \boldsymbol{r}_v, -\boldsymbol{n}_v \rangle \end{array} \tag{D.3}$$

This gives

$$\begin{array}{ll} L = aE + bF, & M = cE + dF, \\ M = aF + bG, & N = cF + dG. \end{array} \tag{D.4}$$

These four scalar equations are equivalent to the single matrix equation

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} L & M \\ M & N \end{bmatrix} \begin{bmatrix} E & F \\ F & G \end{bmatrix}^{-1} \tag{D.5}$$

Thus we can derive

$$
\begin{aligned}
\boldsymbol{n}_u \wedge \boldsymbol{n}_v &= (a\boldsymbol{r}_u + b\boldsymbol{r}_v) \wedge (c\boldsymbol{r}_u + d\boldsymbol{r}_v) \\
&= (ad - bc)\boldsymbol{r}_u \wedge \boldsymbol{r}_v \\
&= \det\left(\begin{bmatrix} L & M \\ M & N \end{bmatrix}\begin{bmatrix} E & F \\ F & G \end{bmatrix}^{-1}\right)\boldsymbol{r}_u \wedge \boldsymbol{r}_v \\
&= \frac{\det\left(\begin{bmatrix} L & M \\ M & N \end{bmatrix}\right)}{\det\left(\begin{bmatrix} E & F \\ F & G \end{bmatrix}\right)}\boldsymbol{r}_u \wedge \boldsymbol{r}_v \\
&= \frac{LN - M^2}{EG - F^2}\boldsymbol{r}_u \wedge \boldsymbol{r}_v \\
&= K\boldsymbol{r}_u \wedge \boldsymbol{r}_v
\end{aligned}
\tag{D.6}
$$