Manifold Constraints for Imperceptible Adversarial Attacks on Point Clouds

Keke Tang[1]*, Xu He[1]*, Weilong Peng[2]†, Jianpeng Wu[1], Yawen Shi[1], Daizong Liu[3], Pan Zhou[4], Wenping Wang[5], Zhihong Tian[1]

[1]Cyberspace Institute of Advanced Technology, Guangzhou University
[2]School of Computer Science and Cyber Engineering, Guangzhou University
[3]Wangxuan Institute of Computer Technology, Peking University
[4]Hubei Engineering Research Center on Big Data Security, School of Cyber Science and Engineering, Huazhong University of Science and Technology
[5]Department of Computer Science and Engineering, Texas A&M University
tangbohutbh@gmail.com, heexu976@gmail.com, wlpeng@tju.edu.cn, lesswu666@gmail.com, shiyawen666@gmail.com, dzliu@hust.edu.cn, zhoupannewton@gmail.com, wenping@cs.hku.hk, tianzhihong@gzhu.edu.cn

Abstract

Adversarial attacks on 3D point clouds often exhibit unsatisfactory imperceptibility, which primarily stems from the disregard for manifold-aware distortion, i.e., distortion of the underlying 2-manifold surfaces. In this paper, we develop novel manifold constraints to reduce such distortion, aiming to enhance the imperceptibility of adversarial attacks on 3D point clouds. Specifically, we construct a bijective manifold mapping between point clouds and a simple parameter shape using an invertible auto-encoder. Consequently, manifold-aware distortion during attacks can be captured within the parameter space. By enforcing manifold constraints that preserve local properties of the parameter shape, manifold-aware distortion is effectively mitigated, ultimately leading to enhanced imperceptibility. Extensive experiments demonstrate that integrating manifold constraints into conventional adversarial attack solutions yields superior imperceptibility, outperforming the state-of-the-art methods.

Introduction

With the advancement of deep learning techniques (Tang et al. 2022b) and the accessibility of affordable depth-sensing devices, 3D point cloud perception utilizing deep neural networks (DNNs) has emerged as the go-to solution (Guo et al. 2020). However, numerous recent studies have revealed that DNN classifiers are vulnerable to adversarial attacks (Xiang, Qi, and Li 2019; Liu, Yu, and Su 2019), i.e., imperceptible perturbations on the input point clouds can lead to erroneous predictions, hindering their deployment in real-world scenarios. Consequently, investigating adversarial attacks on DNN classifiers for 3D point clouds is a crucial step, as it lays the groundwork for the assessment and enhancement of their adversarial robustness.

Typical methodologies to ensure the imperceptibility of adversarial attacks on 3D point clouds employ metrics such as the l_2-norm, Chamfer distance, and Hausdorff distance to constrain the perturbation. More recent studies have sought to further reduce the distortion by limiting the change of curvature (Wen et al. 2022), guiding the perturbation along the normal direction (Liu and Hu 2023; Tang et al. 2022c) and along the tangential direction (Huang et al. 2022), etc. Despite these advances, the generated adversarial point clouds do not yet achieve the desired level of imperceptibility, often exhibiting noticeable outliers or deformations in shape.

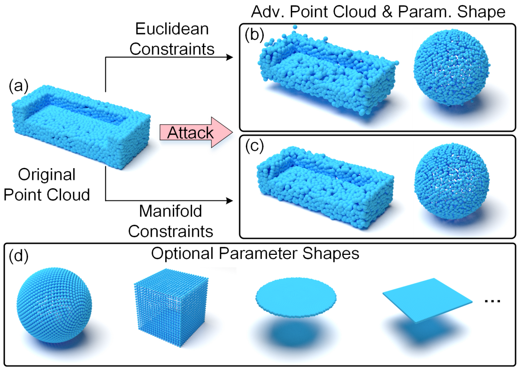

Figure 1: Illustration of adversarial attacks: (a) original point cloud; (b, c) adversarial point clouds generated under Euclidean and manifold constraints respectively, and their corresponding parameter shapes obtained using our parameterization-like operation; (d) optional parameter shapes. Note that the SOFA point clouds in (b, c) are misclassified by PointNet as BATHTUB after the IFGM attack.

While the aforementioned solutions successfully limit the distortion of 3D point clouds from a Euclidean perspective, they overlook the common assumption that these point clouds are generally sampled from 2-manifold surfaces (Spanier 1989). As a result, they have yet to apply any constraints in this aspect, leading to manifold-aware distortion. We argue that the *persistent manifold-aware distortion* is the primary cause of unsatisfactory imperceptibility.

In this paper, we propose novel manifold constraints aimed at minimizing the manifold-aware distortion during attacks. Specifically, we employ an invertible auto-encoder

*These authors contributed equally.
†Corresponding author.

to create a bijective mapping that transforms original point clouds into simple-shaped parameter shapes, see Fig. 1. This process simplifies the capture of manifold-aware distortion. Next, we quantify the distortion based on changes in the local properties of the parameter shapes, i.e., distances and angles, and employ these measures as the manifold constraints. These constraints can serve as an add-on that can be seamlessly integrated into existing adversarial attack solutions, e.g., IFGM (Dong et al. 2020), to restrict modifications to the underlying 2-manifold surfaces of 3D point clouds, thereby enhancing the imperceptibility. We validate the effectiveness of our manifold constraints on various adversarial attack solutions in attacking common DNN classifiers for 3D point clouds. Extensive experimental results show that the generated adversarial point clouds are significantly more imperceptible after applying the manifold constraints, outperforming those generated by state-of-the-art methods.

Overall, our contribution is summarized as follows:

- We are the first to attribute the inadequate imperceptibility of adversarial attacks on 3D point clouds to the neglect of the manifold-aware distortion.
- We develop novel manifold constraints that restrict the distortion of the parameter shape, which is bijectively mapped to the original point clouds, during attacks.
- We show by experiments that adversarial attacks under manifold constraints achieve superior performance in terms of imperceptibility.

Related Work

Adversarial Attacks on 3D Point Clouds. Adversarial attacks, aimed at generating samples that can mislead target networks (Zhu et al. 2023a,b; Tang et al. 2023a; Li et al. 2023), originated in 2D image classification and have been successfully extended to 3D point clouds. Within the realm of 3D, these attacks are categorized into three types: addition-based, introducing independent points to induce errors (Xiang, Qi, and Li 2019); deletion-based, involving the removal of critical points to affect classification (Zheng et al. 2019; Yang et al. 2019; Wicker and Kwiatkowska 2019; Zhang et al. 2021); and perturbation-based, which involves altering existing points to facilitate attacks (Xiang, Qi, and Li 2019; Zhao et al. 2020; Kim et al. 2021). This paper specifically focuses on perturbation-based methods.

Xiang, Qi, and Li (2019) and Liu, Yu, and Su (2019) pioneered perturbation-based 3D adversarial attacks by extending C&W attack (Carlini and Wagner 2017) and FGSM (Goodfellow, Shlens, and Szegedy 2015). Zhao et al. (2020) introduced an isometric transformation attack using simple rotations instead of altering individual points. Kim et al. (2021) aimed to perturb only a minimal subset rather than all points. Generative solutions extend beyond point-coordinate perturbations, with approaches such as noise injection into latent features (Lee et al. 2020) or the use of generative adversarial networks (GANs) (Zhou et al. 2020). To facilitate imperceptibility, most adversarial attack solutions apply intentional constraints to restrict the perturbation.

Constraints for Imperceptible Adversarial Attacks. The most commonly used constraints to ensure the imperceptibility of adversarial attacks on point clouds include restrictions on the l_2-norm, Chamfer distance, and Hausdorff distance between the original and adversarial point clouds (Xiang, Qi, and Li 2019; Liu, Yu, and Su 2019; Zhou et al. 2020). Beyond these standard constraints, GeoA^3 (Wen et al. 2022) maintains local curvatures after the attack. Liu and Hu (2023) constrained the perturbation direction of each point to its normal vector, and Huang et al. (2022) directed the perturbation along the tangent plane. These constraints were adaptively relaxed by Tang et al. (2023b) to be near the normal or tangential direction. Our approach also emphasizes the use of constraints to render perturbations imperceptible; however, unlike the aforementioned methods that apply geometric constraints in Euclidean space, we adopt a novel manifold perspective to address the issue.

Deep 3D Point Cloud Classification. Deep learning techniques for 3D point cloud classification have evolved significantly (Bronstein et al. 2017; Tang et al. 2022a; Tang, Song, and Chen 2016; Chen et al. 2022), moving from initial voxel grid methods (Maturana and Scherer 2015) to advanced direct processing of points (Qi et al. 2017; Wu, Qi, and Fuxin 2019). We aim to attack these classifiers imperceptibly.

Manifold Concept. The concept of a manifold in mathematics denotes a topological space resembling Euclidean space near each point (Lee and Lee 2012). In Tang et al. (2023c)'s strategy, manifold mapping via injective mapping is utilized for distortion in attacks. Our method, in contrast, employs bijective mapping to limit distortion, thereby improving the imperceptibility of our attacks.

Problem Formulation

Preliminary on Adversarial Attacks. Given a point cloud $P \in R^{n\times 3}$ sampled from the object surface $\mathcal{S}$ and its label $y \in Z$, adversarial attack aims to mislead a 3D deep classification model $\mathcal{F}$ by feeding an adversarial point cloud $P^{'}$ via applying an intentionally designed perturbation σ on P , such that the model makes an error prediction. Formally, the perturbation σ can be obtained by solving the below equation, e.g., via gradient descent,

$$\min_{\sigma} L_{mis}(\mathcal{F}, P+\sigma, y) + \lambda_1 D(P, P+\sigma), \tag{1}$$

where $L_{mis}(\cdot,\cdot,\cdot)$ is the loss to promote misclassification, e.g., the negation of cross-entropy loss, $D(\cdot,\cdot)$ is the constraints on distortion to facilitate imperceptibility, and λ_1 is a weighting parameter. Here, our focus is primarily on untargeted attacks, and targeted attacks can be readily facilitated.

Some widely-adopted options of $D(\cdot,\cdot)$ include the l_2-norm, Chamfer distance, Hausdorff distance, curvature, and perturbation direction. Since without any constraints from a manifold perspective, the underlying 2-manifold surface $\mathcal{S}$ can still experience significant distortion.

Our Solution against Manifold-aware Distortion. Since $\mathcal{S}$ is typically too complex to measure distortion, we opt for another simpler shape as a bridge. Suppose there exists a bijective mapping $\mathcal{M}$ between a simple shape $\mathcal{U}$, which is also 2-manifold, and the surface $\mathcal{S}$ (Hormann, Polthier, and Sheffer 2008):

$$\mathcal{M} : \mathcal{U} \longleftrightarrow \mathcal{S}, \tag{2}$$

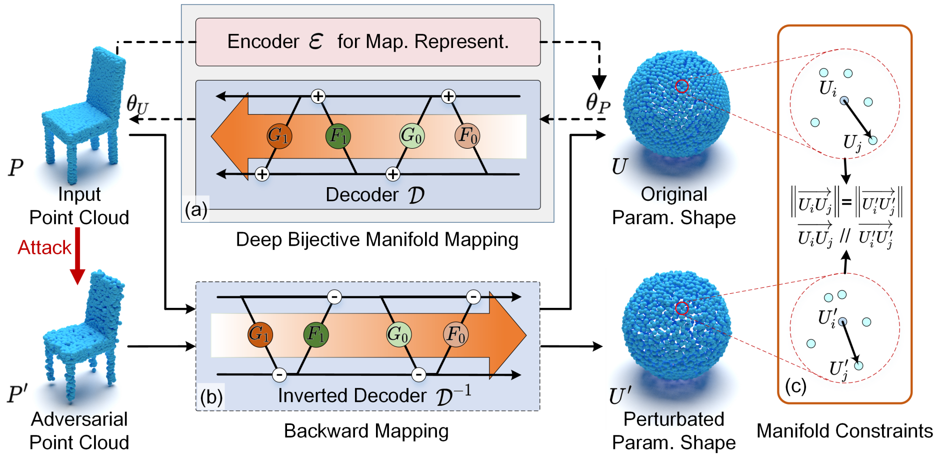

Figure 2: Illustration of imperceptible adversarial attacks under manifold constraints: (a) deep bijective manifold mapping between input point cloud and parameter shape; (b) backward mapping to transform perturbation into the parameter space for capturing manifold-aware distortion; (c) enforcement of two manifold constraints to minimize the distortion.

we thus could measure the distortion of $\mathcal{U}$ instead,

$$D_m(\mathcal{U},\mathcal{U}') = D_m(\mathcal{M}^{-1}(\mathcal{S}), \mathcal{M}^{-1}(\mathcal{S}')), \tag{3}$$

where $\mathcal{S}'$ is the perturbed version of $\mathcal{S}$ after attack, $\mathcal{M}^{-1}$ is the inverted mapping of $\mathcal{M}$, and $\mathcal{U}'$ is the perturbed version of $\mathcal{U}$. Since the mapping process closely resembles the concept of parameterization (Hormann, Polthier, and Sheffer 2008), we refer to $\mathcal{U}$ as the *parameter shape*.

By introducing Eqn. (3) as the manifold constraints into Eqn. (1), we obtain the updated objective,

$$\min_{\sigma} L_{mis}(\mathcal{F}, P+\sigma, y) + \lambda_1 D(P, P+\sigma) + \lambda_2 D_m(\mathcal{U},\mathcal{U}'), \tag{4}$$

where λ_2 is a weighting parameter. Since $\mathcal{S}'$ is additionally constrained in the parameter space, the newly generated adversarial point cloud P' is expected to have improved imperceptibility.

Method

In this section, we will describe how to represent the bijective manifold mapping for 3D point clouds using DNNs, and then outline the manifold constraints under the mapping, along with its usage for imperceptible adversarial attacks. Please refer to Fig. 2 for demonstration.

Deep Bijective Manifold Mapping

We employ an invertible auto-encoder to realize the representation of $\mathcal{M}$. Specifically, given a point cloud P as input, the encoder $\mathcal{E}$ outputs the deep mapping representation θ_P,

$$\theta_P = \mathcal{E}(P), \tag{5}$$

and then the invertible decoder $\mathcal{D}$ reconstructs P by manipulating the parameter shape U guided by θ_P,

$$(\theta_P, U)\mathcal{D}^{-1}\overset{\mathcal{D}}{\rightleftharpoons}(\theta_U, P), \tag{6}$$

where U is a discretized version of $\mathcal{U}$, i.e., the point clouds that constitute $\mathcal{U}$, and θ_U is a deep representation that is related to U. In particular, $\mathcal{D}$ is implemented using invertible neural networks (Behrmann et al. 2019; Gomez et al. 2017) to enable bijective mapping between U and P, see Fig. 2.

Forward Mapping. The decoder $\mathcal{D}$, which consists F_0, G_0, F_1, G_1, maps (θ_P, U) to (θ_U, P) via,

$$Y_1 = \theta_P + F_0(U), \quad Y_2 = U + G_0(Y_1),$$
$$\theta_U = Y_1 + F_1(Y_2), \quad P = Y_2 + G_1(\theta_U).$$

Backward Mapping. The inverted decoder $\mathcal{D}^{-1}$ maps (θ_U, P) to (θ_P, U) via,

$$Y_2 = P - G_1(\theta_U), \quad Y_1 = \theta_U - F_1(Y_2),$$
$$U = Y_2 - G_0(Y_1), \quad \theta_P = Y_1 - F_0(U).$$

By learning the above bijective manifold mapping, we can effectively capture the manifold-aware distortion on the parameter shapes by transforming the Euclidean distortion present in the original point clouds.

Imperceptible Adversarial Attacks under Manifold Constraints

To avoid large manifold-aware distortion of the generated adversarial point cloud P', we transform P' in the Euclidean space to U' in the parameter space by backward mapping,

$$U' = P' - G_1(\theta_U) - G_0(\theta_U - F_1(P' - G_1(\theta_U))), \tag{7}$$

and then apply our manifold constraints between U' and U.

Manifold Constraints. To constrain the distortion of the parameter shape U, we enforce the perturbation to maintain local properties. Specifically, given a local shape with U_i as the center, i.e., the i-th point of U, and U_i's k-nearest neighbors $\{U_{i_1}, \ldots, U_{i_k}\}$, we consider the constraints in two aspects, i.e., distance and angle.

First, we expect the relative distance between each point pair to be maintained after the attack,

$$\mathrm{dist}(U_i, U_i') = \frac{1}{k}\sum_{j\in\{i_1,\ldots,i_k\}} \left| \|U_i - U_j\|_2 - \|U_i' - U_j'\|_2 \right|, \tag{8}$$

where $U_i^{'}$ is the corresponding point of U_i after applying perturbation, and $\| \cdot \|_2$ indicates the l_2-norm.

Second, we expect the direction between each point pair to be maintained after the attack,

$$\text{angle}(U_i, U_i^{'}) = \frac{1}{k} \sum_{j \in \{i_1, \ldots, i_k\}} \left(1 - \frac{(U_i - U_j) \cdot (U_i^{'} - U_j^{'})}{(\|U_i - U_j\|_2 \|U_i^{'} - U_j^{'}\|_2)} \right). \quad (9)$$

Finally, the manifold constraints between U and $U^{'}$ are defined as the weighted sum of the two losses,

$$D_m(U, U^{'}) = \frac{1}{n} \sum_{i \in \{1, \ldots, n\}} \text{dist}(U_i, U_i^{'}) + \beta \text{angle}(U_i, U_i^{'}), \quad (10)$$

where β is a weighting parameter.

Usage for Imperceptible Adversarial Attacks. The final objective for imperceptible adversarial attacks can be formulated by replacing $D_m(\mathcal{U}, \mathcal{U}^{'})$ in Eqn. (4) with $D_m(U, U^{'})$,

$$\min_{\sigma} L_{mis}(\mathcal{F}, P + \sigma, y) + \lambda_1 D(P, P + \sigma) + \lambda_2 D_m(U, U^{'}). \quad (11)$$

Therefore, imperceptible adversarial perturbation σ can be obtained by solving the above equation.

Note that, our manifold constraints can be seamlessly integrated into many common adversarial attack solutions, e.g., IFGM, to obtain enhanced imperceptibility. Moreover, our method allows different options for the parameter shape U, e.g., a square plate, a circular plate, a sphere, and a cube.

Experiments

Experimental Setup

Implementation. We implement the invertible auto-encoder using PyTorch (Paszke et al. 2019). The encoder is identical to that in (Yang et al. 2018), while the decoder is four-component invertible networks, each with two layers of MLPs. The mapping representation θ_P is a codeword of size 1×512, and the parameter shapes are represented with discrete points sampled in a relatively uniform manner. In particular, the square plate-based parameter shape is a 45×45 point grid in the range of $[-1.0, 1.0]$, and the sphere-based parameter shape consists of 2048 points uniformly sampled on a unit sphere centered at $(0.0, 0.0, 0.0)$ with a radius of 1.0. For calculating the two manifold constraints, we select 10-nearest neighbors for each point. For hyperparameters, we set $\lambda_1 = 1.0$, $\lambda_2 = 0.1$, and $\beta = 1.0$. We pre-train the invertible auto-encoder under the Chamfer distance constraint for a total of 2000 epochs. All experiments are conducted on a workstation with one NVIDIA RTX 3090 GPU.

Datasets. We utilize two public datasets for evaluation: ModelNet40 (Wu et al. 2015) and ShapeNet Part (Chang et al. 2015). Particularly, we randomly sample 2,048 points from each point cloud.

Our Attack Solutions. We incorporate our manifold constraints into two classic attack methods, i.e., IFGM (Dong et al. 2020) and C&W (Xiang, Qi, and Li 2019), with two representative parameter shapes, i.e., square **p**late and **s**phere. This results in four unique configurations: IFGM-P, IFGM-S, C&W-P, and C&W-S.

Baseline Attack Methods. We choose 10 baseline solutions: Drop-600 (Zheng et al. 2019), which drops the 600 most critical points; FGM, PGD, and IFGM (Dong et al. 2020), which are perturbation-based methods utilizing gradient; C&W (Xiang, Qi, and Li 2019) and AdvPC (Hamdi et al. 2020), which are perturbation-based method employing optimization techniques; LG-GAN (Zhou et al. 2020), a generative-based approach; GeoA^3 (Wen et al. 2022), SI-Adv (Huang et al. 2022) and ITA (Liu and Hu 2023), whose focus is also imperceptibility. Regarding the selection of non-manifold constraint $D(\cdot, \cdot)$, we adhere to the methodologies described in their original papers.

Victim Models. We choose three common DNN classifiers to attack, i.e., PointNet (Qi et al. 2017), DGCNN (Wang et al. 2019) and PointConv (Wu, Qi, and Fuxin 2019). We train these models according to their original papers.

Evaluation Setting and Metrics. We evaluate adversarial attack methods on various imperceptibility metrics, e.g., perturbation size measured using Chamfer distance (CD) and Hausdorff distance (HD), under the maximal adversarialness setting (Liu and Hu 2023), where each method is configured to achieve its highest achievable attack success rate (ASR) within 100 iterations. We also adopt the novel manifold-aware distortion metric to measure the perturbation on the parameter shape using l_2-norm (l_2), CD, and HD. Note that all l_2 and HD values in this paper are scaled by 10^{-4}, while CD values are scaled by 10^{-3}, for clarity.

Performance on Adversarial Attacks

ASR and Imperceptibility. We provide the ASR and imperceptibility results of various adversarial attack methods tested on ModelNet40 and ShapeNet Part in Tab. 1. It is evident that most adversarial attack solutions can achieve high ASR, with iterative-based ones notably reaching 100%. In terms of imperceptibility, the majority of methods exhibit low CD and HD values. By imposing manifold constraints, both IFGM and C&W demonstrate even lower values. As for the manifold constraints, applying a sphere-based parameter shape yields greater improvements compared to applying a square plate-based one. In particular, IFGM-S stands out, achieving the most superior imperceptibility performance.

We also visualize adversarial point clouds generated by various adversarial attack methods aimed at fooling PointNet in Fig. 3. It can be observed that the adversarial point clouds generated by all baseline methods exhibit noticeable outliers. By integrating manifold constraints, both IFGM-S and C&W-S show a significant reduction in outliers, with IFGM-S demonstrating the best performance.

Manifold-aware Distortion. Given that the distortion from the Euclidean perspective has already been minimized, the distinctions among various methods are less significant. Therefore, we backward-map adversarial point clouds generated by different adversarial attack approaches to the parameter shape using the pretrained manifold mapping, and then quantitatively analyze the manifold-aware distortion. The results in Tab. 2 reveal that our methods consistently surpass the state-of-the-art methods by a considerable margin, irrespective of whether a square plate or a sphere is employed as the parameter shape. Remarkably, IFGM-S

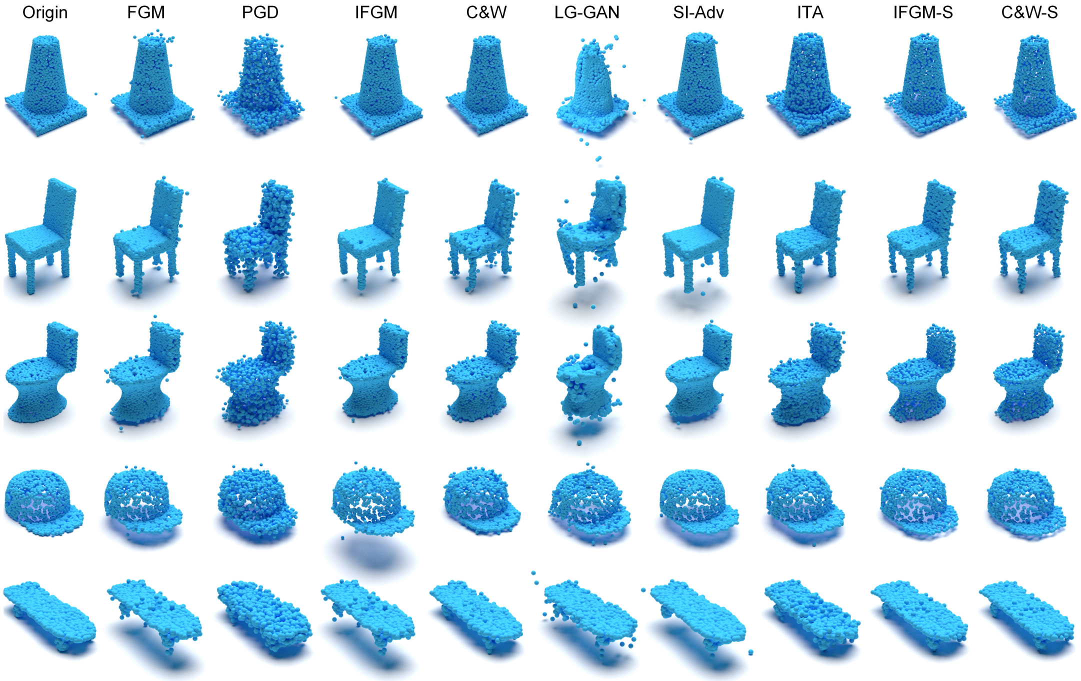

Figure 3: Visualization of original point clouds and the corresponding adversarial point clouds generated by different attack methods for attacking PointNet. The predicted categories before and after attack from top to bottom are: CONE → RANGE HOOD; CHAIR → TOILET; TOILET → CHAIR; CAP → LAMP; SKATEBOARD → TABLE.

Attack	ModelNet40									ShapeNet Part								
	PointNet			DGCNN			PointConv			PointNet			DGCNN			PointConv		
	ASR	CD	HD	ASR	CD	HD	ASR	CD	HD	ASR	CD	HD	ASR	CD	HD	ASR	CD	HD
Drop-600	0.85	37.98	81.52	0.44	51.63	96.73	0.29	38.79	78.61	0.45	49.83	104.20	0.23	69.61	121.50	0.30	45.88	95.78
FGM	0.75	1.37	31.19	0.27	2.17	13.37	0.25	1.94	19.73	0.33	8.09	40.14	0.12	**6.06**	38.16	0.16	10.31	43.90
PGD	1.00	18.47	30.18	1.00	18.63	20.16	1.00	18.50	12.87	1.00	17.27	43.36	0.85	18.80	62.67	1.00	17.76	44.22
AdvPC	1.00	12.60	34.45	1.00	11.77	18.62	1.00	9.88	12.65	1.00	19.25	54.25	1.00	29.18	64.92	1.00	15.42	40.41
IFGM	1.00	0.96	22.62	1.00	0.68	4.67	1.00	0.54	12.19	1.00	5.87	41.10	0.86	8.20	57.32	1.00	4.93	38.50
C&W	1.00	3.48	6.22	1.00	8.28	4.51	1.00	5.83	7.68	1.00	4.22	15.70	1.00	15.29	22.40	1.00	10.71	17.01
LG-GAN	1.00	10.87	51.19	1.00	9.94	72.51	1.00	8.23	41.73	0.98	54.76	112.21	0.75	83.02	150.01	0.61	67.82	115.61
GeoA3	1.00	4.50	4.20	1.00	8.91	5.11	1.00	10.09	4.88	1.00	14.99	22.42	1.00	35.65	45.05	1.00	27.97	21.33
SI-Adv	1.00	1.61	20.46	1.00	1.08	5.46	1.00	0.98	11.57	0.96	9.93	43.32	0.95	8.73	41.81	0.95	7.74	38.71
ITA	1.00	1.08	2.45	1.00	1.89	8.42	1.00	2.39	1.97	1.00	3.71	17.06	1.00	11.29	19.51	1.00	9.78	12.47
C&W-P	1.00	1.30	2.25	1.00	5.83	2.81	1.00	3.75	2.51	1.00	5.59	8.32	1.00	13.10	16.64	1.00	8.37	7.96
C&W-S	1.00	1.33	2.05	1.00	6.17	2.84	1.00	2.75	1.86	1.00	2.97	**8.21**	1.00	12.24	16.06	1.00	8.19	7.14
IFGM-P	1.00	0.40	1.98	1.00	0.62	1.01	1.00	0.42	0.63	1.00	**0.93**	12.30	1.00	6.45	14.29	1.00	**1.72**	6.41
IFGM-S	1.00	**0.37**	**1.68**	1.00	**0.57**	**0.83**	1.00	**0.39**	**0.49**	1.00	0.95	12.21	1.00	6.44	**14.27**	1.00	3.20	**4.46**

Table 1: Comparison on the perturbation sizes required by different methods to reach their highest achievable ASR. The evaluation is conducted across different DNN classifiers on ModelNet40 and ShapeNet Part.

achieves the most superior performance across all cases. Additionally, we provide a visualization of the manifold-aware distortion brought by IFGM and IFGM-S in Fig. 4. Clearly, by introducing manifold constraints, the adversarial point clouds generated by IFGM-S undergo significantly less manifold-aware distortion, validating its usefulness.

Analysis on Bijective Manifold Mapping

Visualization of Learned Bijective Mapping. To demonstrate the feasibility of our invertible auto-encoder-based manifold mapping, we visualize the original point clouds and four distinct parameter shapes (sphere, cube, square plate, and circular plate), along with their bijectively mapped equivalents in Fig. 5. It is evident that the bijective manifold mappings are successfully learned. In particular, most of these mappings preserve local continuity, especially when 3D spheres and cubes are used as the parameter shapes. These findings substantiate our methodology for constructing the bijective manifold mapping, thereby facilitating the enforcement of manifold constraints.

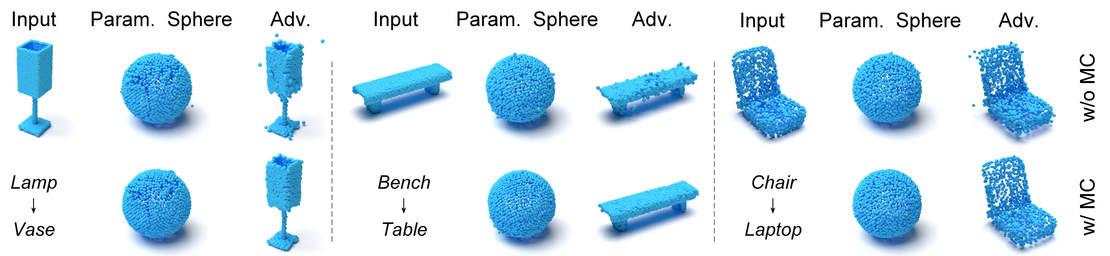

Figure 4: Visualization of adversarial point clouds and the corresponding bijective mapped parameter spheres with and without using manifold constraints (MC) for IFGM. The victim classifier is PointNet.

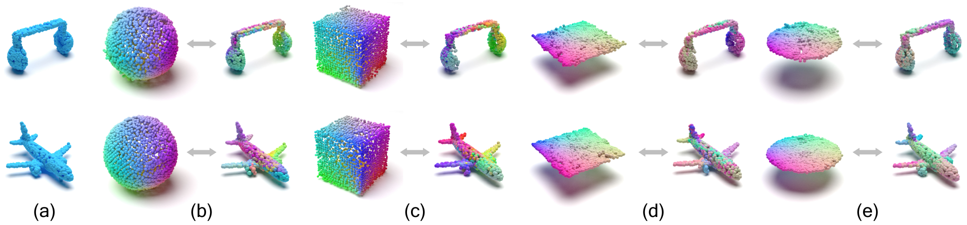

Figure 5: Bijective manifold mapping results: (a) original point clouds; learned manifold mapping pairs with four different parameter shapes including (b) sphere, (c) cube, (d) square plate, and (e) circular plate. We use a color gradient to illustrate the correspondence in the mapping.

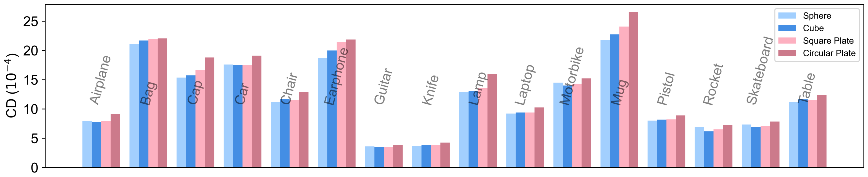

Figure 6: Comparison on the category-wise reconstruction error for the bijective manifold mapping learned using four different parameter shapes on the ShapeNet Part dataset.

Effects of Different Parameter Shapes. We conduct a quantitative evaluation of the manifold mapping with respect to different parameter shapes on ShapeNet Part. Specifically, we reconstruct the original point clouds by inverting the mapping for four different types of parameter shapes. Then, we compute the average CD value between the reconstructed and the original point clouds. The category-wise statistics are visualized in Fig. 6. Upon observation, it is noticeable that the manifold mapping with a 3D closed parameter shape, such as a sphere or a cube, typically outperforms the one utilizing 2D shapes, like a square plate or a circular plate. This disparity becomes particularly pronounced for objects with elongated components, such as EARPHONE and MUG. Intriguingly, the performances exhibit a similar trend between the two 3D shapes, i.e., the sphere and the cube, and likewise between the two 2D shapes, i.e., the square plate and the circular plate.

Ablation Studies and Other Analysis

Importance of Invertibility. We further investigate the significance of invertibility in our approach. Without invertibility, we can only perturb the parameter shape, indirectly influencing the point clouds, while simultaneously enforcing manifold constraints on the parameter shape. The results presented in Tab. 3 reveal that, although the ASR is comparable between methods with and without invertibility, our invertible methods yield smaller perturbation sizes on the point clouds. This finding underscores the importance of invertibility in our approach.

Effects of Different Manifold Constraints. We further investigate the impact of employing different terms of manifold constraints on adversarial attacks. First, we consider a case where the l_2-norm constraint is applied to the parameter shape. The results presented in Tab. 4 reveal that the ASR only reaches 76% on ShapeNet Part and 81% on ModelNet40, even after 100 iterations. This suggests the overly restrictive nature of the l_2-norm constraint. Additionally, we demonstrate the adversarial attack performance when only a single term of the proposed two manifold constraints, either distance or angle, is utilized. The results reveal that using either constraint can still achieve 100% ASR. However, there is a notable drop in imperceptibility performance, thereby confirming the essentiality of both distance and angle components in the manifold constraints.

Attack	Square Plate									Sphere								
	PointNet			DGCNN			PointConv			PointNet			DGCNN			PointConv		
	l_2	CD	HD	l_2	CD	HD	l_2	CD	HD	l_2	CD	HD	l_2	CD	HD	l_2	CD	HD
FGM	3.17	0.61	10.67	3.66	1.74	5.25	3.51	1.34	7.50	3.26	1.15	17.80	3.89	2.60	11.66	3.64	2.02	11.98
PGD	18.05	30.64	20.88	18.13	30.79	20.64	18.06	30.70	20.23	17.50	15.17	14.16	17.58	15.18	14.01	17.53	15.20	13.81
AdvPC	5.04	4.50	8.09	12.50	19.43	16.65	10.50	15.80	13.14	4.88	2.41	5.89	11.72	10.47	10.73	10.12	8.82	8.91
IFGM	1.57	0.53	4.89	1.91	0.86	4.78	1.59	0.66	3.49	1.28	0.35	2.90	1.82	0.64	2.67	1.51	0.50	1.98
C&W	2.91	1.91	4.13	8.45	12.59	7.60	6.06	7.69	4.85	2.81	1.15	2.96	7.96	7.00	5.03	5.89	4.90	3.44
GeoA3	3.39	2.29	4.18	9.12	13.98	8.18	7.19	8.24	7.13	3.13	2.32	3.98	8.43	7.23	6.25	6.19	5.34	4.76
SI-Adv	1.94	0.51	9.96	2.50	1.28	5.17	2.46	1.14	7.08	1.82	0.30	5.87	2.37	0.96	3.05	2.35	0.81	3.82
ITA	2.32	0.87	2.86	3.33	1.67	2.99	3.34	1.84	2.55	2.43	1.27	3.83	3.46	2.46	4.55	3.48	2.82	3.55
C&W-P/S	2.54	2.00	3.40	6.51	9.00	7.10	3.41	3.00	2.80	2.40	1.16	2.70	5.73	5.07	4.73	4.28	3.25	3.20
IFGM-P/S	**1.12**	**0.30**	**2.70**	**1.22**	**0.39**	**1.80**	**0.97**	**0.30**	**0.90**	**1.05**	**0.23**	**2.07**	**1.09**	**0.29**	**1.36**	**0.86**	**0.23**	**0.75**

Table 2: Comparison on the manifold-aware distortion required by various methods to reach their highest achievable ASR. The evaluation is conducted on ModelNet40, using square plate (-P) or sphere (-S) as the parameter shape.

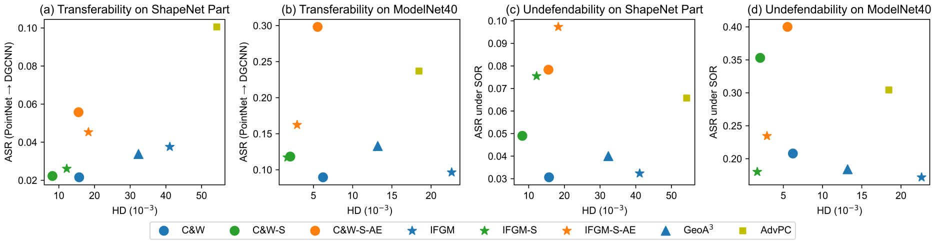

Figure 7: Visualization of (a-b) the transferability of different methods from PointNet to DGCNN on ShapeNet Part and ModelNet40, and (c-d) the undefendability of different methods under the statistical outlier removal (SOR) defense (Zhou et al. 2019) on ShapeNet Part and ModelNet40. The suffix "-AE" indicates the use of an auto-encoder (AE).

	ShapeNet Part			ModelNet40		
	ASR	CD	HD	ASR	CD	HD
IFGM-S (w/o Inv.)	0.82	2.67	34.46	0.87	0.78	3.46
IFGM-S (w/ Inv.)	1.00	**0.95**	**12.21**	1.00	**0.37**	**1.68**
C&W-S (w/o Inv.)	0.99	6.15	16.98	1.00	2.97	4.11
C&W-S (w/ Inv.)	1.00	**2.97**	**8.21**	1.00	**1.33**	**2.05**

Table 3: Comparison on the ASR and perturbation size brought by different methods under manifold constraints w/ and w/o using invertibility (Inv.) when attacking PointNet.

	ShapeNet Part			ModelNet40		
	ASR	CD	HD	ASR	CD	HD
l_2	0.76	0.67	23.87	0.81	0.32	5.79
EMD	1.00	1.45	19.38	1.00	0.49	3.78
angle	1.00	1.54	23.56	1.00	0.77	4.47
distance	1.00	1.59	22.02	1.00	0.73	4.51
angle & distance	1.00	**0.95**	**12.21**	1.00	**0.37**	**1.68**

Table 4: Comparison on the ASR and perturbation size brought by IFGM-S for attacking PointNet under different items of manifold constraints.

We also explore using earth mover's distance (EMD) to constrain the parameter shape. The results in Tab. 4 show the CD and HD values under EMD are lower than under any single component of our manifold constraints. However, the performance is worse than that achieved with our complete constraints, considering both distance and angle, highlighting the importance of this combined approach.

Analysis on Transferability and Undefendability. In an effort to explore further improvements in transferability and undefendability for our method, we incorporate an auto-encoder (AE) as a filter, a strategy inspired by AdvPC (Hamdi et al. 2020). This approach helps the point cloud to retain its adversarial characteristics after filtering, and lessens the dependence on specific classification models. As illustrated in Fig. 7, the application of AE in conjunction with our IFGM-S and C&W-S methods notably enhances transferability and undefendability. Although a slight increase in perturbation size is observed, as measured by HD, it still falls below that of the original methods that do not utilize manifold constraints. Interestingly, our AE-enhanced solutions, namely IFGM-S-AE and C&W-S-AE, provide improved performance in transferability and undefendability over AdvPC (Hamdi et al. 2020), but with significantly reduced distortion costs.

Conclusion

This paper has proposed novel manifold constraints to enhance the imperceptibility of adversarial attacks on 3D point clouds. The rationale involves capturing and constraining the manifold-aware distortion, by transforming it to the parameter space. Extensive experiments validate the severity of manifold-aware distortion in adversarial attacks and the efficacy of constraining it in enhancing imperceptibility.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (62102105, U20B2046), the Guangdong Basic and Applied Basic Research Foundation (2022A1515011501, 2022A1515010138), the Guangzhou Science and Technology Program (202201020229), and the Grant of Guangzhou University (YJ2023048, GD22TWCXGC15).

References

Behrmann, J.; Grathwohl, W.; Chen, R. T.; Duvenaud, D.; and Jacobsen, J.-H. 2019. Invertible residual networks. In *ICML*, 573–582.

Bronstein, M. M.; Bruna, J.; LeCun, Y.; Szlam, A.; and Vandergheynst, P. 2017. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4): 18–42.

Carlini, N.; and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, 39–57.

Chang, A. X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.

Chen, Y.; Peng, W.; Tang, K.; Khan, A.; Wei, G.; Fang, M.; et al. 2022. Pyrapvconv: efficient 3d point cloud perception with pyramid voxel convolution and sharable attention. *Computational Intelligence and Neuroscience*, 2022.

Dong, X.; Chen, D.; Zhou, H.; Hua, G.; Zhang, W.; and Yu, N. 2020. Self-Robust 3D Point Recognition via Gather-Vector Guidance. In *CVPR*, 11513–11521.

Gomez, A. N.; Ren, M.; Urtasun, R.; and Grosse, R. B. 2017. The reversible residual network: Backpropagation without storing activations. In *NeurIPS*, 2215–2225.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and harnessing adversarial examples. In *ICLR*.

Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; and Bennamoun, M. 2020. Deep learning for 3d point clouds: A survey. *IEEE TPAMI*, 43(12): 4338–4364.

Hamdi, A.; Rojas, S.; Thabet, A.; and Ghanem, B. 2020. AdvPC: Transferable adversarial perturbations on 3d point clouds. In *ECCV*, 241–257.

Hormann, K.; Polthier, K.; and Sheffer, A. 2008. Mesh Parameterization: Theory and Practice. In *ACM SIGGRAPH ASIA 2008 Courses*, SIGGRAPH Asia. New York, NY, USA. ISBN 9781450379243.

Huang, Q.; Dong, X.; Chen, D.; Zhou, H.; Zhang, W.; and Yu, N. 2022. Shape-invariant 3D Adversarial Point Clouds. In *CVPR*, 15335–15344.

Kim, J.; Hua, B.-S.; Nguyen, T.; and Yeung, S.-K. 2021. Minimal adversarial examples for deep learning on 3d point clouds. In *ICCV*, 7797–7806.

Lee, J. M.; and Lee, J. M. 2012. *Smooth manifolds*. Springer.

Lee, K.; Chen, Z.; Yan, X.; Urtasun, R.; and Yumer, E. 2020. ShapeAdv: Generating Shape-Aware Adversarial 3D Point Clouds. *arXiv preprint arXiv:2005.11626*.

Li, Q.; Li, X.; Cui, X.; Tang, K.; and Zhu, P. 2023. HEPT Attack: Heuristic Perpendicular Trial for Hard-label Attacks under Limited Query Budgets. In *CIKM*, 4064–4068.

Liu, D.; and Hu, W. 2023. Imperceptible Transfer Attack and Defense on 3D Point Cloud Classification. *IEEE TPAMI*, 45(4): 4727–4746.

Liu, D.; Yu, R.; and Su, H. 2019. Extending adversarial attacks and defenses to deep 3d point cloud classifiers. In *ICIP*, 2279–2283.

Maturana, D.; and Scherer, S. 2015. Voxnet: A 3d convolutional neural network for real-time object recognition. In *IROS*, 922–928.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS*, 8026–8037.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 652–660.

Spanier, E. H. 1989. *Algebraic topology*. Springer Science & Business Media.

Tang, K.; Chen, Y.; Peng, W.; Zhang, Y.; Fang, M.; Wang, Z.; and Song, P. 2022a. RepPVConv: attentively fusing reparameterized voxel features for efficient 3D point cloud perception. *The Visual Computer*, 1–12.

Tang, K.; Lou, T.; He, X.; Shi, Y.; Zhu, P.; and Gu, Z. 2023a. Enhancing Adversarial Robustness via Anomaly-aware Adversarial Training. In *International Conference on Knowledge Science, Engineering and Management*, 328–342.

Tang, K.; Ma, Y.; Miao, D.; Song, P.; Gu, Z.; Tian, Z.; and Wang, W. 2022b. Decision Fusion Networks for Image Classification. *IEEE TNNLS*, 1–14.

Tang, K.; Shi, Y.; Lou, T.; Peng, W.; He, X.; Zhu, P.; Gu, Z.; and Tian, Z. 2023b. Rethinking Perturbation Directions for Imperceptible Adversarial Attacks on Point Clouds. *IEEE Internet of Things Journal*, 10(6): 5158–5169.

Tang, K.; Shi, Y.; Wu, J.; Peng, W.; Khan, A.; Zhu, P.; and Gu, Z. 2022c. NormalAttack: Curvature-aware shape deformation along normals for imperceptible point cloud attack. *Security and Communication Networks*, 2022.

Tang, K.; Song, P.; and Chen, X. 2016. Signature of geometric centroids for 3d local shape description and partial shape matching. In *ACCV*, 311–326.

Tang, K.; Wu, J.; Peng, W.; Shi, Y.; Song, P.; Gu, Z.; Tian, Z.; and Wang, W. 2023c. Deep Manifold Attack on Point Clouds via Parameter Plane Stretching. In *AAAI*, volume 37, 2420–2428.

Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019. Dynamic graph cnn for learning on point clouds. *ACM TOG (SIGGRAPH)*, 38(5): 1–12.

Wen, Y.; Lin, J.; Chen, K.; Chen, C. P.; and Jia, K. 2022. Geometry-Aware Generation of Adversarial Point Clouds. *IEEE TPAMI*, 44(6): 2984–2999.

Wicker, M.; and Kwiatkowska, M. 2019. Robustness of 3d deep learning in an adversarial setting. In *CVPR*, 11767–11775.

Wu, W.; Qi, Z.; and Fuxin, L. 2019. Pointconv: Deep convolutional networks on 3d point clouds. In *CVPR*, 9621–9630.

Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, 1912–1920.

Xiang, C.; Qi, C. R.; and Li, B. 2019. Generating 3D Adversarial Point Clouds. In *CVPR*, 9136–9144.

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

Yang, Y.; Feng, C.; Shen, Y.; and Tian, D. 2018. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *CVPR*, 206–215.

Zhang, J.; Jiang, C.; Wang, X.; and Cai, M. 2021. Td-Net: Topology Destruction Network For Generating Adversarial Point Cloud. In *ICIP*, 3098–3102.

Zhao, Y.; Wu, Y.; Chen, C.; and Lim, A. 2020. On isometry robustness of deep 3d point cloud models under adversarial attacks. In *CVPR*, 1201–1210.

Zheng, T.; Chen, C.; Yuan, J.; Li, B.; and Ren, K. 2019. Pointcloud saliency maps. In *ICCV*, 1598–1606.

Zhou, H.; Chen, D.; Liao, J.; Chen, K.; Dong, X.; Liu, K.; Zhang, W.; Hua, G.; and Yu, N. 2020. Lg-gan: Label guided adversarial network for flexible targeted attack of point cloud based deep networks. In *CVPR*, 10356–10365.

Zhou, H.; Chen, K.; Zhang, W.; Fang, H.; Zhou, W.; and Yu, N. 2019. Dup-net: Denoiser and upsampler network for 3d adversarial point clouds defense. In *ICCV*, 1961–1970.

Zhu, P.; Fan, Z.; Guo, S.; Tang, K.; and Li, X. 2023a. Improving adversarial transferability through hybrid augmentation. *Computers & Security*, 103674.

Zhu, P.; Hong, J.; Li, X.; Tang, K.; and Wang, Z. 2023b. SGMA: a novel adversarial attack approach with improved transferability. *Complex & Intelligent Systems*, 1–13.