APC: Transferable and Efficient Adversarial Point Counterattack for Robust 3D Point Cloud Recognition

Supplementary Material

A. Experimental Settings

In this section, we present implementation details for training our Adversarial Point Counterattack (APC), along with various adversarial attacks and defenses.

A.1. Implementation Details

The default hyperparameters of APC training for PointNet [12] on ModelNet40 (MN40) [18] dataset are summarized in Tab. A. Here, `n_cln` and `n_adv` denote the proportion of clean examples and examples from each attack that are used for APC training, all of which are randomly selected from their respective training set. The `Learning Rate` is set to 1e-5 for PointNet++ [13] and DGCNN [16] on the MN40 dataset. For the ScanObjectNN (SONN) [15] dataset, APC is trained for 100 epochs.

Table A. Default hyperparameters for PointNet on ModelNet40.

Hyperparameters	Value
Num. Point	1,024
α	5.0
β	5.0
n_cln	0.3
n_adv	0.3
Epoch	200
Batch Size	256
Optimizer	Adamw
Learning Rate	0.0001
LR Scheduler	Cosine Scheduler

A.2. Adversarial Attacks

A.2.1. Add

Following [20], we add 100 points and employ the C&W [1] optimization framework with the Chamfer distance [3] as the perturbation metric. We perform a 10-step binary search with 500 iterations per step.

A.2.2. Cluster

Similar to Add, Cluster [20] inserts adversarial clusters instead of points. We add 3 clusters, each containing 32 points. We use the C&W optimization framework with the Chamfer distance as the perturbation metric. We conduct a 5-step binary search with 500 iterations per step.

A.2.3. Perturbation

For the Perturbation, we utilize the C&W optimization framework with the L_2 norm as the perturbation metric. The numbers of binary search steps and iterations are 10 and 500, respectively.

A.2.4. KNN

KNN [14] is another perturbation attack with an additional kNN distance constraint. We perform attack for 2500 iterations.

A.2.5. IFGM and PGD

For the IFGM and PGD [7], we set the number of iterations to 100 and the perturbation budget to 0.5.

A.2.6. HiT

We use the default attack settings described in HiT [9]. The attack is performed with a 10-step binary search and 100 iterations.

A.2.7. Drop

We follow the Drop [22] method by applying a greedy search that recomputes the saliency map over the remaining points at every step, removing the five points with the highest saliency scores in each iteration. In total, we remove 200 points.

A.2.8. AdvPC

For the AdvPC [4], we adopt the C&W optimization framework with the L_2 norm as the perturbation metric, and the numbers of binary search steps and iterations are 2 and 200, respectively.

A.2.9. SI

For the SI [5], we run the attack for 50 iterations.

A.2.10. GeoA3

We implement the GeoA3 [17] using the C&W optimization framework, incorporating the Chamfer distance, Hausdorff distance, and local curvature as the perturbation metrics. It uses a 10-step binary search with 500 iterations.

A.3. Adversarial Defenses

A.3.1. Simple Random Sampling

Simple Random Sampling (SRS) is a defense method that randomly drops several points. We drop 500 points for each point cloud.

A.3.2. Statistic Outlier Removal

Statistic Outlier Removal (SOR) [23] is a defense method that discards points that are considered outliers. It computes the average pairwise distance of each point by leveraging the k-nearest neighbors (kNN). Then, points that exceed a

Table B. Classification results (%) of adversarial examples on ScanObjectNN OBJ_BG split.

Model	Defense	Add	Cluster	Perturb	IFGM	PGD	Drop	AdvPC	Avg.
PointNet	No Defense	0.0	1.0	0.0	4.3	3.9	47.1	27.5	12.0
	SRS	74.3	69.5	73.6	74.1	73.4	46.4	27.5	62.7
	SOR	76.5	79.1	76.0	75.9	77.6	47.3	40.4	67.5
	DUP-Net	74.6	75.0	77.2	76.5	76.9	51.6	59.0	70.1
	IF-Defense	44.7	45.2	46.2	45.7	45.4	38.8	43.0	44.1
	CausalPC	74.4	74.2	72.8	73.5	72.3	52.3	64.7	69.2
	AT	75.9	70.3	76.7	77.6	78.3	55.4	65.0	71.3
	HT	74.0	68.3	75.3	75.3	75.5	**66.9**	65.2	71.5
	APC	**79.0**	**79.3**	**79.7**	**79.2**	**79.7**	62.1	**67.6**	**75.2**
PointNet++	No Defense	83.3	22.3	79.8	38.7	37.5	63.3	17.3	48.9
	SRS	85.3	69.3	85.5	81.0	78.4	65.2	56.6	74.5
	SOR	**87.6**	86.6	85.4	82.9	82.9	65.9	60.2	78.8
	DUP-Net	85.8	85.8	85.3	83.4	82.9	69.1	64.3	79.5
	IF-Defense	51.4	49.5	49.0	49.7	47.1	44.9	40.2	47.4
	AT	83.3	71.0	83.3	76.7	76.0	72.4	62.6	75.0
	HT	80.5	64.3	81.9	77.2	74.3	**75.7**	63.3	73.9
	APC	85.4	**86.7**	**86.1**	**84.3**	**83.8**	67.8	**66.8**	**80.1**
DGCNN	No Defense	4.4	4.4	3.7	23.9	22.5	68.6	1.2	18.4
	SRS	71.7	62.1	67.9	71.9	69.7	41.6	37.8	60.4
	SOR	81.9	82.9	83.4	82.0	82.9	71.7	37.0	74.5
	DUP-Net	55.9	53.7	53.0	54.3	53.0	49.3	40.9	51.4
	IF-Defense	55.2	51.4	52.8	55.0	53.8	45.6	37.6	50.2
	CausalPC	80.0	78.0	79.3	80.2	78.3	66.8	**73.1**	76.5
	AT	82.7	79.1	83.4	84.1	85.0	76.0	62.8	79.0
	HT	81.7	74.1	82.9	**84.3**	84.5	**79.6**	61.4	78.4
	APC	**84.5**	**84.7**	**85.0**	**84.3**	**85.9**	74.5	65.6	**80.6**

Table C. Classification results (%) of adversarial examples on ScanObjectNN PB split.

Model	Defense	Add	Cluster	Perturb	IFGM	PGD	Drop	AdvPC	Avg.
PointNet	No Defense	0.0	0.2	0.0	2.8	2.5	37.4	0.0	6.1
	SRS	65.6	62.0	64.8	66.0	66.3	40.1	24.1	55.6
	SOR	66.2	69.4	69.4	68.4	69.0	41.0	36.7	60.0
	DUP-Net	62.6	63.0	65.0	65.4	64.0	44.1	46.7	58.7
	IF-Defense	43.9	43.5	43.8	43.6	44.7	36.4	38.8	42.1
	CausalPC	60.5	59.6	59.2	58.5	58.6	44.1	52.6	56.2
	AT	69.8	62.5	70.0	70.8	70.6	54.6	55.6	64.8
	HT	67.6	62.6	68.0	68.8	69.4	**58.3**	58.1	64.7
	APC	**71.7**	**72.1**	**72.7**	**72.5**	**72.4**	57.6	**59.2**	**68.3**
PointNet++	No Defense	73.5	30.7	64.8	19.9	17.1	54.9	10.8	38.8
	SRS	75.3	64.9	74.4	69.8	69.5	58.6	49.5	66.0
	SOR	76.8	**77.1**	**77.9**	70.7	71.3	56.3	48.3	68.3
	DUP-Net	75.4	74.8	74.8	72.5	72.2	59.5	54.7	69.1
	IF-Defense	42.2	42.2	42.3	41.1	40.7	37.0	36.8	40.3
	AT	74.8	62.3	72.5	67.3	67.3	62.2	49.4	65.1
	HT	74.6	62.0	72.0	67.9	68.2	65.8	53.6	66.3
	APC	**77.3**	**77.1**	**77.9**	**74.7**	**75.7**	**66.7**	**60.9**	**72.9**
DGCNN	No Defense	5.7	3.7	3.8	21.5	21.0	62.1	1.4	17.0
	SRS	69.3	59.5	61.5	67.2	65.8	37.6	37.0	56.8
	SOR	75.6	76.7	77.2	77.4	78.0	66.0	38.2	69.9
	DUP-Net	56.4	55.1	55.1	56.8	56.4	50.4	44.7	53.6
	IF-Defense	47.9	46.8	47.6	48.4	47.8	43.3	35.2	45.3
	CausalPC	71.6	70.3	70.9	71.5	70.8	60.7	**60.3**	68.0
	AT	77.2	67.9	74.3	78.9	80.1	68.5	53.0	71.4
	HT	76.5	68.5	75.4	**80.2**	80.3	**72.4**	51.5	72.1
	APC	**78.8**	**77.8**	**78.9**	**80.2**	**80.8**	68.6	55.9	**74.4**

Table D. Adversarial accuracies (%) on ScanObjectNN OBJ_BG split in the cross-model transfer setting.

Defense	Source	Target		Source	Target		Source	Target	
	PointNet	PointNet++	DGCNN	PointNet++	PointNet	DGCNN	DGCNN	PointNet	PointNet++
SRS	62.7	74.5	60.4	74.5	62.7	60.4	60.4	62.7	74.5
SOR	67.5	78.8	74.5	78.8	67.5	74.5	74.5	67.5	78.8
DUP-Net	70.1	79.5	51.4	79.5	**70.1**	51.4	51.4	**70.1**	79.5
IF-Defense	44.1	47.4	50.2	47.4	44.1	50.2	50.2	44.1	47.4
APC	**75.2**	**80.7**	**78.3**	**80.1**	67.4	**74.9**	**80.6**	69.2	**79.7**

Table E. Adversarial accuracies (%) on ScanObjectNN PB split in the cross-model transfer setting.

Defense	Source	Target		Source	Target		Source	Target	
	PointNet	PointNet++	DGCNN	PointNet++	PointNet	DGCNN	DGCNN	PointNet	PointNet++
SRS	55.6	66.0	56.8	66.0	55.6	56.8	56.8	55.6	66.0
SOR	60.0	68.3	69.9	68.3	60.0	69.9	69.9	60.0	68.2
DUP-Net	58.7	69.1	53.6	69.1	58.7	53.6	53.6	58.7	69.1
IF-Defense	42.1	40.3	45.3	40.3	42.1	45.3	45.3	42.1	40.3
APC	**68.3**	**71.2**	**72.3**	**72.9**	**64.8**	**71.8**	**74.4**	**63.7**	**70.9**

threshold are removed. The threshold is defined as $\mu+\alpha\cdot\sigma$, where μ and σ are the mean and standard deviation, respectively. We set k and α to 2 and 1.1.

A.3.3. DUP-Net

Following the default implementation of DUP-Net [23], we use PU-Net [21] and set the upsampling rate to 4.

A.3.4. IF-Defense

For IF-Defense [19], we adopt ConvONet [11] as the baseline model, which demonstrates superior performance.

A.3.5. CausalPC

Inspired by randomized smoothing [2, 8] in the 2D domain, CausalPC [6] performs defense by averaging outputs from multiple augmented adversarial examples. We implement it using 1,024 points, following the default settings reported in the original paper.

B. Additional Experimental Results

B.1. 3D Point Cloud Recognition

Tables B and C present the adversarial classification results on the OBJ_BG and PB splits of the ScanObjectNN [15]

Table F. Adversarial accuracy of PD-LTS on PointNet.

Dataset	Add	Cluster	Perturb	KNN	IFGM	PGD	HiT	Drop	AdvPC	SI	GeoA3	Avg.
MN40	74.0	38.6	83.9	7.3	83.8	83.9	8.5	35.3	31.9	21.8	65.7	48.6
ONLY	70.0	49.2	74.0	-	77.4	77.6	-	49.2	21.7	-	-	59.8

Table G. Adversarial accuracy w/o (left) and w/ (right) SOR.

Dataset	DUP	IF	CausalPC	APC
MN40	55.3	77.9	53.6	**83.9**
ONLY	53.5	40.5	39.2	**73.2**

Dataset	AT	HT	APC
MN40	78.5	80.9	**84.7**
ONLY	72.8	72.9	**76.1**

dataset. APC achieves state-of-the-art average robustness for all victim models, demonstrating its effectiveness on both the easier OBJ_BG split and the more challenging PB split.

B.2. Cross-model Transferability

Tables D and E summarize the cross-model transfer experiments on the OBJ_BG and PB splits. We report the average adversarial accuracy computed over all seven attacks. Consistent with the results on the OBJ_ONLY split in Tab. 3, APC shows strong transferability, effectively defending against multiple attacks across various source–target pairs on both splits.

B.3. Effect of SOR

Table G shows the effect of SOR. As all input-level defenses use SOR while model-level do not, we report results of input-level without SOR and model-level with SOR. APC achieves the best performance with and without it.

C. Relation to 3D Point Cloud Denoising

While APC may appear similar to 3D point cloud denoising methods, their objectives are fundamentally different. Denoising focuses on coordinate-level restoration through reconstruction losses, whereas APC is designed for adversarial defense. In particular, APC enforces not only geometric fidelity but also semantic consistency, which clearly distinguishes it from standard denoising approaches.

Importantly, recent adversarial attacks emphasize shape preservation and imperceptibility—properties that conventional denoising methods fail to address. As shown in Table F, PD-LTS [10], a denoising model, exhibits limited adversarial robustness. Its performance degrades significantly under shape-aware attacks such as KNN, HiT, and SI. These results further highlight the fundamental differences between denoising and adversarial purification.

References

[1] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017. 1

[2] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *Proceedings of the 36th International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019. 2

[3] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[4] Abdullah Hamdi, Sara Rojas, Ali Thabet, and Bernard Ghanem. Advpc: Transferable adversarial perturbations on 3d point clouds. In *European Conference on Computer Vision (ECCV)*, pages 241–257, 2020. 1

[5] Qidong Huang, Xiaoyi Dong, Dongdong Chen, Hang Zhou, Weiming Zhang, and Nenghai Yu. Shape-invariant 3d adversarial point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15335–15344, 2022. 1

[6] Yuanmin Huang, Mi Zhang, Daizong Ding, Erling Jiang, Zhaoxiang Wang, and Min Yang. Causalpc: Improving the robustness of point cloud classification by causal effect identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19779–19789, 2024. 2

[7] Daniel Liu, Ronald Yu, and Hao Su. Extending adversarial attacks and defenses to deep 3d point cloud classifiers. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 2279–2283, 2019. 1

[8] Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble. In *Computer Vision – ECCV 2018*, pages 381–397, 2018. 2

[9] Tianrui Lou, Xiaojun Jia, Jindong Gu, Li Liu, Siyuan Liang, Bangyan He, and Xiaochun Cao. Hide in thicket: Generating imperceptible and rational adversarial perturbations on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24326–24335, 2024. 1

[10] Aihua Mao, Biao Yan, Zijing Ma, and Ying He. Denoising point clouds in latent space via graph convolution and invertible neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5768–5777, 2024. 3

[11] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European Conference on Computer Vision (ECCV)*, pages 523–540, 2020. 2

[12] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference*

on Computer Vision and Pattern Recognition (CVPR), pages 652–660, 2017. 1

[13] 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*, 2017. 1

[14] Tzungyu Tsai, Kaichen Yang, Tsung-Yi Ho, and Yier Jin. Robust adversarial objects against deep learning models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:954–962, 2020. 1

[15] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 1, 2

[16] 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 Trans. Graph.*, 38(5), 2019. 1

[17] Yuxin Wen, Jiehong Lin, Ke Chen, C. L. Philip Chen, and Kui Jia. Geometry-aware generation of adversarial point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):2984–2999, 2022. 1

[18] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1

[19] Ziyi Wu, Yueqi Duan, He Wang, Qingnan Fan, and Leonidas J. Guibas. If-defense: 3d adversarial point cloud defense via implicit function based restoration, 2021. 2

[20] Chong Xiang, Charles R. Qi, and Bo Li. Generating 3d adversarial point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[21] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-net: Point cloud upsampling network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

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

[23] Hang Zhou, Kejiang Chen, Weiming Zhang, Han Fang, Wenbo Zhou, and Nenghai Yu. Dup-net: Denoiser and upsampler network for 3d adversarial point clouds defense. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 1, 2