Benchmarking and Analyzing Robust Point Cloud Recognition: Bag of Tricks for Defending Adversarial Examples

Qiufan Ji[1], Lin Wang [2,3,*], Cong Shi[1], Shengshan Hu[4], Yingying Chen[5], Lichao Sun[6] *
[1]New Jersey Institute of Technology, [2]AI Thrust, HKUST(GZ), [3]Dept. of CSE, HKUST, [4]HUST,
[5]Rutgers University, [6]Lehigh University
qj39@njit.edu, linwang@ust.hk, cong.shi@njit.edu, hushengshan@hust.edu.cn,
yingche@scarletmail.rutgers.edu, james.lichao.sun@gmail.com

Abstract

Deep Neural Networks (DNNs) for 3D point cloud recognition are vulnerable to adversarial examples, threatening their practical deployment. Despite the many research endeavors have been made to tackle this issue in recent years, the diversity of adversarial examples on 3D point clouds makes them more challenging to defend against than those on 2D images. For examples, attackers can generate adversarial examples by adding, shifting, or removing points. Consequently, existing defense strategies are hard to counter unseen point cloud adversarial examples. In this paper, we first establish a comprehensive, and rigorous point cloud adversarial robustness benchmark to evaluate adversarial robustness, which can provide a detailed understanding of the effects of the defense and attack methods. We then collect existing defense tricks in point cloud adversarial defenses and then perform extensive and systematic experiments to identify an effective combination of these tricks. Furthermore, we propose a hybrid training augmentation methods that consider various types of point cloud adversarial examples to adversarial training, significantly improving the adversarial robustness. By combining these tricks, we construct a more robust defense framework achieving an average accuracy of 83.45% against various attacks, demonstrating its capability to enabling robust learners. Our codebase are open-sourced on: `https://github.com/qiufan319/benchmark_pc_attack.git`.

1. Introduction

As an prominent form of 3D data representation, point clouds are extensively employed in various real-world sensing applications, such as autonomous driving [36], robotics [14], and healthcare [1]. To achieve precise perceive 3D objects, prior studies [15, 16, 29] have investigated the development of deep neural networks (DNNs) capable of detecting, segmenting, and identifying objects from point cloud data. While these DNN-based methods have exhibited notable success, recent studies have exposed their susceptibility to adversarial examples [33, 38, 30]. Specifically, the addition, removal, or shifting of a small proportion of 3D points from an object can significantly degrade the performance of the DNNs.

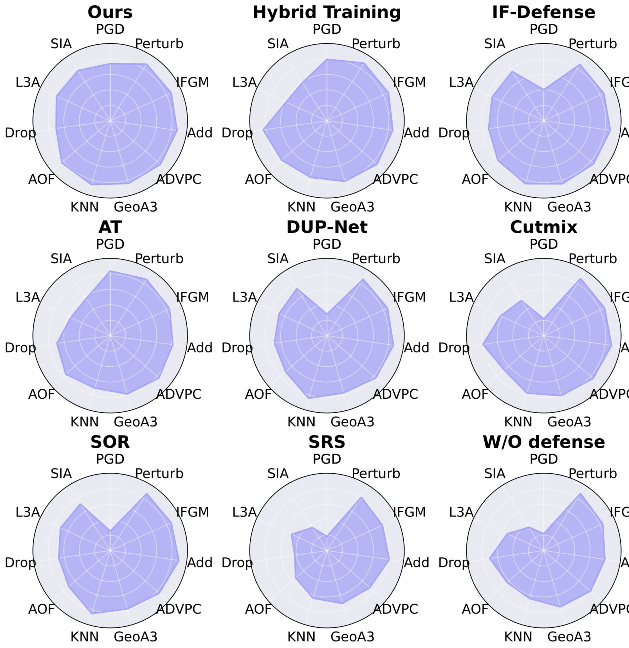

Figure 1. Point Cloud defense's adversarial robustness to various attacks in a radar chart. We evaluate the defense under 9 attack methods, including PGD [9], SIA [7], L3A [23], Drop [38], AOF [8], KNN [26], GeoA3 [30], AdvPC [5], Add [33], IFGM [9], Perturb [33]. Our method achieve good adversarial robustness against all attacks.

To mitigate the risk of adversarial examples, several de-

*L. Wang and L. Sun are the corresponding authors.

fense strategies have been proposed to enhance the robustness of point cloud DNNs [32, 39, 9]. For example, preprocessing techniques are applied to remove the points perturbed by adversarial examples [39, 32]. In addition, adversarial training [9, 27, 11], which incorporates adversarial examples in the model training process, is designed to improve the adversarial robustness.

Despite the initial success of investigating the adversarial robustness of point cloud DNNs, there are three obvious limitations for existing attacks and defenses:

L1: Unrealistic attack and defense scenarios. The current state-of-the-art (SOTA) adversarial learning has primarily focused on wihite-box attacks and defenses [33, 30, 39], where the attacker has complete knowledge of the model architecture and paramteres. While these scenarios are useful for testing the limits of existing methods and understanding their vulnerabilities, they do not reflect the real-world security threat landscape. In many security-critical applications, such as autonomous driving and financial systems, attackers may not access to the model parameters.

L2: Lack of a unified and comprehensive adversarial robustness benchmark. While several studies [18, 28, 25, 19] have been proposed to evaluate the robustness of point cloud DNNs, they have are all focused on benchmark under diverse types of corruptions. However, existing benchmarks research for studying adversarial robustness remains unexplored. Compared with the corruption-oriented attack methods, adversarial examples are difficult to be detected by both humans and machines. Moreover, perturbation generated using gradient descent are more effective than random corruptions, resulting in higher error rate and better imperceptibility. Despite recent studies exploring adversarial examples and defense on point cloud DNNs [33, 26], most of them have employ substantially different evaluation settings such as datasets, attacker's capability, perturbation budget, and evaluation metrics. The lack of a unified evaluation framework makes it challenging to fairly quantify the adversarial robustness. Additionally, current adversarial robustness evaluations only focus on one or a few attacks, defenses, and victim models, limiting the generalization and comparability of the results. For instance, the effectiveness of point cloud attack methods [33, 30] is typically evaluated under a limited set of defenses and models. Similarly, defense strategies are often evaluated against only a few early attacks, making it difficult to capture their strengths and weaknesses based on incomplete evaluations.

L3: Poor generalization of defense strategies. Differ 2D image attacks that modify the pixel value in a fixed data size, the adversarial example on point cloud offer a wider attack space and arbitrary data size. For instance, an attacker can generate adversarial example by adding, shifting, or removing points on the original point cloud. Unfortunately, most of existing defense strategies only consider one or two types, which can not handle unseen adversarial examples.

In this paper, we propose the first comprehensive and systematic point cloud adversarial robustness benchmark. Our benchmark provides a unified adversarial setting and comprehensive evaluation metrics that enable a fair comparison for both attacks and defenses. By analyzing the quantitative results, we propose a hybrid training strategy and construct a more robust defense framework by combining effective defense tricks. Our main contributions are summarized below:

1) Pratical Scenario. To evaluate the real-world performance of attacks and defenses, we refine the capability of both the attacker and defender, For example, we limited the maximum number of points added and the knowledge level of the attacker and defender. In our benchmark, all attackers are processed under the black-box setting, where the attacker does not have any additional knowledge about the model parameters, model structure, and training dataset.

2) Unified Evaluation Pipeline. Our benchmarks provide a comprehensive and standardized evaluation methodology, enabling fair comparison and reproducibility of the proposed methods. For example, we evaluate the attack from attack success rate, transferability, and imperceptible, which are essential metrics for assessing the effectiveness, imperceptibility, and generalization of the attacks.

3) Bag-of-tricks for Defending Adversarial Examples. Based on our adversarial robustness analyses with our benchmark, we proposed a hybrid training approach that jointly consider different types of adversarial examples, including adding, shifting, and removing points, to perform adversarial training. Through analysis of experiment result, we further construct a more robust defense framework by combining the effective defense tricks. As shown in Figure 1, our framework achieve the SOTA adversarial robustness under various attacks.

2. Related works

3D Point Cloud Recognition. In contrast to 2D image data, 3D point cloud data is irregular and unordered, making it hard to be consumed by the neural networks designed for the 2D domain. PointNet [15] is the pioneering work that directly consumes point cloud. It achieves permutation invariance of points by learning each point independently and subsequently using a symmetric function to aggregate features. Due to its high accuracy and efficiency, it has been widely used as the backbone for 3D point cloud recognition. As the update of PointNet, PointNet++ [16] improves point cloud learning by capturing local information from the neighborhood of each point. Another representative work is DGCNN [29], which enhances the representation capability by building neighbor graphs between adjacent points and using a convolution-like operation (EdgeConv) on each connecting edge to capture local information. Recently, some transformer-based methods [12, 4, 34] have been proposed, achieving good performance.

Robustness Benchmark for Point Cloud. Several benchmarks [21, 20, 2, 22, 6] have been built for studying the robustness of point cloud learning. [17] build a real-world dataset to evaluate the gap between simulation and real-world. To evaluate the corruption robustness, ModelNet-C [19] categorizes common corruptions and builds a new corruption dataset ModelNet-C by corrupting the ModelNet40 test set with simulated corruptions like occlusion, scale, and rotation. RobustPointset [25] evaluates the robustness of point cloud DNNs and shows that existing data augmentation methods can not work well to unseen corruptions. However, little attention has been paid to adversarial examples of point cloud recognition. In this paper, we aim to present the first comprehensive, systematic benchmark to evaluate the point cloud adversarial examples and defenses.

3. Benchmark

3.1. Preliminaries

Problem Formulation. We defined the point cloud as $X \in \mathbb{R}^{N\times 3}$, where N is the number of points . Each point $x_i \in \mathbb{R}^3$ indicates the 3D coordinate (x_i, y_i, z_i). Formally, a classifier $f_\theta(X) \to Y$ maps the input point cloud X to its corresponding label $y \in Y$ with parameter θ. For adversarial examples on point cloud DNNs, an attacker generates an adversarial example $\hat{X}$, which makes the classifier f_θ output an incorrect label $\hat{Y}$. Generally, the objective function of generating adversarial examples can be formulated as:

$$\min D(X, \hat{X}), \qquad \text{s.t. } f_\theta(\hat{X}) = \hat{Y}, \tag{1}$$

where $D(\cdot, \cdot)$ is the distance function measuring similarity between X and $\hat{X}$. The distance is normally constrained to a small budget ρ ensuring the imperceptibility. Because the equation (1) is non-convex, according to [33] we reformulated it as gradient-based optimization algorithms:

$$\min f_{adv}(X, \hat{X}) + \lambda * D(X, \hat{X}) \qquad \text{s.t. } D(X, \hat{X}) < \rho, \tag{2}$$

where f_{adv} is the adversarial loss function, including logits loss and cross-entropy loss, and λ is a hyperparameter to balance distance and adversarial loss.

Attack Types. An attacker can have different targets of generating adversarial examples. In our benchmark, we divided the attacks into targeted and untargeted. Targeted: A targeted attack tries to make the victim model outputs a result that it desired, as $f_\theta(\hat{X}) = Y^*$, where y^* is the target label. Untargeted: an untargeted attack only aims to make the victim model outputs a wrong result, as $f_\theta(\hat{X}) \neq Y$, where Y is the true label.

Attack Knowledge. The attacker can have different levels of knowledge of the victim model. Based on the knowledge level, the attacks can be divided into Black-Box and White-Box. Black-Box: The attacker can not get any information about the victim model, such as gradient information, model structure, and parameters. However, they have limited chances to query the victim model and obtain the output. White-Box: The attacker can get any information about the victim model. In both knowledge settings, the attacker can access the training dataset.

Attack Stage. Based on the stage where the attacks happened, we divided the attacks into Poisoning and Evasion. Poisoning: The attacker generate the adversarial examples and inject them into the training dataset. Once the attacker change the training dataset, the victim model will be retrained on changed dataset to get a worse model. Evasion: The parameter of the victim model is fixed, and attackers inject adversarial perturbation into testing data.

3.2. Practice Scenario

In real-world, the victim model is usually trained in a confidential manner, and the attacker is hard to modified the model meaning that white-box and poisoning setting are normally infeasible.

In our benchmark, we make the following assumptions for a unified and practical adversarial robustness evaluation protocol: (1) Black-box: the attacker does not know the defender's strategies, and vice versa. (2) Evasion: The point cloud DNNs are trained with trusted data and training model is inaccessible to the attacker. (3) Untargeted: in our benchmark, we select untargeted attacks for the evaluation of adversarial robustness. Because untargeted attack is easier than targeted attacks for attacker, thus the untargeted attack is the upper bound of attack intensities and more difficult for defense strategies. We define the full capabilities of attackers and defenders in our benchmark:

Attacker: 1) The attacker can access the testing point cloud data to produce adversarial examples, but they should not have knowledge about the victim model or defense mechanism. 2) To preserve adversarial examples imperceptible, the attacker is only allowed to add or delete a limited number of points in the point cloud. 3) The attacker can not modify the training dataset. 4) The attacker can only query the victim model with limited times.

Defender: 1) The defender has full access to the training dataset. 2) The defender can use any solution to improve the robustness without additional knowledge about the attack.

Both sides: We assume that attackers know the architecture of victim model (e.g., PointNet, PointNet++), and then they can train a corresponding surrogate model to generate adversarial examples. Similarly, the defender can have some assumptions on the effects of adversarial examples (e.g., point cloud adversarial examples usually exist some outliers). For both the attacker and the defender, the generalization (e.g., an attack can bypass multiple defense techniques) is an important factor for adversarial robustness quantifica-

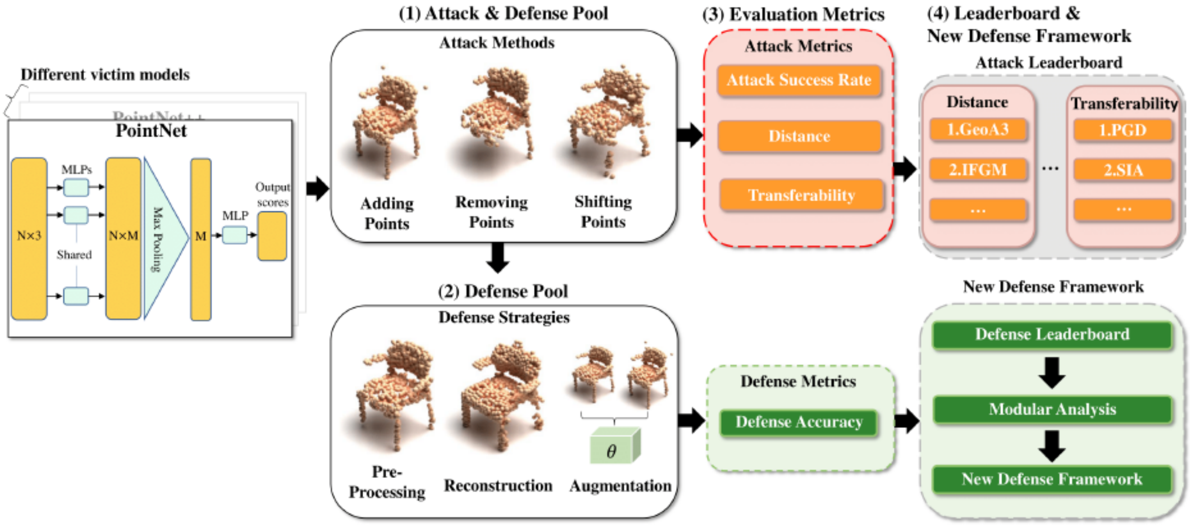

Figure 2. The pipeline of our benchmark.

tion. Thus, we evaluate the effectiveness of SOTA attacks against various defense techniques and model architecutres. We also conduct similar quantifications for the defenses in our benchmark.

By following the above rules, we provide a unified evaluation scenario for attacks and defenses in a principled way. It is worth nothing that the unified scenario is not the only valid ones, our benchmark will include more scenarios as this field evolved over time. As shown in Figure 2, the attack and defense pool include all attack methods and defense strategies in our benchmark. Our evaluation metrics incorporate three attack metrics, namely, attack success rate, distance, and transferability, to assess the performance of attack methods. Additionally, we use one defense accuracy metric to evaluate the effectiveness. We construct attack and defense leaderboards based on the metrics values. Further, we conduct modular analysis on each defense strategy, and subsequently integrate effective modules to construct a more robust defense framework.

3.3. Generating Adversarial Examples

Adversarial examples were first discovered by [24] in 2D image classification tasks. With the development of adversarial learning, some works [33, 7, 30] proved that point clouds also be vulnerable to adversarial examples. The adversarial examples on point cloud can be divided into adding points, removing points, and shifting points attacks.

Adding Points. The attacker generate adversarial examples by adding a set of adversarial points $Z \in \mathbb{R}^{k\times 3}$ where k is the number of modified points in different attack settings. Given the adversarial perturbations $\rho \in \mathbb{R}^{k\times 3}$ on added points, the objective function of adding points attacks can be formulated as:

$$\min f_{adv}(X, X \cup (Z+\rho)) - \lambda D(X, X \cup (Z+\rho)), \quad (3)$$

Adding independent points [33] chooses the critical points that are still active after max-pooling operation, as the initialized positions, and then uses C&W [3] to output their final coordinates. Although other adding points attacks exist, such as add clusters [33] and adversarial sticks [10]. These methods are not practical because they create a noticeable continuous deformation and then produce large perturbations. Consequently, for the purpose of adding points attack, only independent points are considered.

Removing Points. The attacker remove some points to spoof the classifier. As the representative work of removing points attack, saliency map [38] constructs a saliency map to measure the contribution of each point and then removes the points based on the saliency score. In our benchmark, we limit the number of dropped points to keep the drop attack imperceptible.

Shifting Points. The attacker perturbs the coordinates of a set of points to implement an attack. The objective function of shifting points attacks can be formulated as:

$$\min f_{adv}(X, (X+\rho)) - \lambda D(X, (X+\rho)), \quad (4)$$

The iterative fast gradient method (IFGM) [9] is an extension of the fast gradient method (FGSM) that repeats FGSM multiple times to generate better adversarial examples. The project gradient descent method (PGD) [9] projects the perturbed point onto the triangle plane where the points are sampled. Perturb [33] proposes a C&W based algorithm to generate adversarial examples. To reduce the outliers, KNN [26] incorporates Chamfer measurement and KNN distance to encourages the compactness of local neighborhoods in the generated adversarial examples. GeoA3 [30] perturbs points in a geometrically aware way by adding local curvatures to the loss function, thereby making the adversarial examples more imperceptible. L3A [23] proposes a novel optimization method to avoid local optima, making the attack more efficient. AdvPC [5] utilizes a point cloud auto-encoder (AE) during the generation, improving the transferability of adversarial examples. SIA [7] builds a tangent plane to each point and limits the point perturbation along the optimal direction on the tangent plane, making the adversarial examples more imperceptible. AOF [8] proposes a more robust and transferable attack by perturbing the low-frequency in the frequency domain.

4. Analysis and Bag-of-Tricks for Defending Adversarial Examples

To alleviate the adversarial behaviors, the most popular defending techniques can be divided into three directions, i.e., pre-processing, reconstruction, and augmentation, as shown on Figure 3.

Pre-processing. Advanced pre-processing aims to reduce the noise before inference. A straightforward approach is Simple Random Sampling (SRS), which random samples a subset of points from the original point cloud as input. Statistical Outlier Removal (SOR) [39] computes KNN distance and removes the points with a large KNN distance.

Reconstruction. Adversarial examples often result in the absence or alteration of geometric features in the original point cloud. With the development of 3D point cloud reconstruction, some works employed 3D reconstruction networks to improve robustness. We consider two 3D Reconstruction networks in our benchmark:

DUP-Net [39]: DUP-Net employs the PU-Net [35] as its reconstruction network. The PU-Net utilizes point cloud up-sampling to reconstruct the surface and generate a high-resolution point cloud that captures the missing structures of the object's surface. More experiment results of DUP-Net can be found in the appendix.

IF-Defense [32]: In contrast to DUP-Net, IF-Defense employs the ConvONet [31] as its reconstruction network. The ConvONet uses the point cloud as input for shape latent code extraction, while the encoder produces a learned implicit function network. By pre-training the implicit model on clean data, the decoder's output space situates on the accurate and complete shape manifold.

We present the results of the adversarial robustness evaluation of IF-Defense and ConvONet in the appendix. We find that both reconstruction networks can improve adversarial robustness. Especially, ConvONet, with its superior 3D reconstruction performance, exhibits better adversarial robustness in most attacks.

Augmentation. The principle of augmentation is aimed at enhancing the robustness of the model when encountering minor noise. One notable approach is adversarial training [9], which incorporates adversarial examples into the training phase. Another augmentation method is PointCutmix [37], which utilizes mix-based point cloud to enhance the model's robustness. However, due to the variety of attack types, existing augmentation methods performance poorly against adversarial attacks in point cloud.

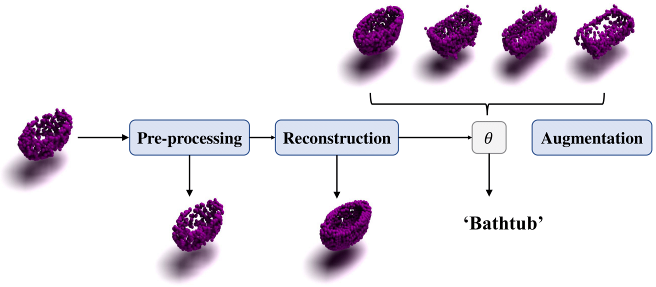

Figure 3. Robust defense framework paradigm. The adversarial robustness of point clouds against various attacks is influenced by three critical components, including pre-processing, reconstruction, and augmentation methods.

Table 1. The accuracy of defense strategies.

	AOF	GEOA3	SIA
w/o defense	54.54	61.26	31.40
SOR (Pre-processing)	68.48	73.22	59.12
IF-Defense (Reconstruction)	66.99	65.68	43.76
Hybrid Training (Augmentation)	73.43	75.45	76.26

Hybrid Training. To enhance the performance of augmentation, we propose a hybrid training method that leverages multiple attack approaches. Especially, hybrid training selects attack approaches, including adding, removing, and shifting attacks. For each class in the dataset, the proposed method divides the samples equally into parts and applies different attack approaches to each part. Finally, all generated adversarial examples are integrated to augment the training data. The result of adversarial robustness of augmentation methods is reported in the appendix. Our hybrid training achieves the highest level of adversarial robustness among all augmentation methods.

In Table 1, we show the accuracy of defense strategies. We find the three components can improve the robustness of adversarial robustness. Based on the aforementioned analyses, we propose a robust defense framework that integrates SOR, hybrid training, and ConvONet. In Table 2, our defense framework demonstrates superior robustness compared to other existing defense strategies. Moreover, in the ablation study presented in the appendix, we demonstrate that all aforementioned modules contribute significantly to the adversarial robustness of our defense framework, with hybrid training being the critical component for enhancing adversarial robustness. These results substantiate our conclusions from the modular analysis and establish our framework as a solid baseline for future research on adversarial robustness.

Table 2. Leaderboard. Bold: best in column. Underline: second best in column. Blue: best in row. Red: worst in row. Compared with existing augmentation methods, our hybrid training achieves the SOTA performance.

Defense & (Acc)	Model	Clean	PGD	SIA	L3A	Drop	AOF	KNN	GeoA3	AdvPC	Add	IFGM	Perturb
Ours (**83.45**)	PointNet	87.36	76.70	76.26	75.04	80.79	81.60	85.53	84.04	86.22	87.28	86.26	86.35
	PointNet++	88.65	71.56	78.61	76.70	83.14	84.12	86.87	85.78	**87.40**	88.45	88.17	88.70
	DGCNN	87.97	72.93	81.81	77.96	82.94	84.08	86.39	84.81	86.71	88.17	87.76	87.32
	Pointconv	87.12	70.14	80.67	76.50	83.79	81.77	86.30	85.53	86.83	87.88	87.52	86.95
	PCT	83.27	73.91	78.32	75.77	76.86	79.66	82.41	80.71	81.60	82.78	82.21	83.71
	Curvenet	88.57	76.13	80.26	78.16	83.67	83.59	87.20	84.97	86.14	88.05	86.99	87.88
	RPC	88.86	74.11	**82.66**	**78.20**	82.98	**84.48**	**87.93**	85.49	87.38	88.01	88.45	88.13
	GDANet	88.57	75.32	80.79	77.31	83.59	83.75	86.59	84.44	86.63	88.65	86.95	87.28
Hybrid Training (79.35)	PointNet	88.57	80.15	53.08	50.28	77.55	73.34	64.71	75.45	84.12	83.39	85.17	87.64
	PointNet++	89.75	77.39	52.80	57.74	85.74	79.85	74.68	82.74	86.08	85.45	88.29	89.00
	DGCNN	89.47	**81.40**	66.29	61.14	86.14	80.19	80.59	82.74	87.28	87.76	88.53	89.95
	Pointconv	90.19	80.06	45.30	57.37	**86.47**	69.65	81.69	83.31	85.13	88.82	**89.83**	**90.28**
	PCT	87.44	45.95	75.97	76.70	72.69	78.57	85.86	83.02	86.35	87.12	87.86	87.24
	Curvenet	87.16	44.57	76.22	76.00	74.15	81.48	85.45	83.43	86.14	87.40	87.84	86.35
	RPC	85.45	53.85	76.46	74.80	70.30	80.31	83.95	82.17	84.60	84.52	85.90	84.68
	GDANet	87.24	38.74	79.01	75.04	72.16	80.71	85.41	82.86	85.21	86.43	86.87	86.67
IF-Defense (78.4)	PointNet	85.33	44.89	68.60	68.76	65.19	75.28	82.46	82.74	83.79	85.41	82.86	85.01
	PointNet++	87.52	38.61	72.45	72.33	73.01	78.28	85.56	85.17	85.66	87.20	85.37	87.12
	DGCNN	87.88	40.32	77.92	76.05	71.48	80.35	85.78	85.41	85.49	86.75	85.86	87.32
	Pointconv	85.53	28.61	78.24	73.66	73.87	75.89	84.85	84.24	84.48	85.58	84.20	85.37
	PCT	88.33	45.83	75.24	73.45	72.85	79.42	85.86	85.29	86.35	87.16	85.94	86.83
	Curvenet	88.33	45.38	76.18	75.45	74.19	80.51	85.45	86.02	86.14	88.01	86.75	86.67
	RPC	88.05	40.44	76.13	73.62	73.58	77.43	83.95	**86.06**	84.60	87.72	85.90	87.64
	GDANet	87.93	38.05	81.65	75.45	72.37	80.92	85.41	85.94	85.21	87.72	86.10	86.87
SOR (75.19)	PointNet	86.95	42.10	63.21	63.70	57.86	68.48	80.06	73.22	80.49	86.10	84.16	85.53
	PointNet++	88.57	25.00	64.30	72.49	66.25	71.31	85.13	80.23	84.89	88.70	87.72	88.98
	DGCNN	88.57	18.00	73.58	69.89	66.94	66.25	85.25	74.24	82.33	87.88	87.64	87.44
	Pointconv	72.12	11.70	71.84	69.73	72.12	65.92	85.53	77.47	84.68	88.49	86.02	87.12
	PCT	88.41	38.94	72.97	70.75	67.50	74.84	85.01	80.31	84.52	88.65	86.79	87.76
	Curvenet	88.33	33.63	74.51	76.86	69.81	76.62	86.43	83.39	85.17	**89.10**	86.83	87.72
	RPC	89.43	15.07	72.57	69.17	69.00	69.85	85.53	78.04	84.04	88.86	87.44	87.72
	GDANet	89.26	17.34	79.90	72.12	68.40	74.39	86.59	82.29	85.94	88.85	88.01	87.88
No Defense (67.06)	PointNet	87.64	34.32	31.40	45.38	59.64	54.54	45.10	61.26	76.94	71.76	74.59	85.58
	PointNet++	89.30	15.56	16.82	44.89	71.47	60.01	54.25	74.51	73.62	72.37	81.22	88.17
	DGCNN	89.38	18.96	51.01	57.25	73.10	62.84	70.10	77.39	76.86	83.71	86.91	88.74
	Pointconv	88.65	9.81	25.41	47.57	76.50	51.26	71.80	77.67	76.90	85.15	86.51	88.09
	PCT	89.99	32.33	41.05	53.75	71.27	65.92	67.08	78.32	82.58	82.33	85.62	89.22
	Curvenet	89.47	27.96	38.70	53.53	71.29	69.52	66.73	79.17	84.48	79.86	85.09	88.33
	RPC	89.42	15.36	32.33	54.01	69.89	72.16	70.58	77.92	83.67	80.75	83.55	85.98
	GDANet	89.10	20.71	50.57	59.64	72.33	67.30	72.20	80.92	85.13	83.47	86.63	88.33
Avg.ASR		-	51.84	36.15	33.85	27.19	25.41	20.59	18.83	15.77	14.26	13.98	12.49

5. Experiments

5.1. Experimental Setup

Dataset and DNNs. All of our experiments are conducted commonly on ModelNet40 dataset. ModelNet40 consists of 123,11 CAD models for 40 object classes. In our experiments, we split ModelNet40 dataset into two parts: 9,843 and 2,468 samples for training and testing, respectively. Following [16], we use farthest points sampling (FPS) to uniformly sample 1024 points from the surface of each object as input data. We adopt eight widely used point cloud DNNs as victim models, including PointNet [15], Pointnet++ [16], DGCNN [29], PointConv [31], PCT [4], Curvenet [12], PRC [18], and GDANet [34]. For PointNet++ and PointConv, we use the single scale grouping (SSG) strategy. All models are trained without data augmentation.

Attack Settings. According to the attacker capability setting, we implemented all attacks on the testing dataset and using a PointNet model as surrogate model. It should be note that the hyperparameters of the surrogate model differed from those of the victim models. Specifically, We employed 11 different attacks. In the adding points attack [33], we added 100 points, and in the removing points attack [38], we removed 200 points. Regarding the shifting points attack, our method including SIA [7], L3A [23], KNN [26], GeoA3 [30], IFGM [9], PGD [9], Perturb [33], AOF [8] and AdvPC [5]. To ensure fair verification, we constrained all Shifting points adversarial examples equally using an normal ball with a radius of 0.16, and we performed untargeted attacks under the same setting.

Defense Settings. For SRS [39], we randomly dropped 500 points from the input point cloud. To perform SOR [39], we first computed the average distance from its k-nearest neighbors and subsequently removed points if the average distance exceeded the threshold of $\mu + \alpha \cdot \sigma$, where μ and σ are the mean and standard deviation, respectively, and k and α are hyperparameters. We set the hyperparameters to be $k = 2$ and $\alpha = 1.1$. For IF-Defense [32], we chose ConvOnet [13], which achieved the superior performance for most attacks in their experiment results. In adversarial training, all victim models were trained on clean data and adversarial examples generated by PGD with $l_\infty = 0.20$. For hybrid training, we combined adding independent points, saliency map, and PGD, with adversarial training. In the adding independent points attack, we added 200 points to point cloud. In the saliency map attack, we removed 300 points form the point cloud based on their saliency map. In PGD, we set the perturbation budget to $l_\infty = 0.20$.

Evaluate Metrics. To evaluate the imperceptibility of generated adversarial examples, we adopt Chamfer Distance (CD) and Hausdorff Distance (HD) as distance metrics for each adversarial example in our study. (1) HD: measures the distance between two points clouds in a metric space by computing the nearest original point for each adversarial point and outputting the maximum square distance among all nearest point pairs, as shown below:

$$\mathcal{D}_H(X, \hat{X}) = \min_{x \in X} \max_{\hat{x} \in \hat{X}} \|x - \hat{x}\|_2^2, \tag{5}$$

(2) CD: CD is similar to HD but takes an average rather than a maximum, It defined as:

$$\mathcal{D}_C(X, \hat{X}) = \frac{1}{\|\hat{X}\|_0} \sum_{\hat{x} \in \hat{X}} \min_{x \in X} \|x - \hat{x}\|_2^2. \tag{6}$$

Moreover, for generating adversarial example methods, we use attack success rate to evaluate their effusiveness. (3) Attack Success Rate (ASR): it computes the attack success rate against defense strategies. For defense strategies, we use defense accuracy (ACC) to evaluate their adversarial robustness. (4) ACC: it measures the accuracy of defense strategies against attack methods.

5.2. Experimental Results

Point Cloud Adversarial Robustness leaderboard. Following the process of Figure 2, we evaluate the performance of attacks vs. defenses. An illustrated example of leaderboard for point cloud adversarial robustness is presented in Table 2, where the attacks and defenses are ranked based on their respective average attack success rate and average defense accuracy.

1) The effectiveness of defense strategies may can vary depending on the models and attacks they are applied to. In

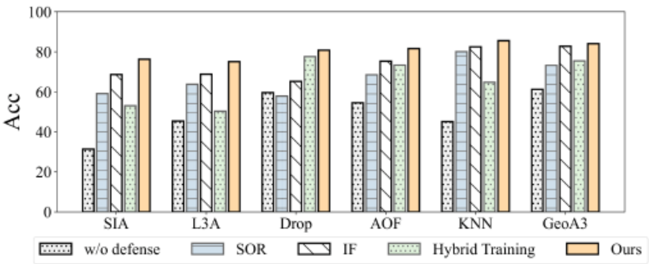

Figure 4. Adversarial robustness of 5 defense strategies under SIA, L3A, Drop, AOF, KNN, and GeoA3 attacks with PointNet.

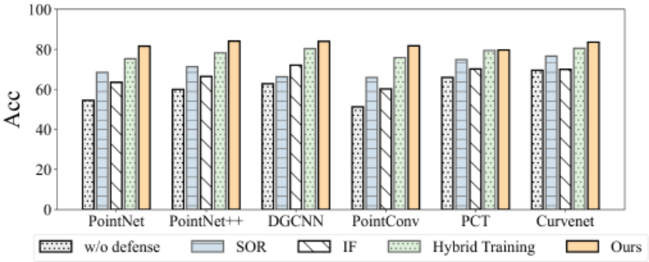

Figure 5. Adversarial robustness of 5 defense strategies under AOF attack with different victim models.

Figure 4 We examine the adversarial robustness of 5 defense strategies across various attacks. Our result reveal that while hybrid training exhibits a high defense accuracy against SIA, Drop, and AOF attacks, it performs poorly against KNN and L3A attacks. In addition, we explore the defense accuracy of defense strategies with different victim models under same attack, as depicted in Figure 5. We find that IF-Defense has a large defense performance gap between PointConv and DGCNN. To obtain more convincing results, we recommend that researchers comprehensively evaluate the adversarial robustness of defense strategies by subjecting them to a wide spectrum of attacks and victim models. Such evaluations are essential for accurately evaluating the generalization capabilities of defenses and promoting their practical viability.

2) Among current point cloud DNNs, it has been observed that models incorporating advanced grouping operations, such as Curve grouping in Curvenet and frequency grouping in GDANet and RPC, exhibit superior performance against various attacks. This performance superiority can potentially be attributed to the high expressiveness of these models' architectures.

3) Some defense methods, such as SOR, show worse performance than No defense model. There are two reasons to conduct this phenomenon. For attack, some early attacks (e.g., Pertub and IFGM) exhibit poor transferability. Thus, training settings differences in the target model can degrade the ASR. For defense, some defense methods modifying the shape of point cloud (e.g., SOR and ConvONet) also impacted the classification performance. In some cases, these defensive modifications may degrade the model performances more significantly than the early adversarial attacks, resulting in worse performance than No defense.

The complete leaderboard is provided in the appendix. The leaderboard is dynamic and subject to modification with the advent of more potent attacks, defenses, or models. We will analyze the effectiveness, transferability, and imperceptible of adversarial examples in the following.

Attack Effectiveness. In Table 2, we observe the effectiveness of adding points attack is considerably low, indicating that adding point attack poses a significant challenge in affecting the performance of existing models. Furthermore, the average success rate of most shifting points attacks is below 25%, implying that the majority of existing shifting attacks fail to significantly degrade point cloud DNNs. It indicates most of the previous works may not be applicable in real-world. Therefore, future research should priories designing more practical attack methods that take into account real-world situations.

Attack Transferability and Imperceptibility. In the benchmark, attackers do not have knowledge about the victim model, which makes the transferability of adversarial examples crucial. To evaluate the transferability of adversarial examples, we selected three wildly-used point cloud DNNs, including PointNet, PointNet++, and DGCNN as the surrogate model. Adversarial examples generated on these surrogate models were tested on all victim models. The transferability results are presented in the appendix. All adversarial examples are tested without any defense strategies, and the transferability was ranked based on the average attack success rate. Furthermore, we evaluate the imperceptibility of adversarial examples by calculating the Hausdorff distance and Chamfer measurement, respectively. The imperceptibility results are also presented in the appendix. We ranked the adversarial examples based on the average distance of Hausdorff distance and Chamfer measurement. After observing the transferability and imperceptibility results, we identified several good imperceptible adversarial examples, such as GeoA3, IFGM, Perturb, and Add, with poor transferability, indicating a trade-off between imperceptibility and transferability. Therefore, how to balance the transferability and imperceptibility of adversarial examples is a potential research direction.

5.3. Ablation Study and New Findings

In this section, we present an ablation study of our proposed defense framework, as illustrated in Figure 6. Specifically, we conduct experiments by selectively removing individual defense components and evaluating the resulting adversarial robustness against adversarial examples, such as AOF, GeoA3, and SIA. From the results, we demonstrate that all modules within our defense framework significantly contribute to the overall robustness of the system. Meanwhile, each module has different effectiveness for robustness. For example, Hybrid training combined with SOR defense can achieve almost the same performance as all modules, but SOR plus ConvOnet gets the lowest defense performance, which reveals the significance of Hybrid training.

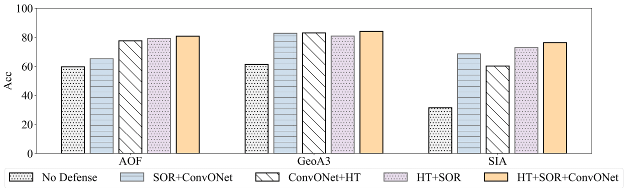

Figure 6. The ablation study of our new defense framework. All attacks are generated on PointNet. HT: hybrid training.

Our New Findings. We present new findings on the transferability of adversarial examples in 3D point cloud DNNs. Table 2 and the transferability results in the appendix show that the transferability of point cloud adversarial examples is limited compared with 2D adversarial examples. This limitation can be attributed to the unique characteristics of 3D point cloud DNNs. To enable practical use of adversarial examples in the real-world, it is necessary to design more transferable adversarial examples. Although hybrid training has demonstrated promising accuracy results, it comes with significantly higher training costs. Therefore, investigating novel techniques that can effectively reduce training costs is a potential research direction.

6. Conclusion and Future Direction

In this paper, we revisit the limitations of previous point cloud adversarial works and establish a comprehensive, rigorous, and unified benchmark for fair comparison of the adversarial robustness of point cloud DNNs. Moreover, we propose a hybrid training method that combines various adversarial examples, including adding, removing, and shifting, to enhance adversarial robustness. Through analysis of the benchmark results, we propose a more robust defense framework by integrating effective defense modules, achieving state-of-the-art adversarial robustness.

The remarkable defense accuracy achieved by ConvOnet demonstrates a direct relationship between the performance of the reconstruction network and the adversarial robustness. Thus, we recommend further investigation and implementation of advanced reconstruction networks to improve adversarial robustness. We highly encourage the community to contribute more advanced point cloud DNNs, attacks, and defenses to enrich future point cloud adversarial robustness benchmarks, benefitting real-world applications.

7. Acknowledgement

This work was partially supported by the National Science Foundation Grants CRII-2246067, CCF2211163, and the National Natural Science Foundation of China under Grant No. 62002126, NSFC22FYT45.

References

[1] HA Aziz and A Guled. Cloud computing and healthcare services. 2016.

[2] Y Cao, S Li, Y Liu, Z Yan, Y Dai, PS Yu, and Lichao Sun. A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt 2023. arxiv 2023. *arXiv preprint arXiv:2303.04226*.

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

[4] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, 2021.

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

[6] Shengshan Hu, Junwei Zhang, Wei Liu, Junhui Hou, Minghui Li, Leo Yu Zhang, Hai Jin, and Lichao Sun. Pointca: Evaluating the robustness of 3d point cloud completion models against adversarial examples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 872–880, 2023.

[7] 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*, pages 15335–15344, 2022.

[8] Binbin Liu, Jinlai Zhang, and Jihong Zhu. Boosting 3d adversarial attacks with attacking on frequency. *IEEE Access*, 10:50974–50984, 2022.

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

[10] Daniel Liu, Ronald Yu, and Hao Su. Adversarial shape perturbations on 3d point clouds. In *European Conference on Computer Vision*, pages 88–104. Springer, 2020.

[11] Wenxin Ma, Jian Chen, Qing Du, and Wei Jia. Pointdrop: Improving object detection from sparse point clouds via adversarial data augmentation. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 10004–10009. IEEE, 2021.

[12] AAM Muzahid, Wanggen Wan, Ferdous Sohel, Lianyao Wu, and Li Hou. Curvenet: Curvature-based multitask learning deep networks for 3d object recognition. *IEEE/CAA Journal of Automatica Sinica*, 8(6):1177–1187, 2020.

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

[14] François Pomerleau, Francis Colas, and Roland Siegwart. A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends in Robotics*, 4(1):1–104, 2015.

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

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

[17] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10901–10911, 2021.

[18] Jiawei Ren, Liang Pan, and Ziwei Liu. Benchmarking and analyzing point cloud classification under corruptions. *arXiv preprint arXiv:2202.03377*, 2022.

[19] Jiachen Sun, Qingzhao Zhang, Bhavya Kailkhura, Zhiding Yu, Chaowei Xiao, and Z Morley Mao. Benchmarking robustness of 3d point cloud recognition against common corruptions. *arXiv preprint arXiv:2201.12296*, 2022.

[20] Lichao Sun, Kazuma Hashimoto, Wenpeng Yin, Akari Asai, Jia Li, Philip Yu, and Caiming Xiong. Adv-bert: Bert is not robust on misspellings! generating nature adversarial samples on bert. *arXiv preprint arXiv:2003.04985*, 2020.

[21] Lichao Sun, Yingtong Dou, Carl Yang, Kai Zhang, Ji Wang, S Yu Philip, Lifang He, and Bo Li. Adversarial attack and defense on graph data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[22] Lichao Sun, Xiaobin Rui, and Wei Chen. Scalable adversarial attack algorithms on influence maximization. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 760–768, 2023.

[23] Yiming Sun, Feng Chen, Zhiyu Chen, and Mingjie Wang. Local aggressive adversarial attacks on 3d point cloud. In *Asian Conference on Machine Learning*, pages 65–80. PMLR, 2021.

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

[25] Saeid Asgari Taghanaki, Jieliang Luo, Ran Zhang, Ye Wang, Pradeep Kumar Jayaraman, and Krishna Murthy Jatavallabhula. Robustpointset: A dataset for benchmarking robustness of point cloud classifiers. *arXiv preprint arXiv:2011.11572*, 2020.

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

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

[28] 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*, pages 1588–1597, 2019.

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

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

[31] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019.

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

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

[34] Mutian Xu, Junhao Zhang, Zhipeng Zhou, Mingye Xu, Xiaojuan Qi, and Yu Qiao. Learning geometry-disentangled representation for complementary understanding of 3d object point cloud. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3056–3064, 2021.

[35] 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*, pages 2790–2799, 2018.

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

[37] Jinlai Zhang, Lyujie Chen, Bo Ouyang, Binbin Liu, Jihong Zhu, Yujing Chen, Yanmei Meng, and Danfeng Wu. Pointcutmix: Regularization strategy for point cloud classification. *arXiv preprint arXiv:2101.01461*, 2021.

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

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