Deep Manifold Attack on Point Clouds via Parameter Plane Stretching

Keke Tang[1*], Jianpeng Wu[1*], Weilong Peng[1†], Yawen Shi[1], Peng Song[2], Zhaoquan Gu[3,4], Zhihong Tian[1], Wenping Wang[5]

[1] Guangzhou University
[2] Singapore University of Technology and Design
[3] Harbin Institute of Technology (Shenzhen)
[4] Peng Cheng Laboratory
[5] Texas A&M University
tangbohutbh@gmail.com, lesswu666@gmail.com, wlpeng@gzhu.edu.cn, shiyawen666@gmail.com, peng_song@sutd.edu.sg, guzhaoquan@hit.edu.cn, tianzhihong@gzhu.edu.cn, wenping@cs.hku.hk

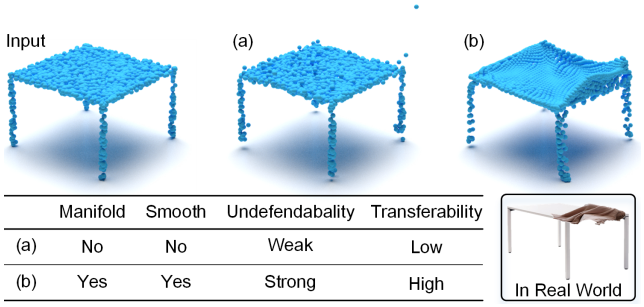
Abstract

Adversarial attack on point clouds plays a vital role in evaluating and improving the adversarial robustness of 3D deep learning models. Existing attack methods are mainly applied by point perturbation in a non-manifold manner. In this paper, we formulate a novel manifold attack, which deforms the underlying 2-manifold surfaces via parameter plane stretching to generate adversarial point clouds. First, we represent the mapping between the parameter plane and underlying surface using generative-based networks. Second, the stretching is learned in the 2D parameter domain such that the generated 3D point cloud fools a pretrained classifier with minimal geometric distortion. Extensive experiments show that adversarial point clouds generated by manifold attack are smooth, undefendable and transferable, and outperform those samples generated by the state-of-the-art non-manifold ones.

Introduction

With the advance of depth sensing devices and 3D deep learning techniques, applications of 3D point cloud perception are now booming. However, 3D deep learning models for point clouds are vulnerable to adversarial attacks as reported in (Xiang, Qi, and Li 2019), i.e., imperceptible modifications on input point clouds can lead to erroneous predictions of victim models, hindering their deployment in the real world, especially for safety-critical scenarios. Therefore, investigating adversarial attack on point clouds is critical for evaluating and improving the adversarial robustness of 3D deep learning models.

Due to the unstructured nature of point clouds, adversarial attack on them can be uniquely applied by dropping salient points (Zheng et al. 2019; Wicker and Kwiatkowska 2019) or adding adversarial points, clusters and objects (Xiang, Qi, and Li 2019). Meanwhile, mainstream adversarial attack approaches apply perturbation to point coordinates, e.g., via extending from image-based attacks (Xiang, Qi, and Li 2019; Liu, Yu, and Su 2019). Despite their success in attack, the generated adversarial point clouds tend to contain clearly visible outliers (Dong et al. 2020; Xiang, Qi, and Li 2019); see Fig. 1(a). Besides non-smoothness, most adversarial attacks on point clouds can hardly transfer to different classification models, and can be easily defended by adversarial defense techniques. Latest studies employ generative-based networks (Zhou et al. 2020), geometry-aware objectives (Wen et al. 2022), etc., to alleviate the above issues. However, very few attack methods can generate adversarial point clouds satisfying all three properties simultaneously, i.e., smoothness, undefendability, and transferability, since their perturbations are applied in non-manifold manners.



	Manifold	Smooth	Undefendability	Transferability
(a)	No	No	Weak	Low
(b)	Yes	Yes	Strong	High

Figure 1: Adversarial point clouds of category TABLE generated by non-manifold and manifold attack that mislead a victim model to recognize as CHAIR: (a) the adversarial point cloud generated by non-manifold attacks, e.g., I-FGM (Dong et al. 2020), contains clearly visible outliers, and thus is easily defended and hardly transferable to unseen models; (b) the adversarial point cloud generated by our manifold attack is more smooth, undefendable and transferable. Note that the adversarial point cloud in (b) can be captured if textile is put on the table in the real world, such that humans are not aware of the attack.

Generally, point clouds of 3D objects are assumed to be sampled along 2-manifold surfaces embedded in 3D Euclidean space (Gu, Gortler, and Hoppe 2002; Spanier 1989). If the perturbation is applied to a point cloud in a non-manifold manner without modifying the underlying 2-manifold surface, its adversarialness may only hold for at-

*These authors contributed equally.
†Corresponding author.

tacking a specified victim model but fail for other unseen ones. Besides, since the perturbation is applied out of the underlying surface, it will introduce noticeable outliers and thus can be defended by outlier removal techniques easily. These issues motivate us to design an attack methodology whose generated adversarial point clouds are smooth, transferable to unseen models and undefendable against adversarial defense, by intentionally applying perturbation to the underlying 2-manifold surfaces of point clouds; see Fig. 1(b).

In this paper, we formulate a novel manifold attack that perturbs point clouds by deforming their underlying surfaces. Specifically, we first utilize a manifold auto-encoder to learn the mapping between the 2D parameter plane and the underlying 2-manifold surface of point cloud embedded in 3D space, and then the deformation of surface can be realized by stretching the parameter plane instead, guided by a pretrained classifier to obtain adversarialness. In particular, we utilize thin plate spline (TPS) transformation (Bookstein 1989) to stretch the parameter plane, such that only small deformation is introduced for better smoothness. Besides, we observe that manifold attack can be *hidden in the human psyche*, since its perturbation could be associated with the disturbances in the real world to become justified, as in (Xiang, Qi, and Li 2019); see Fig. 1. We validate the effectiveness of our manifold attack framework by attacking multiple different deep classification models and adversarial defense methods. Extensive experimental results show that smooth adversarial point clouds generated by our manifold attack framework are undefendable against adversarial defenses and transferable to unseen classification models even under defense, which outperform those generated by the state-of-the-art methods in non-manifold manners.

Overall, our contribution is summarized as follows:

- We are the first to formulate manifold attack on 3D point clouds by perturbing the underlying surfaces explicitly.
- We devise a smooth perturbation mechanism that deforms the underlying 2-manifold surfaces of point clouds via parameter plane stretching.
- We show by experiments that manifold attack achieves superior performance to non-manifold ones in both undefendability and transferability.

Related Work

Adversarial Attacks on Point Clouds. Adversarial attack aims to generate an example that will lead the victim network to make a mistake, and has been successfully extended from 2D image classification (Szegedy et al. 2014; Goodfellow, Shlens, and Szegedy 2015; Moosavi-Dezfooli, Fawzi, and Frossard 2016; Carlini and Wagner 2017; Yuan et al. 2019) to that of 3D point clouds. According to the way to apply adversarial attacks, current approaches can be divided into three categories: addition-based, deletion-based and perturbation-based. Addition-based methods lead to the mistakes of models by adding independent points, clusters or objects (Xiang, Qi, and Li 2019). Deletion-based methods influence the classifier by dropping critical points (Zheng et al. 2019; Yang et al. 2019; Wicker and Kwiatkowska 2019; Zhang et al. 2021). Perturbation-based methods apply attacks by perturbing existing points (Xiang, Qi, and Li 2019; Liu, Yu, and Su 2019; Zhao et al. 2020; Kim et al. 2021; Wen et al. 2022; Huang et al. 2022; Liu and Hu 2022; Tang et al. 2022c,d). In this paper, we only consider perturbation-based methods.

Perturbation-based 3D Adversarial Attacks. Xiang, Qi, and Li (2019) and Liu, Yu, and Su (2019) pioneered the perturbation-based 3D adversarial attack by extending C&W attack (Carlini and Wagner 2017) and FGSM (Goodfellow, Shlens, and Szegedy 2015) under l_2-norm constraint. Instead of manipulating local points, Zhao et al. (2020) proposed an isometric transformation attack that can fool 3D deep learning models using simple rotations. To avoid visible outliers, Kim et al. (2021) proposed to perturb a minimal subset of points, while Wen et al. (2022) and Huang et al. (2022) applied geometry-aware objectives/constraints.

To exploit deep learning techniques for improving adversarial attack, generative-based attack methods perturb points in the latent space by using auto-encoder (AE) (Hinton and Salakhutdinov 2006), variational AE (Kingma and Welling 2013) or generative adversarial network (GAN) (Goodfellow et al. 2014). Hamdi et al. (2020) proposed an AE-based attack model called AdvPC, that optimizes the perturbation for fooling before passing through the AE to enforce less dependent on the victim network and generalize better to different networks. Zhou et al. (2020) employed GAN in generating adversarial point clouds with target labels, and thus are less perceptible. Lee et al. (2020) generated adversarial point clouds by directly adding perturbation noise into the latent space of AEs.

Our manifold attack is also generative-based. Different from the above approaches that add perturbations in the latent space of deep networks, we aim to stretch the underlying 2-manifold surfaces. Although their latent spaces are also considered to be manifold, they are characterized as distribution property w.r.t. a dataset, but not geometric property w.r.t. a surface. We notice that LG-GAN (Zhou et al. 2020) also mentioned their relationship with the manifold concept. Differently, they enforced the points to be adjoined to the manifold surface after applying perturbation. Namely, LG-GAN restricts the perturbation to the manifold, while our approach perturbs the manifold directly.

Deep Learning for Point Clouds and 2-manifold Representation. Since PointNet (Qi et al. 2017a) pioneered the processing of point clouds directly using deep learning techniques (Tang et al. 2022b), a large body of researches have been studied (Qi et al. 2017b; Wang et al. 2019; Li et al. 2018; Chen et al. 2022; Tang et al. 2022a). We aim to evaluate and improve their robustness to adversarial attacks.

Another related direction is 2-manifold representation learning of 3D surfaces (Gu, Gortler, and Hoppe 2002; Sander et al. 2003). Yang et al. (2018) proposed FoldingNet, which learns to deform a canonical 2D grid onto the underlying 3D object surface of a point cloud. To handle objects with larger genus numbers, they further devised an extension, i.e, TearingNet (Pang, Li, and Tian 2021), which can learn more topology-friendly representations. AtlasNet (Groueix et al. 2018) represents a 3D shape as a col-

lection of parametric surface elements, instead of one, and thus can handle more complex shapes. We also aim to represent point cloud shapes in 2-manifold, and we choose FoldingNet in our work for its simplicity. Differently, the 2-manifold representation is utilized for applying adversarial attack, which has not been investigated before.

Problem Formulation

Preliminary. Given a point cloud $P \in \mathbb{R}^{n\times 3}$ sampled from the object surface $\mathcal{S}$ and its label $y \in \mathbb{Z}$, perturbation-based adversarial attack aims to mislead a 3D deep classification model by feeding an adversarial point cloud P^{adv} instead of P via applying an intentionally designed perturbation, such that the model makes an error prediction.

Non-manifold Attack. A classic adversarial attack on point cloud, e.g., Xiang, Qi, and Li (2019), is formulated as:

$$P^{adv} = P + \sigma, \tag{1}$$

where $\sigma \in \mathbb{R}^{n\times 3}$ is the perturbation offset in the 3D Euclidean space. Since the perturbation is not applied on the underlying surface $\mathcal{S}$, the adversarial attack is in a **non-manifold manner**.

In fact, the permutation in non-manifold attack tends to generate out-of-surface points which can be removed by simple techniques of outlier removal (Zhou et al. 2019), and thus is *easy to defend*. Moreover, since the intrinsic property of the surface almost maintains, the success of adversarial attack is attributable to a certain blind spot of the specified victim deep model, instead of the ambiguity of the modified shape. Therefore, these adversarial point clouds are *hardly transferable* to other unseen models.

Manifold Attack. We aim to investigate generating adversarial point clouds in a **manifold manner**. Specifically, we impose a "virtual force" to slightly deform the underlying surface $\mathcal{S}$ of P for fooling the victim model, and thus obtain the adversarial surface:

$$\mathcal{S}^{adv} = Def(\mathcal{S}), \tag{2}$$

where $Def(\cdot)$ is the perturbation function for deformation. Therefore, the corresponding adversarial point cloud of P, i.e., P^{adv}, is the new point set distributed along $\mathcal{S}^{adv}$.

Parameter Plane Stretching for Deformation in Manifold Attack. The deformation for manifold attack in Eqn. 2 can be implemented in many different ways. In this paper, we only consider deforming the surface $\mathcal{S}$ in the tangent space.

Suppose there exists a mapping function $\mathcal{M}$ from the parameter plane u to 2-manifold surface $\mathcal{S}$ (Hormann, Polthier, and Sheffer 2008):

$$\mathcal{M} : u \longrightarrow \mathcal{S}. \tag{3}$$

We can further simplify the problem of deforming in the tangent space to stretching in the parameter plane domain. Specifically, we first apply a 2D transformation τ to the parameter plane u,

$$u^{adv} = \tau(u), \tag{4}$$

and then feed the stretched u_{adv} into $\mathcal{M}$, obtaining the deformed surface of $\mathcal{S}$,

$$\mathcal{S}^{adv} = \mathcal{M}(u^{adv}). \tag{5}$$

Discussion. In contrast to non-manifold attack, manifold attack changes the underlying surfaces of point clouds, and thus brings "inherent" adversarialness. Therefore, adversarial point clouds generated by manifold attack are more undefendable and transferable. Besides, by utilizing the parameter plane as a bridge, manifold attack via complex 3D deformation of surface is finally reduced to via simple 2D stretch of parameter plane, facilitating finding feasible solutions of adversarial perturbation.

Method

In this section, we will first describe how to represent the manifold attack on surfaces using neural networks, and then introduce our framework for attacking point clouds and its training scheme.

Deep Representation of Manifold Attack

Deep Representation of Mapping. We utilize neural networks to represent the mapping $\mathcal{M}$ between parameter plane and 2-manifold surface in Eqn. 3. In specific, one network $E_{\mathcal{M}}$ is utilized to learn the representation of mapping $\mathcal{M}$, i.e., $\theta_{\mathcal{M}}$, from $\mathcal{S}$ and its parameter plane $u_{\mathcal{S}}$:

$$\theta_{\mathcal{M}} = E_{\mathcal{M}}(\mathcal{S}, u_{\mathcal{S}}), \tag{6}$$

and another network $D_{\mathcal{M}}$ to learn the generating of surface $\hat{\mathcal{S}}$ from $u_{\mathcal{S}}$ and $\theta_{\mathcal{M}}$:

$$\hat{\mathcal{S}} = D_{\mathcal{M}}(\theta_{\mathcal{M}}, u_{\mathcal{S}}). \tag{7}$$

By enforcing $\hat{\mathcal{S}}$ to be $\mathcal{S}$, the mapping denoted in Eqn. 3 is represented with neural networks.

Attack under Mapping Representation. With the deep representation of mapping $\theta_{\mathcal{M}}$ fixed, adversarial attack can be realized by parameter plane stretching following Eqn. 4:

$$\mathcal{S}^{adv} = D_{\mathcal{M}}(\theta_{\mathcal{M}}, u_{\mathcal{S}}^{adv}) = D_{\mathcal{M}}\left(\theta_{\mathcal{M}}, \tau(u_{\mathcal{S}})\right). \tag{8}$$

Manifold Attack Framework for Point Clouds

Since point cloud P derives from its inherent 2-manifold surface $\mathcal{S}$, we devise a novel manifold attack framework for point clouds by extending the deep representation of manifold attack on surfaces. It consists of three key components: manifold auto-encoder, TPS-based parameter plane stretching, and adversarial point cloud generation. Please refer to Fig. 2 for demonstration.

Manifold Auto-encoder. We use an auto-encoder to implement the deep representation of $\mathcal{M}$. Given a point cloud P as input, the encoder $\mathcal{E}$ outputs the deep mapping representation θ_P, and then the decoder $\mathcal{D}$ generate $\bar{P}$ to reconstruct P, by folding the fixed parameter plane u denoted with $n\times n$ 2D point grid under the guidance of θ_P:

$$\bar{P} = \mathcal{D}(\theta_P, u), \text{ with } \theta_P = \mathcal{E}(P).$$

To prevent outlier points, we use Hausdorff distance to supervise the reconstruction:

$$L_{rec}(\bar{P}, P) = \max\left(D_H(\bar{P}, P), D_H(P, \bar{P})\right), \tag{9}$$

where $D_H(\bar{P}, P) = \max\limits_{i\in\{1,\dots,n\}} \min\limits_{j\in\{1,\dots,n\}} \left\|\bar{P}_i - P_j\right\|_2^2$. Note that, we omit the input u of $\mathcal{E}$, since it is fixed and thus does not affect learning.

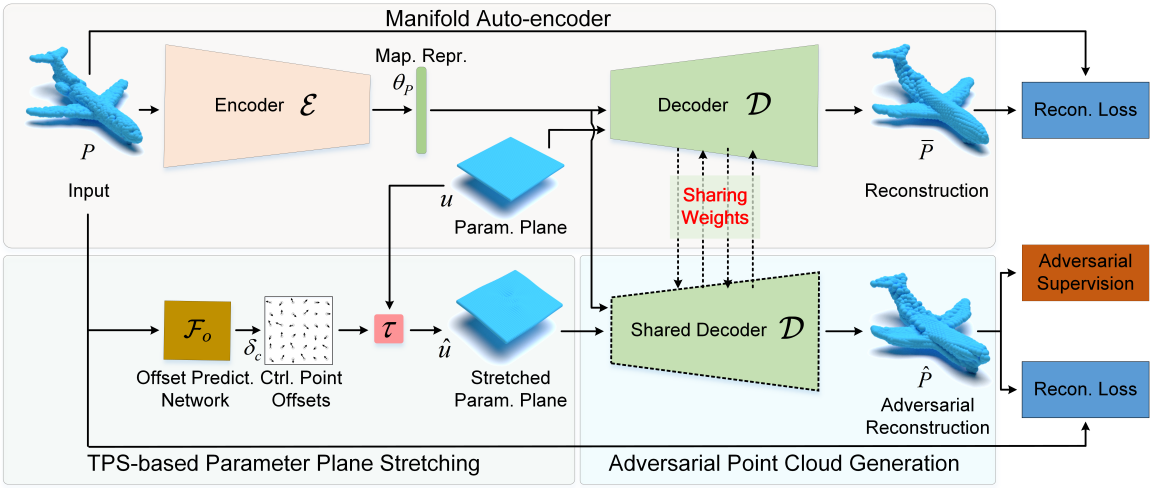


Figure 2: Demonstration of the manifold attack framework for point clouds.

TPS-based Parameter Plane Stretching. To facilitate small and smooth distortion, we utilize thin plate spline (TPS) transformation (Bookstein 1989) to stretch u. Specifically, we first sample $m \times m$ control points u_c evenly, and then adopt an offset prediction network $\mathcal{F}_o$ to learn the offset of u_c along the parameter plane:

$$\delta_c = \mathcal{F}_o(P),$$

and finally the offset will be propagated to all points in u by interpolation, forming the deformed point grid $\hat{u}$:

$$\hat{u} = \tau(u) = OP_{tps}(u, u_c, \delta_c).$$

By perturbing a small number of control points, TPS transformation enforces the stretching from u to $\hat{u}$ to be smooth.

Adversarial Point Cloud Generation. By feeding the stretched parameter plane $\hat{u}$ together with the deep mapping representation θ_P to the decoder $\mathcal{D}$, point cloud $\hat{P}$ can be reconstructed via:

$$\hat{P} = \mathcal{D}(\theta_P, \hat{u}),$$

under the supervision of reconstruction loss $L_{rec}(\hat{P}, P)$ as in Eqn. 9 to enforce the perturbation to be small.

To further make $\hat{P}$ to be adversarial, we enforce it to mislead a pretrained deep classifier $\mathcal{C}$, e.g., PointNet (Qi et al. 2017a), by applying the loss defined as follows:

$$L_{adv}(\hat{P}) = \max(\mathcal{C}(\hat{P})[y] - 1/k, 0), \qquad (10)$$

where y is the ground truth category of P, $\mathcal{C}(\hat{P})[y]$ is the output of classifier $\mathcal{C}$ on $\hat{P}$ of category y, and k is the total number of categories. Since it is impossible that prediction confidences on all categories are lower than $1/k$, there must be an incorrect category predicted with a higher confidence.

Training Scheme

We first pretrain the manifold auto-encoder $\{\mathcal{E},\mathcal{D}\}$ and the classifier $\mathcal{C}$, and then train the manifold attack framework to learn TPS transformation to generate adversarial point clouds, with the parameters of $\mathcal{E}$ and $\mathcal{C}$ fixed. For the training, we apply the loss function with adversarial supervision of $\hat{P}$, and reconstruction requirement for both $\bar{P}$ and $\hat{P}$:

$$L = \alpha L_{adv}(\hat{P}) + L_{rec}(\hat{P}, P) + L_{rec}(\bar{P}, P), \qquad (11)$$

where α is a weighting parameter, setting as 0.2 in our paper.

Experiments

Experimental Setup

Implementation. We implement the manifold auto-encoder using FoldingNet (Yang et al. 2018) in PyTorch. Specifically, the mapping representation θ_P is denoted with the codeword in size of 1×1024, and the parameter plane is denoted with a 45×45 point grid in the range of $[-0.3, 0.3]$. For TPS transformation, we use 4×4 control points. The offset prediction network $\mathcal{F}_o$ is implemented with MLP (3→64→128→1024)-MaxPool-FC (1024→512→256→32)-Tanh to predict the offsets of control points along two axes in the range of $[-1.0, 1.0]$. Both the pretrain of manifold auto-encoder and the training of manifold attack framework are performed on a workstation with one NVIDIA RTX 2080Ti GPU for 1000 epochs.

Datasets. We adopt two public datasets for evaluation: ShapeNet Part (Chang et al. 2015) and ModelNet40 (Wu et al. 2015). We select 14007 point clouds for training and 2874 for testing on ShapeNet Part, while 9843 for training and 2468 for testing on ModelNet40 following (Qi et al. 2017b). To fit our manifold attack framework, we randomly sample $45 \times 45 = 2025$ points from each point cloud.

Attack Methods. We compare our manifold attack framework (Ours) with ten baseline methods, including the deletion-based method, e.g., Drop-400 (Zheng et al. 2019) that drops the most critical 400 points, the perturbation-based methods using gradient, e.g., FGM, I-FGM and PGD (Dong et al. 2020), the perturbation-based methods using optimization, e.g., C&W under l_2-norm (l_2), Chamfer distance (CD), Hausdorff distance (HD) constraints (Xi-

Model	Data	Defense	Attack Method										
			Drop-400	FGM	I-FGM	PGD	CW (l_2)	CW (CD)	CW (HD)	GeoA^3	AdvPC	LG-GAN	Ours
PointNet	SN	-	44.61	51.61	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	98.20	96.45
		SRS	44.29	46.25	76.34	80.45	41.67	47.66	48.82	72.65	**99.60**	92.18	96.39
		SOR	43.84	10.37	4.53	3.20	3.74	3.52	3.52	12.11	48.05	28.72	**90.05**
		DUP-Net	42.03	8.07	3.30	2.75	3.74	3.71	3.51	8.20	29.49	24.13	**86.53**
		IF-Defense	30.34	8.18	5.18	6.09	6.25	4.88	5.47	9.76	17.38	26.25	**66.21**
	MN	-	59.64	85.26	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	99.22	93.80
		SRS	58.14	80.92	91.69	49.76	53.00	67.19	69.94	81.65	**98.87**	92.13	93.48
		SOR	56.28	33.23	21.20	26.78	13.82	15.63	15.80	42.79	46.19	67.25	**87.00**
		DUP-Net	55.39	27.19	16.29	24.63	12.44	13.09	12.89	7.08	30.31	64.04	**86.22**
		IF-Defense	37.84	21.43	13.80	19.94	13.49	14.84	13.70	6.04	16.77	55.97	**81.36**
DGCNN	SN	-	16.25	6.02	79.75	77.97	**100.00**	**100.00**	**100.00**	**100.00**	77.29	75.41	95.20
		SRS	12.39	3.97	44.99	53.06	42.14	12.09	12.08	86.46	48.33	70.29	**93.84**
		SOR	15.03	1.77	10.19	5.46	2.26	2.92	3.13	56.25	11.04	50.17	**93.11**
		DUP-Net	18.55	2.61	3.06	4.00	2.40	3.13	2.97	39.16	8.33	37.64	**93.84**
		IF-Defense	14.93	5.01	4.73	8.56	4.17	4.17	4.58	3.17	12.50	38.41	**89.17**
	MN	-	45.91	51.30	99.96	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	94.58	86.08	97.45
		SRS	35.05	40.64	55.71	74.68	31.09	37.11	32.29	77.71	70.63	80.60	**94.73**
		SOR	34.24	26.09	30.39	42.30	11.33	13.18	14.58	52.50	57.08	71.59	**95.10**
		DUP-Net	39.59	26.37	21.15	39.10	11.72	14.84	16.25	33.34	48.12	65.71	**94.65**
		IF-Defense	37.97	22.81	21.13	29.17	17.97	16.79	18.33	28.75	25.00	60.26	**90.03**
PointConv	SN	-	16.35	21.56	97.56	98.85	**100.00**	**100.00**	**100.00**	**100.00**	98.54	61.21	91.06
		SRS	15.00	12.94	54.39	86.22	26.67	27.29	26.95	33.84	**90.42**	55.75	90.29
		SOR	17.64	6.12	26.20	24.60	12.71	9.16	8.59	19.14	72.92	50.48	**88.83**
		DUP-Net	22.96	7.16	11.80	24.00	10.83	8.96	5.46	17.97	55.00	46.52	**93.35**
		IF-Defense	13.74	5.57	5.15	9.99	9.17	8.75	6.66	7.42	26.04	34.22	**78.25**
	MN	-	37.12	46.68	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	96.09	98.54	78.04	94.98
		SRS	35.09	40.19	95.06	**99.43**	37.29	28.95	27.77	21.48	93.54	71.88	94.06
		SOR	34.44	22.97	59.36	72.77	18.13	17.29	19.16	18.35	**91.25**	63.88	90.71
		DUP-Net	33.55	21.83	29.90	70.34	13.74	14.58	13.33	9.38	76.04	65.17	**87.32**
		IF-Defense	32.82	23.78	25.61	43.76	11.46	10.62	12.08	4.29	35.62	57.77	**83.51**

Table 1: ASR (%) of different attack methods with and without defense on ShapeNet Part (SN) and ModelNet40 (MN).

ang, Qi, and Li 2019), GeoA^3 (Wen et al. 2022) that applies geometric-aware constraints, AdvPC (Hamdi et al. 2020) whose focus is to achieve high transferability, and the generative-based LG-GAN (Zhou et al. 2020).

Victim Models. We choose three representative deep classification models for point clouds to attack, including the MLP-based PointNet (Qi et al. 2017a), the graph-based DGCNN (Wang et al. 2019) and the kernel-based PointConv (Wu, Qi, and Fuxin 2019). For the training of these models, we follow their original papers.

Defense Methods. We adopt four adversarial defense methods: simple random sampling (SRS), statistical outlier removal (SOR), denoiser and upsampler network (DUP-Net) (Zhou et al. 2019), and IF-Defense (Wu et al. 2020). SRS applies defense by randomly dropping 20% points from the input point clouds, and SOR trims the irregular points that violate the mean and standard deviation of the nearest neighbor distances. Based on SOR, DUP-Net further applies point cloud upsampling to remap off-the-manifold adversarial samples on to the natural manifold. IF-Defense predicts implicit functions that capture the clean shapes and then restore the adversarial point clouds.

Evaluation Setting and Metrics. We choose the optimal configurations of these adversarial attack methods to achieve the best attack success rates they could reach. Under this maximal adversarialness setting (Liu and Hu 2022), we evaluate the robustness against defense and the transferability for fair comparisons. For all performance metrics, we measure them using the *attack success rate (ASR)*, which is the percentage of generated adversarial point clouds that lead the victim model to make mistakes.

Performance Comparison and Analysis

Attack Performance. The results in Tab. 1 show that Drop-400 and FGM perform the worst while the others reveal their strong abilities in attacking models and achieve over 95% ASR in most cases. In particular, LG-GAN and Ours also attack these models successfully, but the ASRs are slightly lower than those of I-FGM, C&W, and GeoA^3, due to the dataset-oriented nature of generation models instead of the sample-oriented one. AdvPC has similarly lower ASR, due to the trade-off imposed by auto-encoder for transferability.

Data	Source	Target	Attack Method										
			Drop-400	FGM	I-FGM	PGD	CW (l_2)	CW (CD)	CW (HD)	GeoA^3	AdvPC	LG-GAN	Ours
ShapeNet Part	PointNet	DGCNN	14.13	4.46	4.14	4.66	3.75	3.52	3.32	8.20	9.17	**40.87**	23.94
		PointConv	19.17	4.53	4.07	28.57	4.17	4.30	4.59	13.28	26.76	**38.72**	23.00
	DGCNN	PointNet	18.72	8.39	37.30	45.38	18.96	17.50	16.45	19.59	53.33	**61.84**	34.86
		PointConv	15.83	11.38	20.04	45.13	15.90	7.50	8.33	16.87	69.17	51.92	**71.37**
	PointConv	PointNet	12.32	10.86	5.11	5.95	4.17	4.16	3.91	5.08	6.67	**52.25**	23.42
		DGCNN	6.27	4.91	2.82	3.28	2.92	2.13	2.73	4.30	3.54	39.21	**43.04**
ModelNet40	PointNet	DGCNN	25.24	38.13	23.87	37.20	18.31	18.17	19.08	13.33	29.78	**86.53**	61.84
		PointConv	25.69	29.01	16.90	73.50	15.03	15.43	14.95	14.42	17.26	60.91	**73.87**
	DGCNN	PointNet	25.49	30.27	17.91	26.63	12.89	13.67	15.42	12.92	18.95	**83.60**	67.18
		PointConv	27.92	34.04	18.44	77.19	9.77	8.59	13.54	13.96	29.79	79.42	**87.61**
	PointConv	PointNet	24.47	31.44	18.60	23.34	13.12	13.96	13.54	5.47	17.91	**60.60**	41.61
		DGCNN	25.73	35.41	31.08	34.08	19.58	19.17	18.96	13.28	29.16	65.54	**75.21**

Table 2: Transferability performance of different attack methods. The transferability is measured by ASR (%) on target models using adversarial examples that are generated for attacking source models.

Source	Target	PGD	AdvPC	LG-GAN	Ours
PointNet	DGCNN	-10.67	-13.09	-25.29	**+2.36**
	PointConv	-18.44	-7.09	-9.41	**-2.52**
DGCNN	PointNet	-6.65	-2.49	-22.80	**+2.17**
	PointConv	-20.47	-8.75	-12.14	**-2.34**
PointConv	PointNet	-3.49	-1.66	-9.22	**+8.06**
	DGCNN	-5.96	-3.33	+0.49	**+5.05**

Table 3: Changes on the transferability results of different attack methods on ModelNet40 after applying SOR defense.

Figure 3: Two real-world scenarios that "generate" manifold attack results.

Attack Performance under Defense. The results in Tab. 1 show that the simple SRS can already defend a part of attack methods, while SOR defends almost all strong attack methods except Ours, AdvPC and LG-GAN. By applying the powerful DUP-Net and IF-Defense, the ASRs of most attack methods decrease to below 10%. AdvPC, LG-GAN and our manifold attack are robust in almost all cases, and Ours performs the best. Although the ASR of our method decreases slightly after applying defense, we would like to remind that nearly 90% adversarial point clouds generated by our manifold attack can still mislead the victim models successfully. Therefore, we conclude that our manifold attack has strong undefendability.

Transferability. By comparing the results reported in Tab. 2 and Tab. 1, we can see that adversarial point clouds generated by most attack methods on source models can hardly transfer to attack unseen target models, except PGD, AdvPC, LG-GAN and Ours. In particular, LG-GAN has the strongest transferability, and Ours is the second best.

Transferability under Defense. In real-world scenarios, deployed models are generally invisible to attackers and also equipped with defense. Therefore, we further measure the amplitude changes of the four best methods on transferability after applying SOR defense and report the values in Tab. 3. Remarkably, the performance of PGD, AdvPC and LG-GAN drops a lot, e.g., decreases more than 10% when transferring from PointNet to DGCNN, indicating their transferability is fragile. Instead, the transferability of our manifold attack almost maintains, validating its superiority. Since LG-GAN is also generative-based and deformation-based, we infer that our strong transferability is brought by the unique smooth deformation mechanism, instead of simply by generation-based models or deformation.

Special Effect of Manifold Attack. We observe that the proposed manifold attack can be *hidden in the human psyche* by associating the disturbances in the real world to make the attacks to be justified similar as the adding of adversarial object, e.g., AIRPLANE, in (Xiang, Qi, and Li 2019). The CAP can be captured if some pressures, e.g., gravity, are applied to the top of it, and the SKATEBOARD can be captured if textile is put on it in the real world; see Fig. 3.

Indeed, 3D objects that appear in the real world will not be exactly the same as those in the model library. Besides, since noises are introduced by depth-sensing devices, their differences with the model library will be further enlarged. Both factors make the perturbation applied by our manifold attack justified. Therefore, humans may not notice our attacks.

Visualization. We visualize the point clouds generated by

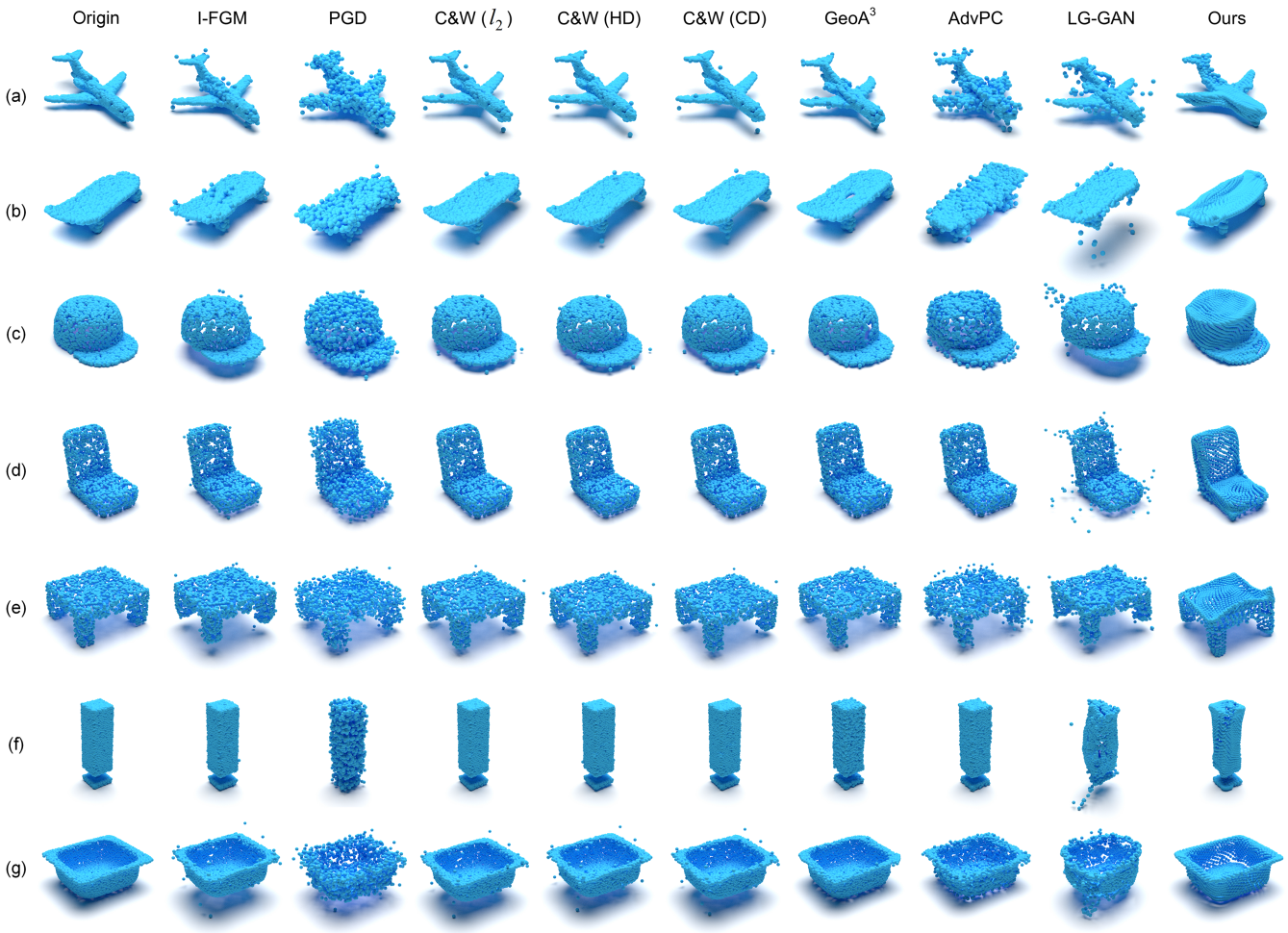


Figure 4: Visualization of original point clouds and the corresponding adversarial point clouds generated by different attack methods for attacking PointNet. The predicted categories before and after attack are: (a) AIRPLANE → SKATEBOARD, (b) SKATEBOARD → TABLE, (c) CAP → TABLE, (d) CHAIR → LAPTOP, (e) TABLE → CHAIR, (f) LAMP → DRESSER, (g) SINK → BATHTUB. The top five shapes are from ShapeNet Part, while the remaining two shapes are from ModelNet40.

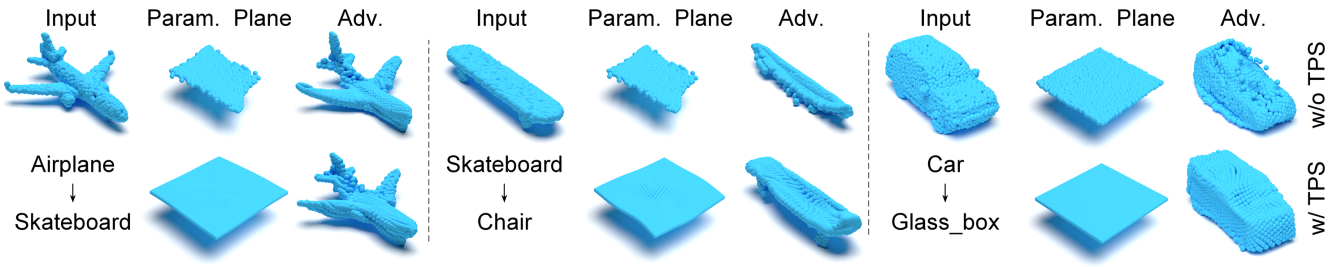


Figure 5: Visualization of parameter planes and the corresponding adversarial point clouds w/ and w/o using TPS.

both manifold and non-manifold adversarial attack methods for attacking PointNet on ShapeNet Part and ModelNet40 in Fig. 4. It could be seen that most adversarial point clouds generated by non-manifold attack methods have clearly visible outliers. Although GeoA^3 utilizes geometric properties, e.g, curvature, to restrict the perturbation, and LG-GAN enforces the points to be adjoined to the manifold shape by applying generative adversarial networks, outliers are still existed in their generated point clouds to obtain adversarialness. Differently, adversarial point clouds generated by our manifold attack are much more smooth, only with tiny shape deformation, validating our intuition.

CP	2×2	4×4	8×8	16×16
ASR	95.79	**96.45**	96.14	96.21
HD	0.039	**0.033**	0.038	0.038

Table 4: ASR (%) and HD vs. the number of control points (CP) in TPS for attacking PointNet on ShapeNet Part.

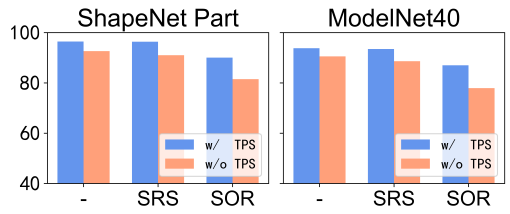


Figure 6: ASR (%) of our manifold attack on PointNet under different defense methods with and without TPS.

Ablation Studies and Analysis

Control Point Number. We evaluate four configurations of control points in TPS for attacking PointNet on the ShapeNet Part dataset. The results reported in Tab. 4 show that all four configurations can support manifold attack. In particular, 4×4 control numbers is the best for both attack performance and distortion.

TPS for Parameter Plane Stretching. To better demonstrate how TPS helps to control the stretching, we visualize the deformed parameter planes and the corresponding adversarial point clouds with and without applying TPS. The results in Fig. 5 show that the parameter planes will be unevenly stretched if without using TPS, and the resulting adversarial point clouds will deform dramatically, validating the usefulness of TPS.

To validate the usefulness of utilizing TPS transformation to control the stretching, we compare the results with the framework that predicts offsets of all the points on the parameter plane directly, without using TPS. The results in Fig. 6 show that the ASR of our framework will drop if without TPS in all cases, validating the effectiveness and necessity of utilizing TPS to control parameter plane stretching.

Conclusion

This paper has proposed a novel manifold attack which deforms the underlying 2-manifold surfaces of 3D point clouds. The key idea is to build the mapping between parameter plane and surface first, and then deform the surface by stretching parameter plane. Extensive experiments validate that adversarial point clouds generated by manifold attack are smooth, undefendable and transferable. We hope this work inspire more studies on manifold-aware deep learning models for point clouds.

Limitations and Future Work. Even though our manifold attack can simulate the disturbances in the real world, it is still challenging to deploy physical attack. In the future, we plan to explore simulation-to-reality transfer of digital adversarial point clouds into real-world objects.

Acknowledgements

We thank the reviewers for the valuable comments. This work was supported in part by the National Natural Science Foundation of China (62102105, U20B2046, and 61902082), Guangdong Basic and Applied Basic Research Foundation (2020A1515110997, 2022A1515011501, and 2022A1515010138), the Science and Technology Program of Guangzhou (202002030263, 202102010419 and 202201020229), the Open Project Program of the State Key Lab of CAD and CG (A2218), Zhejiang University, and the Major Key Project of PCL (No. PCL2022A03).

References

Bookstein, F. L. 1989. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE TPAMI*, 11(6): 567–585.

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.; and Fang, M. 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.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NeurIPS*, 2672–2680.

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

Groueix, T.; Fisher, M.; Kim, V. G.; Russell, B. C.; and Aubry, M. 2018. A papier-mâché approach to learning 3d surface generation. In *CVPR*, 216–224.

Gu, X.; Gortler, S. J.; and Hoppe, H. 2002. Geometry images. In *SIGGRAPH*, 355–361.

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

Hinton, G. E.; and Salakhutdinov, R. R. 2006. Reducing the dimensionality of data with neural networks. *Science*, 313(5786): 504–507.

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.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

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, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; and Chen, B. 2018. PointCNN: Convolution on χ-transformed points. In *NeurIPS*, 820–830.

Liu, D.; and Hu, W. 2022. Imperceptible Transfer Attack and Defense on 3D Point Cloud Classification. *IEEE TPAMI*, 1–18.

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

Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*, 2574–2582.

Pang, J.; Li, D.; and Tian, D. 2021. Tearingnet: Point cloud autoencoder to learn topology-friendly representations. In *CVPR*, 7453–7462.

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

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *NeurIPS*, 5099–5108.

Sander, P. V.; Wood, Z.; Gortler, S. J.; Snyder, J.; and Hoppe, H. 2003. Multi-Chart Geometry Images. In *SGP*, 146–155.

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

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing properties of neural networks. In *ICLR*.

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.; 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. 2022c. Rethinking Perturbation Directions for Imperceptible Adversarial Attacks on Point Clouds. *IEEE Internet of Things Journal*, 1–12.

Tang, K.; Shi, Y.; Wu, J.; Peng, W.; Khan, A.; Zhu, P.; and Gu, Z. 2022d. NormalAttack: Curvature-Aware Shape Deformation along Normals for Imperceptible Point Cloud Attack. *Security and Communication Networks*, 2022.

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.; Duan, Y.; Wang, H.; Fan, Q.; and Guibas, L. J. 2020. If-defense: 3d adversarial point cloud defense via implicit function based restoration. *arXiv preprint arXiv:2010.05272*.

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.

Yuan, X.; He, P.; Zhu, Q.; and Li, X. 2019. Adversarial examples: Attacks and defenses for deep learning. *IEEE TNNLS*, 30(9): 2805–2824.

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.