Shape-invariant 3D Adversarial Point Clouds Supplementary Materials

Qidong Huang[1] Xiaoyi Dong[1] Dongdong Chen[2] Hang Zhou[3]
Weiming Zhang[1,*] Nenghai Yu[1]
[1]University of Science and Technology of China [2]Microsoft Cloud AI [3]Simon Fraser University
{hqd0037@mail., dlight@mail., zhangwm@, ynh@}ustc.edu.cn
{cddlyf@, zhouhang2991@}gmail.com

A. The Primary Motivation

As we stated in the above manuscript, the objective of constructing the sensitivity map is to measure the variance of the recognition result when each point encounters the **imperceptible perturbation**. The reasons for why we use tangent plane gradients are:

First, the saliency map [19] is not effective for point perturbing attack we explored. It measures how the recognition result changes when we drop a point, which could be viewed as a *large* step perturb. While our perturbing attack relies on a proper measure of the change when we *slightly* perturbed the point. For example, on query-based black-box attack on PointNet++, saliency map obtains 92.6% ASR and 987.9 average queries(A.Q), while our method outperform it with 95.8% ASR and 417.0 A.Q under the same settings.

Second, from the perspective of invisibility, the typical gradient map does not have a perturb direction constrain, so the perturbed points are always out of the surface, which is easy to be detected or removed (Figure 1(b) of our manuscript). On the contrary, if adding perturbations along the tangent plane, it would be more invisible.

So we realize our tangent gradient map by projecting the gradient map to the tangent planes, which concerns both adversary effectiveness and invisibility.

B. Limitations and Social Impacts

One of the limitations of shape-invariant adversarial attack is *its unsatisfactory applicability on surfaces with large curvature.* On the one hand, in Eq.(1) of our main paper, the normal vector simulation about the investigated point with limited neighbor points might be not entirely accurate, especially when this point is located at the surface with large curvature (*e.g.*, the edges or corners of 3D shapes). On the other hand, since we design the perturbation along the tangent plane as the "explicit constrain" to maintain our claimed shape invariance, the perturbed point is still possible to escape from the surface if this position has very large curvature. This limitation partially breaks our shape-invariant assumption, leading to the imperfect visual quality. However, the solution to alleviate this problem is also obvious. We can adopt much smaller step size and more attack iterations (*i.e.*, more time budget) to realize attack.

The main social concerns about shape-invariant point-cloud attack might be that *it poses a threat to the security of autonomous driving.* The attacker can materialize the shape-invariant adversarial point clouds with 3D printing or combine the query-based attack with LiDAR spoofing attack [1, 11] to fool the point cloud recognition module of self-driving cars, leading to the traffic accidents. But the prerequisite of successfully realizing such attacks is that attacker needs to get the model output (*e.g.*, at least the predicted logits) to guide his/her attack loops. Therefore, unless pure black-box transfer-based attack, the self-driving can still defend itself by robustness enhancement and data leakage reduction based on technologies like strict encryption. From the positive perspective, as elaborated in our main paper (*i.e.*, Intro and Social Impact section), our method *provides a more effective evaluation method* for the adversarial robustness of point cloud recognition models, especially for black-box requirement *since the common protection agreement of trade secrets in current industry.* In other words, it facilitates the development of the researches on improving 3D recognition robustness.

C. Proof of Theorem 1

Proof. First, we need to clarify the translation relationship between the original coordinate system origin $\boldsymbol{O}$ and the new coordinate system origin $\boldsymbol{O}'$. As illustrated in Figure 3 in our main paper, since $\boldsymbol{O}'$ is the projection of $\boldsymbol{O}$ on the tangent plane Ω_i, so $\overrightarrow{\boldsymbol{O}'\boldsymbol{O}}$ is parallel to the normal vector $\boldsymbol{n}_i$. Thus the translation relationship can be calculated by

$$\overrightarrow{\boldsymbol{O}\boldsymbol{O}'} = k\boldsymbol{n}_i \Leftrightarrow \boldsymbol{O}(0,0,0), \boldsymbol{O}'(kn_{i1}, kn_{i2}, kn_{i3}), \quad (1)$$

*Corresponding author.

where $k = (\boldsymbol{p}_i \cdot \boldsymbol{n}_i)$ is the module length of vector $\boldsymbol{OO'}$.

Then, with the coordinate of $\boldsymbol{O'}$, we can easily obtain the vector $\overrightarrow{\boldsymbol{BA}}$ and $\overrightarrow{\boldsymbol{CO'}}$ that define the directions of the transformed axis x' and y' respectively. Based on the position representations in Eq.(3) of our main paper, we have

$$\overrightarrow{\boldsymbol{BA}} = (\frac{k}{n_{i1}}, -\frac{k}{n_{i2}}, 0), \tag{2}$$

$$\overrightarrow{\boldsymbol{CO'}} = (kn_{i1}, kn_{i2}, kn_{i3} - \frac{k}{n_{i3}}). \tag{3}$$

Therefore, if we define the standard orthogonal coordinate bases of the original coordinate system $\boldsymbol{O} - xyz$ as $\overrightarrow{\boldsymbol{x}}$, $\overrightarrow{\boldsymbol{y}}$ and $\overrightarrow{\boldsymbol{z}}$, the standard orthogonal coordinate bases $\overrightarrow{\boldsymbol{x}}'$, $\overrightarrow{\boldsymbol{y}}'$ and $\overrightarrow{\boldsymbol{z}}'$ of the new coordinate system $\boldsymbol{O'} - x'y'z'$ can be formulated as

$$\overrightarrow{\boldsymbol{x}}' = \frac{\overrightarrow{\boldsymbol{BA}}}{|\overrightarrow{\boldsymbol{BA}}|}, \quad \overrightarrow{\boldsymbol{y}}' = \frac{\overrightarrow{\boldsymbol{CO'}}}{|\overrightarrow{\boldsymbol{CO'}}|}, \quad \overrightarrow{\boldsymbol{z}}' = \frac{\overrightarrow{\boldsymbol{O'O}}}{|\overrightarrow{\boldsymbol{O'O}}|}, \tag{4}$$

$$\overrightarrow{\boldsymbol{x}}' = \Big(\frac{n_{i2}}{\sqrt{1-n_{i3}^2}}\Big)\overrightarrow{\boldsymbol{x}} + \Big(-\frac{n_{i1}}{\sqrt{1-n_{i3}^2}}\Big)\overrightarrow{\boldsymbol{y}} + 0\overrightarrow{\boldsymbol{z}}, \tag{5}$$

$$\overrightarrow{\boldsymbol{y}}' = \Big(\frac{n_{i1}n_{i3}}{\sqrt{1-n_{i3}^2}}\Big)\overrightarrow{\boldsymbol{x}} + \Big(\frac{n_{i2}n_{i3}}{\sqrt{1-n_{i3}^2}}\Big)\overrightarrow{\boldsymbol{y}} \tag{6}$$

$$+ \Big(-\sqrt{1-n_{i3}^2}\Big)\overrightarrow{\boldsymbol{z}}, \tag{7}$$

$$\overrightarrow{\boldsymbol{z}}' = n_{i1}\overrightarrow{\boldsymbol{x}} + n_{i2}\overrightarrow{\boldsymbol{y}} + n_{i3}\overrightarrow{\boldsymbol{z}}. \tag{8}$$

The above relationships between the bases of old/new orthogonal coordinate system reveal the rotation transformation f_{ir} of axes. Hence we can reorganize the above equations to get the translation transformation matrix, *i.e.*,

$$\boldsymbol{R}_i = \begin{pmatrix} \frac{n_{i2}}{\sqrt{1-n_{i3}^2}} & \frac{-n_{i1}}{\sqrt{1-n_{i3}^2}} & 0 \\ \frac{n_{i1}n_{i3}}{\sqrt{1-n_{i3}^2}} & \frac{n_{i2}n_{i3}}{\sqrt{1-n_{i3}^2}} & -\sqrt{1-n_{i3}^2} \\ n_{i1} & n_{i2} & n_{i3} \end{pmatrix}, \tag{9}$$

which denotes the rotation transformation from $\boldsymbol{O} - xyz$ to $\boldsymbol{O'} - x'y'z'$. Note that the denominator $\sqrt{1-n_{i3}^2}$ is equal to 0 when $n_{i3} = 1$. Thus we further consider the limit for this boundary case when $n_{i1} = n_{i2} = 0$ and $n_{i3} = 1$, *i.e.*,

$$\lim_{|n_{i3}|\to 1} \boldsymbol{R}_i = \begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ n_{i3}/\sqrt{2} & n_{i3}/\sqrt{2} & 0 \\ 0 & 0 & n_{i3} \end{pmatrix}. \tag{10}$$

Likewise, the translation transformation matrix can be obtained from the origin shift $\overrightarrow{\boldsymbol{OO'}}$ as

$$\boldsymbol{T}_i = (\boldsymbol{p}_i \cdot \boldsymbol{n}_i)\boldsymbol{n}_i, \tag{11}$$

which denotes the translation transformation from $\boldsymbol{O} - xyz$ to $\boldsymbol{O'} - x'y'z'$. To get the transformed coordinate of $\boldsymbol{p}_i$ (*i.e.*, its coordinate in the new coordinate system $\boldsymbol{O'} - x'y'z'$), we first shift it according to $\overrightarrow{\boldsymbol{OO'}}$ and then rotate it, which can be formulated as

$$\boldsymbol{p}_i' = \boldsymbol{R}_i(\boldsymbol{p}_i + \boldsymbol{T}_i). \tag{12}$$

By contrast, we can first apply rotation and then apply translation to realize the reverse coordinate transformation by

$$\boldsymbol{p}_i = \boldsymbol{R}_i^\top \boldsymbol{p}_i' - \boldsymbol{T}_i. \tag{13}$$

When we combine the above two equations, we can get

$$\boldsymbol{p}_i' = \boldsymbol{R}_i(\boldsymbol{p}_i + \boldsymbol{T}_i) = \boldsymbol{R}_i(\boldsymbol{R}_i^\top \boldsymbol{p}_i') = (\boldsymbol{R}_i\boldsymbol{R}_i^\top)\boldsymbol{p}_i', \tag{14}$$

where we can get $\boldsymbol{R}_i\boldsymbol{R}_i^\top = \boldsymbol{I}$. In this way, the l_2-norm of the rotation matrix $\boldsymbol{R}_i^\top$ can be determined by the maximal eigenvalue denoted by σ_{max}, *i.e.*,

$$\|\boldsymbol{R}_i^\top\|_2^2 = \sigma_{max}(\boldsymbol{R}_i\boldsymbol{R}_i^\top) = 1. \tag{15}$$

Thus the proof of Theorem 1 is completed. □

D. More Visualization Results

We provide more visual results for the proposed point-cloud sensitivity maps in Figure 1.

E. More Experimental Results

(1). White-box Performance on ShapeNetPart

Except for ModelNet40 [13], we also compare the proposed shape-invaraint white-box attack with baselines on ShapeNetPart [18]. We train three popular point cloud models (*i.e.*, PointNet [2], PointNet++ [10] and CurveNet [15]) on ShapeNetPart for 150 epochs. Initially, all of their clean recognition accuracy is nearly 99%. Similarly, we adopt the same attack settings with the settings clarified in our main paper (Sec 4.3). As the results listed in Table 1, our method can still achieve the low Chamfer distance [4] while maintaining over 90% attack success rate (ASR). As an gradient-based iterative attack method, it is hard-won for our method to achieve high ASR (effectiveness), low geometry distances (invisibility) and low A.T (efficiency) at the same time.

(2). Black-box Performance with PointNet as $\mathcal{H}_w$

Since the results reported in the main paper are obtained by utilizing DGCNN [12] as the surrogate model, the readers may be just wondering that *what if using the weaker model (*e.g.*, PointNet [2]) as the surrogate model?* As showcased in Table 2, even when we take PointNet as the weak surrogate model to implement our query-based attack, the attack performance just has few degradation. Specifically, when attacking on three of the most advanced point cloud recognition models including SimpleView [5], PAConv [16] and CurveNet [15], our performance on query cost and visual quality are still much better than SimBA [7] and SimBA+ [17] though few ASR is sacrificed.

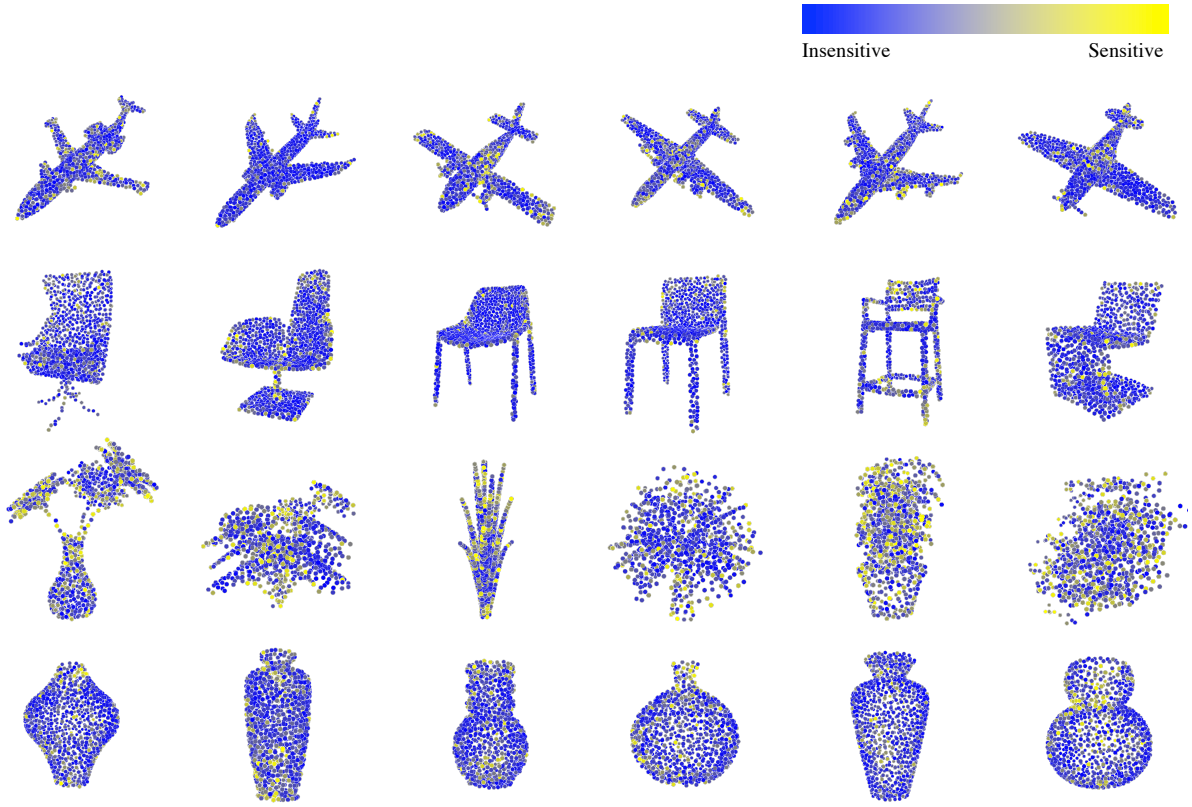

Figure 1. Visualization results of the proposed point-cloud sensitivity maps obtained on CurveNet.

Attack	Defense	PointNet [2]				DGCNN [12]				CurveNet [15]			
		ASR↑ (%)	CD↓ (10^{-4})	HD↓ (10^{-2})	A.T↓ (s)	ASR↑ (%)	CD↓ (10^{-4})	HD↓ (10^{-2})	A.T↓ (s)	ASR↑ (%)	CD↓ (10^{-4})	HD↓ (10^{-2})	A.T↓ (s)
I-FGM [6]	-	99.9	6.68	3.88	**1.05**	97.8	24.46	3.60	2.12	98.2	16.20	3.77	11.68
MI-FGM [3]	-	97.5	21.54	4.88	1.08	60.2	114.09	5.51	**2.11**	81.1	80.92	5.42	**11.56**
PGD [9]	-	99.9	6.63	3.88	1.06	97.8	24.26	3.61	3.16	98.2	16.33	3.77	11.58
3d-Adv [14]	-	**100.0**	7.30	3.75	5.09	**100.0**	15.43	5.13	19.10	**100.0**	15.47	3.59	121.94
AdvPC [8]	-	83.6	10.98	4.43	2.73	95.4	13.42	3.24	7.83	68.6	16.17	3.74	56.64
Ours	-	95.2	**3.59**	**3.46**	1.26	93.5	**9.08**	**2.95**	3.69	90.9	**8.29**	**3.58**	21.41

Table 1. Quantitative comparison on ShapeNetPart between our white-box shape-invariant attack and different white-box attacks, on attack success rate (ASR), Chamfer distance (CD), Hausdorff distance (HD) and average time budget for each adversarial point cloud generation (A.T), where Chamfer distance is multiplied by 10^4 and Hausdorff distance is multiplied by 10^2 for better comparison.

Attack	SimpleView [5]					PAConv [16]					CurveNet [15]				
	ASR↑ (%)	A.Q↓ (times)	CD↓ (10^{-4})	HD↓ (10^{-2})	A.T↓ (s)	ASR↑ (%)	A.Q↓ (times)	CD↓ (10^{-4})	HD↓ (10^{-2})	A.T↓ (s)	ASR↑ (%)	A.Q↓ (times)	CD↓ (10^{-4})	HD↓ (10^{-2})	A.T↓ (s)
SimBA [7]	100.0	119.5	8.65	5.35	0.61	100.0	73.2	4.51	4.93	0.36	100.0	114.5	4.67	4.93	13.51
SimBA+ [17]	100.0	115.8	10.01	11.64	0.69	100.0	67.4	5.02	9.58	0.36	**100.0**	98.9	5.29	9.76	12.94
Ours-D	**100.0**	**101.6**	**7.38**	**4.77**	**0.54**	**100.0**	**53.9**	**3.89**	**4.58**	0.31	99.9	**81.5**	**3.97**	**4.59**	**8.77**
Ours-P	98.6	105.7	8.03	4.88	0.57	99.7	55.3	4.60	4.77	**0.30**	98.9	86.6	4.51	4.78	9.36

Table 2. Quantitative comparison on ModelNet40 between our shape-invariant black-box attack and different black-box attacks with step size 0.32, on attack success rate (ASR), average query cost (A.Q), Chamfer distance (CD), Hausdorff distance (HD) and average time budget (A.T), where "Ours-D" means choosing DGCNN as surrogate model $\mathcal{H}_w$ and "Ours-P" means choosing PointNet as $\mathcal{H}_w$.

References

[1] Yulong Cao, Chaowei Xiao, Benjamin Cyr, Yimeng Zhou, Won Park, Sara Rampazzi, Qi Alfred Chen, Kevin Fu, and Z. Morley Mao. Adversarial sensor attack on lidar-based perception in autonomous driving. In *the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2019. 1

[2] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 3

[3] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[4] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[5] 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 (ICML)*, 2021. 2, 3

[6] Shixiang Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015. 3

[7] Chuan Guo, Jacob R. Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Q. Weinberger. Simple black-box adversarial attacks. In *International Conference on Machine Learning (ICML)*, 2019. 2, 3

[8] Abdullah Hamdi, Sara Rojas, Ali K. Thabet, and Bernard Ghanem. Advpc: Transferable adversarial perturbations on 3d point clouds. In *16th European Conference on Computer Vision (ECCV)*, 2020. 3

[9] 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. 3

[10] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 2

[11] 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)*, 2020. 1

[12] 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)*, 2019. 2, 3

[13] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[14] Chong Xiang, Charles R. Qi, and Bo Li. Generating 3d adversarial point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[15] Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2, 3

[16] Mutian Xu, Runyu Ding, Hengshuang Zhao, and Xiaojuan Qi. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 3

[17] Jiancheng Yang, Yangzhou Jiang, Xiaoyang Huang, Bingbing Ni, and Chenglong Zhao. Learning black-box attackers with transferable priors and query feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2, 3

[18] Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas J. Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Trans. Graph.*, 2016. 2

[19] Tianhang Zheng, Changyou Chen, Junsong Yuan, Bo Li, and Kui Ren. Pointcloud saliency maps. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 1