This CVPR Findings paper is the Open Access version, provided by the Computer Vision Foundation. Except for this watermark, it is identical to the accepted version; the final published version of the proceedings is available on IEEE Xplore.

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

Geunyoung Jung[1] Soohong Kim[1] Inseok Kong[2] Jiyoung Jung[1*]
[1]Department of Artificial Intelligence, University of Seoul, South Korea
[2]Department of Geo Informatics, University of Seoul, South Korea
{gyjung975, shk00315, kong7963, jyjung}@uos.ac.kr

Abstract

The advent of deep neural networks has led to remarkable progress in 3D point cloud recognition, but they remain vulnerable to adversarial attacks. Although various defense methods have been studied, they suffer from a trade-off between robustness and transferability. We propose Adversarial Point Counterattack (APC) to achieve both simultaneously. APC is a lightweight input-level purification module that generates instance-specific counter-perturbations for each point, effectively neutralizing attacks. Leveraging clean–adversarial pairs, APC enforces geometric consistency in data space and semantic consistency in feature space. To improve generalizability across diverse attacks, we adopt a hybrid training strategy using adversarial point clouds from multiple attack types. Since APC operates purely on input point clouds, it directly transfers to unseen models and defends against attacks targeting them without retraining. At inference, a single APC forward pass provides purified point clouds with negligible time and parameter overhead. Extensive experiments on two 3D recognition benchmarks demonstrate that the APC achieves state-of-the-art defense performance. Furthermore, cross-model evaluations validate its superior transferability. The code is available at https://github.com/gyjung975/APC.

1. Introduction

3D point cloud recognition is a fundamental task in computer vision, gaining increasing attention as 3D sensors like LiDAR provide reliable geometric information about complex real-world environments. Recent advances in deep neural networks (DNNs) have significantly improved the performance of point cloud recognition. However, DNNs remain highly vulnerable to adversarial attacks, in which subtle perturbations can deceive the networks into making incorrect predictions while undetectable to human observers. Since point cloud data have many safety-critical applications, ensuring the robustness of DNNs against adversarial attacks is crucial for the reliable and secure deployment of 3D vision applications.

Existing adversarial attacks on point clouds generally fall into three strategies: adding, dropping, and shifting points. Early approaches [12, 37] generate adversarial examples by inserting adversarial points or clusters into the original input. The dropping strategy [40] constructs a point saliency map to determine the importance of each point and removes the points identified as important. Point shifting is another widely used strategy, where the coordinates of points are perturbed to induce misclassification. For instance, C&W-based [37] and PGD-based [14] attacks iteratively refine perturbations to produce stronger adversarial examples. However, many of these attacks suffer from limited imperceptibility, making them easier to detect. To alleviate this issue, KNN [29] incorporates distance constraints between adversarial and neighboring points, thereby preventing removal by defense methods. More recently, shape- and geometry-aware attacks [9, 19, 33] are introduced to further enhance imperceptibility by preserving local structure and curvature. Notably, HiT [19] hides perturbations within geometrically complex areas, guided by saliency and imperceptibility scores.

Since the robustness of DNNs against adversarial attacks is critical in real-world applications, a variety of defense methods are proposed. Existing approaches can be categorized into two types: input-level [5, 10, 16, 36, 41] and model-level [11, 14, 20, 30] defenses. Input-level defenses directly manipulate adversarial examples in data space. The most straightforward methods are Simple Random Sampling (SRS) and Statistical Outlier Removal (SOR) [41], which remove random points and statistical outlier points, respectively. In a more sophisticated approach, IF-Defense [36] optimizes point coordinates at inference time to restore the underlying surface. Inspired by randomized smoothing [3, 17] in the 2D domain, CausalPC [10] ensembles outputs from multiple augmented

*Corresponding author.

adversarial examples. Model-level defenses aim to improve the robustness of the model itself, rather than manipulating adversarial inputs. Adversarial training is the most widely adopted paradigm [7, 14, 20, 30], where adversarial examples are incorporated into training to enhance inherent robustness. Recently, hybrid training [11] leverages adversarial examples from multiple attacks, achieving strong defense performance.

In general, the two types of defenses exhibit complementary strengths and weaknesses. By operating purely in the data space, input-level defenses are model-agnostic and offer high transferability across models. However, their robustness remains relatively weak, as it is only indirectly induced by input manipulation rather than explicit defensive objectives. In contrast, model-level defenses achieve strong robustness but lack transferability and are less practical, as each model must be retrained from scratch with a specific robust objective.

To address the drawbacks of both types of defense simultaneously, we propose Adversarial Point Counterattack (APC), a lightweight input-level purification module. APC takes adversarial examples as input and generates instance-specific counter-perturbations that neutralize the effects of attacks at each point. By adding these counter-perturbations point-wise to adversarial examples, we obtain purified examples. APC is trained on clean-adversarial pairs with the objective of restoring adversarial examples to their clean counterparts. To further improve restoration capability, two consistency losses are employed: one enforcing coordinate consistency in the data space and the other ensuring feature consistency in the feature space. Additionally, following [11], we adopt a hybrid training strategy to enable universal defense. Once trained with adversarial examples from multiple attack types, APC achieves robustness against both seen and unseen adversarial attacks. While APC outperforms all existing model-level defenses, it also exhibits strong transferability due to its input-level design. Moreover, APC is lightweight and efficient, introducing negligible additional cost at inference time.

Extensive experiments on 3D recognition across diverse datasets and models demonstrate the effectiveness of APC. APC achieves state-of-the-art defense performance, surpassing both input- and model-level defenses. To evaluate its transferability, we conduct cross-model generalization experiments, where APC significantly outperforms existing input-level defenses, highlighting its strong transferability.

The main contributions can be summarized as follows:

- We propose APC, a lightweight input-level defense module that generates counter-perturbations to counteract adversarial attacks. Trained on clean–adversarial pairs with two consistency losses, APC effectively purifies adversarial examples.
- Through hybrid training and its input-level design, a single APC generalizes to unseen attacks and models, providing consistent robustness in both cross-attack and cross-model settings.
- APC achieves state-of-the-art defense performance on 3D recognition, outperforming both input-level and model-level defenses. Cross-model evaluations further demonstrate its strong transferability.

2. Related Works

2.1. Deep Learning on 3D Point Cloud Recognition

3D point clouds are widely used in various real-world applications such as autonomous driving and robotics, but their irregular and unordered nature poses significant challenges for effective processing. PointNet [23] pioneers the direct processing of raw point clouds. PointNet++ [24] further captures local geometric structures at multi-scale through a hierarchical architecture. Recent works introduce 3D convolutions [1, 13, 18, 28, 34] and graph-based models [25, 26, 32, 38]. DGCNN [32] is a representative graph-based model that leverages dynamic graph convolutions, constructing local graphs adaptively in both coordinate and feature spaces. In this paper, we focus on PointNet, PointNet++, and DGCNN, as they are widely adopted across diverse 3D tasks.

2.2. 3D Adversarial Attacks and Defenses

Attacks. Following the works in the 2D domain [2, 27], adversarial attacks are extended to 3D point clouds through three main strategies: adding, dropping, and shifting points. Add-based attacks [12, 37] generate adversarial examples by inserting new points or clusters into the original point clouds. Conversely, Saliency [40] removes points with high saliency, using a saliency map that assigns importance scores to each point. The most widely explored approach is shifting, where the coordinates of existing points are minimally perturbed. C&W-based [37] and PGD-based [14] attacks iteratively optimize per-point perturbations. Since imperceptibility is a crucial aspect for 3D attacks, recent works explicitly consider it. To reduce outlier-like perturbations, KNN [29] incorporates a KNN distance constraint to encourage local compactness. More advanced shape- and geometry-aware approaches [9, 15, 19, 33] improve imperceptibility by preserving the surface structure and local curvature. Additionally, HiT [19] conceals perturbations within less perceptually sensitive regions, based on saliency and imperceptibility scores. AdvPC [8] employs an auto-encoder to generate more transferable adversarial examples, thereby improving the attack transferability.

Defenses. Existing defense methods can be categorized into two main streams: input-level and model-level. Input-level methods operate directly in the data space by trans-

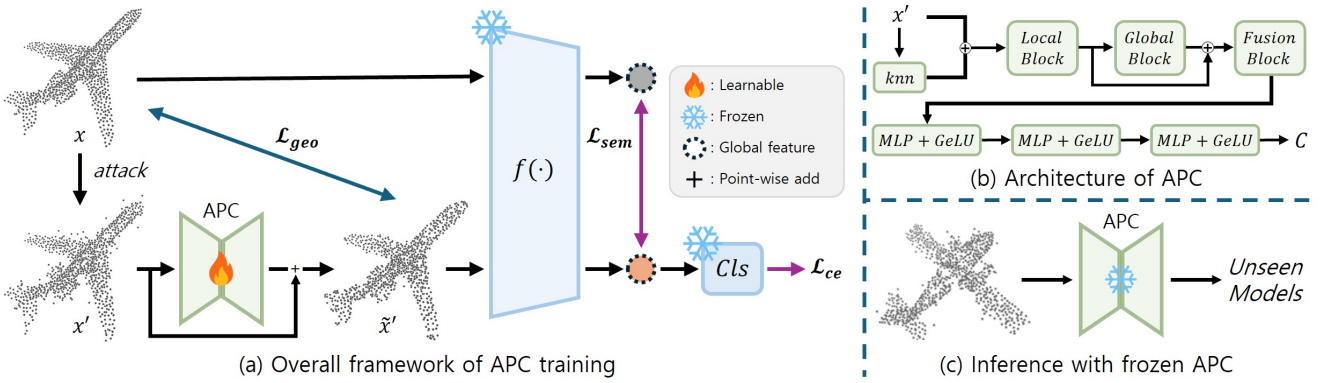


Figure 1. Overall pipeline of APC training and inference. Given an adversarial point cloud x' created by attacking a clean point cloud x, APC generates counter-perturbations C. By point-wise adding C to x', we obtain the purified point cloud $\tilde{x}'$. $f(\cdot)$ and Cls denote feature extractor and classifier of the victim model, respectively. Once trained, APC can be transferred to any model, as depicted in (c).

forming input point clouds. The simplest methods include Simple Random Sampling (SRS) and Statistical Outlier Removal (SOR) [41], which remove random or outlier points. DUP-Net [41] upsamples adversarial examples to restore missing details and refine surfaces using the reconstruction model PU-Net [39]. For more effective defense, IF-Defense [36] performs instance-specific iterative defense by optimizing point coordinates during inference. It reconstructs the surface via an implicit function network, such as ConvONet [22]. Another line of defense utilizes randomized smoothing [10, 16], which averages outputs from multiple augmented adversarial examples. However, they require a large number of forward passes (100–10,000) and additional training of victim models.

Model-level methods enhance robustness by retraining each victim model from scratch. A common strategy is adversarial training [14, 30], which augments training data with adversarial examples. Recently, recognizing that training with a single attack type is insufficient, hybrid training [11] incorporates multiple attack types.

2.3. 3D Point Cloud Denoising

3D point cloud denoising aims to restore clean geometry from noisy inputs, typically achieved via coordinate-level reconstruction using local geometric priors or learned representations. IterativePFN [4] performs iterative filtering through stacked denoising modules with dynamic graph convolution. PD-LTS [21] leverages latent space to uncover noise components and extract clean latent code for denoising. U-CAN [42] adopts an unsupervised multi-step denoising strategy based on point-wise noise-to-noise matching.

Although input-level adversarial defenses resemble denoising in modifying point coordinates, denoising focuses on geometric fidelity, whereas adversarial defense requires preserving task-relevant semantics.

3. Method

We first introduce the architecture and purification process of our Adversarial Point Counterattack (APC), followed by the training objectives in Sec. 3.1. Then, Sec. 3.2 presents a hybrid training strategy that enables universal defense by jointly leveraging multiple attack types. Finally, Section 3.3 discusses the inference process of APC, highlighting its transferability and efficiency. The overall framework of APC is illustrated in Fig. 1.

3.1. Adversarial Point Counterattack

APC is an input-level defense method that takes an adversarial point cloud as input and generates per-point counter-perturbations to neutralize attacks. The purified point cloud is obtained by adding these counter-perturbations point-wise. Leveraging a dataset of clean–adversarial pairs, APC is trained to transform adversarial point clouds into their clean counterparts.

3.1.1. Distribution-aware Counter-Perturbation

APC uses an encoder-decoder architecture, where the encoder extracts point features and the decoder generates per-point counter-perturbations. The encoder consists of three blocks: local, global, and fusion. The local and global blocks employ an MLP followed by max pooling to capture fine-grained local geometry and broader global context, respectively. The fusion block, implemented as an MLP, integrates these features to enhance the robustness of point features against adversarial attacks.

To effectively counteract attacks at individual point level, we begin with k-nearest neighbor (KNN) aggregation, which leverages local geometric information to suppress localized noise. Given an adversarial point cloud $x' \in \mathbb{R}^{N\times 3}$ with N points, we select k-nearest points for each point. Each point and its neighbors are concatenated and passed

through local block g^{local} to generate local point features:

$$P = KNN(x'), \quad P \in \mathbb{R}^{N\times k\times 3} \tag{1}$$

$$L = g^{local}([\text{repeat}_k(x'); P]), \quad L \in \mathbb{R}^{N\times d}, \tag{2}$$

where $\text{repeat}_k(x')$ denotes repeating each point k times along the second dimension, and d is the feature dimension. In addition, we further utilize a global feature to encode overall shape information via the global block g^{global}:

$$G = g^{global}(L), \quad G \in \mathbb{R}^{1\times d} \tag{3}$$

This global feature provides a holistic representation of the object and remains relatively stable against local perturbations. Finally, we fuse the local point features and the global features through g^{fusion} to obtain final point features:

$$E = g^{fusion}([L; \text{repeat}_N(G)]), \quad E \in \mathbb{R}^{N\times d}, \tag{4}$$

where G is repeated N times along the first dimension to match the shape of L. This fusion contextualizes each point with both its local neighborhood and the overall shape, thereby improving robustness against adversarial attacks.

The fused point features are then decoded into counter-perturbations C by a decoder h, composed of a 3-layer MLP with GeLU activations. By adding these counter-perturbations to the adversarial point cloud in a point-wise manner, we obtain the purified point cloud $\tilde{x}'$:

$$C = h(E), \quad C \in \mathbb{R}^{N\times 3} \tag{5}$$

$$\tilde{x}' = x' + C \tag{6}$$

The resulting purified point cloud is subsequently fed into the models for downstream tasks.

3.1.2. Optimization

The optimal goal of input-level defense is to recover the original clean examples from their adversarial counterparts. Therefore, we construct a dataset of clean-adversarial pairs, which provides direct supervision for restoring clean examples. Using this paired dataset, APC is trained with losses that enforce both geometric and semantic consistency.

Data-Space Geometric Consistency Loss. To ensure that a purified example retains geometric structure of its clean counterpart, we introduce a geometric consistency loss. The purified example $\tilde{x}'$ is constrained to align with the clean example x at the coordinate level. We achieve this by minimizing the Chamfer distance [6], which measures how far two point clouds are from each other, between $\tilde{x}'$ and x. It is calculated by averaging the distances of all nearest point pairs. The geometric consistency loss is defined as:

$$\mathcal{L}_{geo} = \frac{1}{\|\tilde{x}'\|_0} \sum_{p'\in\tilde{x}'} \min_{p\in x} \|p' - p\|_2^2 \tag{7}$$

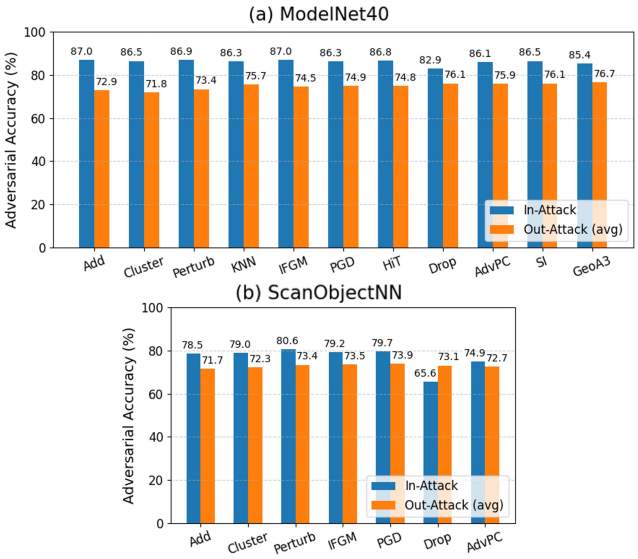


Figure 2. Adversarial accuracies of IAPC on in-attack and out-attack (avg) across two datasets. In-attacks are almost completely defended, but the performance on out-attacks is relatively low.

Feature-Space Semantic Consistency Loss. In addition to geometric consistency, restoring high-level semantics is also important for recognition. To this end, we further introduce a semantic consistency loss between the features of the purified and clean examples to ensure that the purified features retain essential semantics of the clean ones. Specifically, we use mean squared error (MSE) to encourage the global features of $\tilde{x}'$ and x to be similar, defined as:

$$\mathcal{L}_{sem} = \frac{1}{d}\sum_{i=1}^{d}(f(\tilde{x}')_i - f(x)_i)^2, \tag{8}$$

where $f(\cdot)$ denotes the feature extractor of the model, and $f(\tilde{x}')$ and $f(x)$ represent the global features of the purified example and its corresponding clean example, respectively.

By jointly optimizing losses in the data space and feature space, the purification process of APC not only rectifies local coordinate perturbations but also restores high-level semantic information.

Final Loss. The final loss combines the two consistency terms with a standard cross-entropy loss as follows:

$$\mathcal{L} = \mathcal{L}_{ce} + \alpha\cdot\mathcal{L}_{geo} + \beta\cdot\mathcal{L}_{sem}, \tag{9}$$

where α and β are hyperparameters that balance the contributions of each consistency loss.

3.2. Universal Counterattack

Training APC with only a single type of attack may limit its generalizability. Figure 2 illustrates the cross-attack performance of APC trained on a specific attack, denoted as

Table 1. Classification results (%) of adversarial examples of various attacks on ModelNet40 with three victim models. The experiments are conducted under a white-box setting and with untargeted attacks. **Bold** indicates the best performance. Overall, APC consistently achieves the highest robustness across different models and attacks.

Model	Defense	Add	Cluster	Perturb	KNN	IFGM	PGD	HiT	Drop	AdvPC	SI	GeoA3	Avg.
PointNet	No Defense	0.0	0.2	0.0	4.7	12.5	11.7	0.9	36.1	0.0	0.0	0.0	6.0
	SRS	81.5	74.7	79.7	29.7	80.5	79.7	17.7	34.6	24.4	54.3	55.5	55.7
	SOR	80.9	84.3	85.4	65.3	84.6	84.4	16.7	38.6	44.7	72.7	65.8	65.8
	DUP-Net	82.4	83.3	84.8	75.3	84.7	84.5	31.1	44.6	69.3	79.6	76.2	72.3
	IF-Defense	84.2	84.4	84.2	80.4	85.0	83.8	83.6	58.3	79.2	82.0	81.3	80.6
	CausalPC	84.4	84.3	84.2	80.6	84.0	83.9	38.3	44.9	76.6	82.2	78.9	74.8
	AT	85.1	80.7	86.0	68.4	86.0	85.8	51.6	59.1	76.2	77.1	81.5	76.1
	HT	84.3	84.2	84.7	79.6	84.8	83.7	62.5	75.5	78.9	81.8	81.1	80.1
	APC	**86.4**	**86.3**	**86.6**	**84.0**	**86.7**	**86.7**	**85.9**	**79.4**	**81.5**	**85.0**	**83.3**	**84.7**
PointNet++	No Defense	84.4	15.6	80.9	1.9	51.1	48.2	65.3	58.8	13.2	1.8	33.7	41.4
	SRS	86.9	70.3	85.5	24.7	78.9	78.8	62.2	54.4	44.8	56.9	74.1	65.2
	SOR	**88.0**	87.5	87.5	50.4	84.6	84.8	62.5	60.5	59.1	80.2	80.6	75.1
	DUP-Net	**88.0**	86.9	**87.6**	59.8	85.7	85.6	65.7	64.0	64.1	82.2	82.1	77.4
	IF-Defense	86.9	85.9	87.4	73.7	86.2	85.9	86.7	70.7	76.2	83.5	82.8	82.3
	AT	87.0	83.9	87.4	52.1	86.8	86.8	74.0	72.0	71.4	76.4	85.4	78.5
	HT	87.1	85.0	87.2	79.6	**87.8**	86.2	65.4	84.1	82.7	84.1	**87.7**	83.3
	APC	87.6	**87.5**	86.8	**83.3**	87.0	**86.7**	**86.5**	**75.9**	**82.6**	**85.9**	84.6	**84.9**
DGCNN	No Defense	6.7	5.7	4.3	1.9	32.3	30.6	19.7	58.1	0.9	0.9	0.0	14.6
	SRS	80.9	66.7	69.1	29.9	71.8	71.2	49.7	24.4	28.7	48.8	56.5	54.3
	SOR	87.2	86.0	87.8	31.7	87.2	87.1	47.2	62.6	41.2	74.9	72.9	69.6
	DUP-Net	52.3	52.9	51.8	34.7	51.8	51.2	19.5	37.0	30.3	42.2	40.2	42.2
	IF-Defense	87.1	85.5	87.0	70.3	86.9	85.6	84.1	68.8	75.0	82.4	82.7	81.4
	CausalPC	84.4	83.7	83.3	76.9	84.0	83.7	47.7	61.1	**78.2**	81.2	82.7	77.0
	AT	**87.3**	84.0	87.0	51.9	87.3	87.1	72.8	67.4	62.3	81.9	84.8	77.6
	HT	78.4	78.9	**87.9**	52.1	**87.9**	87.1	72.9	68.9	60.9	**83.0**	**85.4**	76.7
	APC	86.9	**86.8**	86.7	**77.3**	86.8	**87.1**	**87.4**	**75.4**	76.7	82.4	83.9	**83.4**

Independent-APC (IAPC). Here, in-attack and out-attack (avg) refer to the performance on the attack used for training and the average performance over the remaining attacks, respectively. Each IAPC exhibits strong robustness in the in-attack setting, achieving an average accuracy of 86.2% on ModelNet40 [35], which is close to the clean accuracy of the vanilla model (87.6%). However, its performance degrades significantly when evaluated on unseen attacks. This suggests that training with a single attack leads to overfitting and hinders generalization across diverse attack patterns. Note that although IAPC's out-attack performance lags behind its in-attack result, it still shows solid robustness on ModelNet40 (74.8%) and even achieves a state-of-the-art performance on ScanObjectNN (73.0%), as shown in Tabs. 1 and 2.

Following [11], we adopt a hybrid training strategy to alleviate this limitation. We train APC jointly on adversarial examples generated from multiple attack types. Specifically, we select one attack from each of the following categories: shifting, shape-aware shifting, and dropping. By exposing APC to heterogeneous attack patterns during training, it learns a more universal purification ability. As we demonstrate in Sec. 4, this approach substantially improves out-attack robustness while maintaining strong in-attack performance.

3.3. Inference with Frozen APC

In this section, we describe the inference process using the trained APC, highlighting its transferability and efficiency. As shown in part (c) of Fig. 1, APC can be transferred in a plug-and-play manner to various models, since it performs input-level purification. With a frozen APC, adversarial examples targeting models other than the one used for APC training can also be defended. For instance, adversarial examples crafted to deceive DGCNN [32] can be purified by an APC trained with PointNet [23].

We emphasize that, through its superior transferability and universal defense capability enabled by hybrid training, APC can be applied to a wide range of unseen models and remains effective against diverse attacks. Moreover, it is lightweight, containing only 0.2M parameters (7% of PointNet), and extremely efficient, as purification requires only a

Table 2. Classification results (%) of adversarial examples on ScanObjectNN OBJ_ONLY split.

Model	Defense	Add	Cluster	Perturb	IFGM	PGD	Drop	AdvPC	Avg.
PointNet	No Defense	0.3	0.3	0.0	10.0	9.6	45.9	0.0	9.4
	SRS	70.9	69.8	70.9	73.8	73.4	46.3	22.9	61.2
	SOR	77.7	79.0	78.6	78.7	78.6	47.5	46.1	69.4
	DUP-Net	77.8	78.6	78.7	79.1	78.3	51.1	57.6	71.6
	IF-Defense	44.2	43.0	43.0	43.2	43.2	38.7	39.9	42.2
	CausalPC	75.6	74.2	73.5	72.8	73.8	50.1	61.8	68.8
	AT	77.1	75.2	77.6	77.2	78.1	63.0	62.0	72.9
	HT	72.2	66.4	75.9	75.9	74.8	**64.7**	61.9	70.3
	APC	**80.0**	**80.0**	**80.6**	**80.4**	**80.4**	59.9	**71.3**	**76.1**
PointNet++	No Defense	77.7	17.3	73.1	31.8	28.4	63.0	14.1	43.7
	SRS	82.7	66.9	83.4	79.5	78.1	64.3	54.7	72.8
	SOR	85.5	85.2	85.7	84.1	83.1	65.4	64.7	79.1
	DUP-Net	85.8	85.3	**87.4**	82.9	84.1	68.5	67.1	80.2
	IF-Defense	48.0	47.8	47.3	46.8	47.2	44.4	40.4	46.0
	AT	84.5	76.4	83.6	81.5	82.1	71.9	69.5	78.5
	HT	84.0	65.5	82.9	80.2	79.7	73.8	64.8	75.8
	APC	**85.7**	**85.3**	85.2	**84.9**	**85.2**	**74.9**	**79.0**	**82.9**
DGCNN	No Defense	4.3	5.5	3.4	21.3	23.6	62.8	0.5	17.4
	SRS	77.1	67.8	72.9	76.6	75.2	44.7	45.1	65.6
	SOR	82.7	**86.5**	85.3	84.5	86.0	65.4	51.8	77.5
	DUP-Net	76.0	73.8	75.2	77.4	75.7	64.9	58.0	71.6
	IF-Defense	52.6	53.2	54.2	54.0	52.1	46.3	41.8	50.6
	CausalPC	80.7	81.1	82.8	81.9	81.1	68.3	**76.6**	78.9
	AT	84.5	81.9	83.5	85.7	85.9	71.1	62.3	79.3
	HT	80.7	76.0	81.4	84.8	85.3	**79.3**	68.5	79.4
	APC	**86.2**	85.5	**86.7**	**87.4**	**87.1**	76.2	63.9	**81.9**

single APC forward pass.

4. Experiments

4.1. Experimental Settings

Datasets and Models. We conduct experiments on ModelNet40 (MN40) [35] and ScanObjectNN (SONN) [31] datasets. MN40 is a synthetic dataset consisting of complete and noise-free point clouds. It contains 12,311 samples across 40 categories, with 9,843 and 2,468 samples for the train and test sets, respectively. SONN is a real-scanned dataset with 15 categories, characterized by substantial missing points and noise. The dataset is divided into three splits according to difficulty level (OBJ_ONLY, OBJ_BG, and PB), and we report results on the OBJ_ONLY split. Additional results on the OBJ_BG and PB splits are provided in the supplementary materials. For both datasets, we uniformly sample 1,024 points from each point cloud. For victim models, we adopt three commonly used point cloud classification models: PointNet [23], PointNet++ [24], and DGCNN [32].

Baseline Attacks. We use eleven representative adversarial attacks: two adding attacks (Add [37], Cluster [37]), one dropping attack (Drop [40]), and seven shifting attacks (Perturb [37], KNN [29], IFGM, PGD [14], HiT [19], AdvPC [8], SI [9], and GeoA3 [33]). All attacks are implemented with untargeted, white-box setting. From the attacker's perspective, untargeted and white-box attacks are generally stronger than targeted and black-box attacks. Therefore, the untargeted under white-box setting represents the upper bound of attack strength and the most challenging scenario to defend.

Baseline Defenses. For defense baselines, we consider seven methods: five input-level defenses (SRS, SOR [41], DUP-Net [41], IF-Defense [36], and CausalPC [10]), and two model-level defenses (adversarial training (AT) [30] and hybrid training (HT) [11]). Since CausalPC has no official implementation, we re-implement it with 1,024 points following the paper. AT is trained with a mixture of PGD adversarial examples and clean examples. Unlike AT, HT leverages multiple attack types—KNN, HiT, and Drop—representing shifting, shape-aware shifting, and dropping attacks, respectively.

Implementation Details. For APC training, we adopt the combination of KNN, HiT, and Drop attacks together with clean examples, same as HT. To improve data efficiency, we only use 30% of the training examples for each attack, as well as clean examples. The victim models are kept frozen while only the APC is updated, and clean-adversarial pairs are constructed from the training split of each dataset. Following existing input-level defenses, we also pre-process input data with SOR before feeding it into APC.

4.2. 3D Point Cloud Recognition

Table 1 summarizes the adversarial classification accuracies on MN40 under eleven different attacks across three victim models. APC consistently performs either best or second-best across nearly all attacks and models. On PointNet, it outperforms all baselines, including both input-level and model-level defenses, achieving a state-of-the art (SOTA) average accuracy of 84.7%. Also on PointNet++ and DGCNN, it achieves average accuracies of 84.9% and 83.4%, respectively, again surpassing existing defenses by a clear margin.

Unlike MN40, SONN does not provide the normal vector of each point. Therefore, we only consider attacks that can be implemented with coordinates alone. For HT and APC training, the combination of attacks becomes Perturb, PGD, and Drop. Similar to the results on MN40, APC achieves the best robustness, as shown in Tab. 2. In particular, APC yields average adversarial accuracies of 75.8%, 82.8%, and 81.9% on PointNet, PointNet++, and DGCNN, respectively, all representing SOTA performance. The remarkable robustness of APC on SONN confirms its practical usability, demonstrating that it remains effective even when evaluated on real-world data.

Although APC is an input-level defense, which typically offers transferability at the cost of robustness, it still outperforms model-level defenses. The transferability of APC is further evaluated in the following section.

Table 3. Adversarial accuracies (%) on ModelNet40 in the cross-model transfer setting. APC is trained with the source model and evaluated with target models without retraining. The results are averages of performance across all attacks. APC exhibits strong transferability, consistently outperforming baseline defenses across diverse source–target pairs.

Defense	Source	Target		Source	Target		Source	Target	
	PointNet	PointNet++	DGCNN	PointNet++	PointNet	DGCNN	DGCNN	PointNet	PointNet++
SRS	55.7	65.2	54.3	65.2	55.7	54.3	54.3	55.7	65.2
SOR	65.8	75.1	69.6	75.1	65.8	69.6	69.6	65.8	75.1
DUP-Net	72.3	77.4	42.2	77.4	72.3	42.2	42.2	72.3	77.4
IF-Defense	80.6	82.3	81.4	82.3	**80.6**	81.4	81.4	**80.6**	82.3
APC	**84.7**	**83.1**	**81.5**	**84.9**	77.2	**81.6**	**83.4**	77.2	**83.0**

Table 4. Adversarial accuracies (%) on ScanObjectNN OBJ_ONLY split in the cross-model transfer setting.

Defense	Source	Target		Source	Target		Source	Target	
	PointNet	PointNet++	DGCNN	PointNet++	PointNet	DGCNN	DGCNN	PointNet	PointNet++
SRS	61.2	72.8	65.6	72.8	61.2	65.6	65.6	61.2	72.8
SOR	69.4	79.1	77.5	79.1	69.4	77.5	77.5	69.4	79.1
DUP-Net	71.6	80.2	71.6	80.2	71.6	71.6	71.6	71.6	80.2
IF-Defense	42.2	46.0	50.6	46.0	42.2	50.6	50.6	42.2	46.0
APC	**76.1**	**81.6**	**80.2**	**82.9**	**72.1**	**79.0**	**81.9**	**74.1**	**81.3**

4.3. Cross-model Transferability

To validate transferability of APC, we conduct cross-model generalization experiment. As an input-level method, APC performs purification directly on input point clouds and can, in principle, be applied to any model as a plug-and-play module once trained. Tables 3 and 4 present the results of cross-model tranfer on MN40 and SONN. We report the average adversarial accuracies across all attacks. APC is trained with source model using adversarial examples targeting that source model. Then, it is transferred to target models and evaluated on adversarial examples targeting corresponding target model, *i.e.*, both models and adversarial examples are unseen during APC training. We compare APC with input-level defenses which have transferability, except CausalPC. While CausalPC also manipulates the input point clouds, it requires dataset-specific fine-tuning to adapt to its modified architecture. Here, the baselines defenses are model-agnostic, meaning that their performances on target models are independent of the source model.

On both datasets, APC effectively purifies adversarial examples from various attacks and consistently yields high robustness across diverse source-target pairs, demonstrating strong cross-model generalization. Compared to the model-level defenses in Tabs. 1 and 2, which lack transferability, APC achieves comparable or even superior performance. Based on standard recognition and cross-model transfer results, we highlight that APC exhibits notable transferability while maintaining state-of-the-art defense performance.

Table 5. Inference time and the number of parameters for each method.

Metric	DUP	IF	APC
Time (s)	0.3	3.2	0.002
#Params (M)	0.8	2.0	0.2

Table 6. Performance by the number of attacks used in hybrid training

#Attacks	MN40		SONN	
	Cln.	Adv.	Cln.	Adv.
1	86.6	75.5	80.0	73.5
2	**87.1**	82.2	79.8	74.5
3	86.6	**84.7**	**80.6**	**76.1**

Efficiency. In addition to strong transferability across models, APC is also highly efficient. We measure the inference time required to purify a single adversarial example as well as the number of parameters of the model used. For comparison, we consider network-based methods like APC, *i.e.*, DUP-Net and IF-Defense. As shown in Tab. 5, APC shows the lowest inference time (0.002s) and requires the fewest parameters (0.2M), corresponding to only about 0.06% and 10% of those of IF-Defense.

4.4. Ablation Studies

Effect of Attack Diversity in Hybrid Training. Table 6 presents performance under varying numbers of attack types used for hybrid training with PointNet. To isolate the effect of the number of attacks, we keep total amount of training data fixed. The first and second rows show averages over all combinations, *i.e.*, three results. The adversarial accuracy gradually improves as the number of attacks increases, reaching 84.7% and 76.1% on MN40 and SONN, respectively. Importantly, clean accuracy is preserved, indicating that APC enhances robustness without compromising normal performance. Note that even with only two

Table 7. Ablations on loss terms and usage of clean data.

$\mathcal{L}_{geo}$	$\mathcal{L}_{sem}$	Clean	ModelNet40		ScanObjectNN	
			Cln.	Adv.	Cln.	Adv.
		✓	85.4	83.6	79.3	74.5
✓		✓	85.7	84.0	79.3	75.0
	✓	✓	86.5	84.6	80.0	75.5
✓	✓		85.6	84.5	79.0	75.6
✓	✓	✓	**86.6**	**84.7**	**80.6**	**76.1**

Table 8. Comparison of two distance metrics.

Distance	ModelNet40			ScanObjectNN		
	PointNet	PointNet++	DGCNN	PointNet	PointNet++	DGCNN
Hausdorff Dist.	84.5	85.1	83.4	75.3	82.8	81.8
Chamfer Dist.	84.7	84.9	83.4	76.1	82.9	81.9

attack types, APC already achieves state-of-the-art performance on both datasets.

Loss Analysis. Table 7 shows ablation studies on the effectiveness of the geometric consistency loss $\mathcal{L}_{geo}$ and the semantic consistency loss $\mathcal{L}_{sem}$. On both datasets, using either loss individually improves performance compared to baseline. The semantic consistency loss contributes more to performance improvement than geometric consistency loss, and combining both yields the best overall performance.

We compare two distance metrics used in the geometric consistency loss, as shown in Tab. 8. The Hausdorff distance is a similar metric to the Chamfer distance, but instead of averaging the distances of all nearest point pairs, it takes their maximum. Both metrics perform almost equally well, and we adopt the Chamfer distance as the default.

4.5. Clean–Adversarial Accuracy Trade-off

Balancing clean accuracy and adversarial robustness is crucial for defense, as the two are generally in a trade-off relationship. Table 9 reports the clean accuracy of defense methods across datasets and models. Our APC maintains competitive clean accuracy, while achieving state-of-the-art adversarial robustness, and even surpasses vanilla performance on the ScanObjectNN.

In addition, the "Clean" column in Tab. 7 compares performance with and without clean examples during training. Incorporating clean examples improves not only clean accuracy but also adversarial robustness. We assume that, as clean examples serve as supervision in the training, their inclusion provides information about data distribution that purified examples should have.

Visualization. Figure 3 visualizes purification results of APC. First column shows clean examples correctly classified as "Guitar" and "Airplane". Adversarial examples are in the middle column, where same objects are misclassified as "Bench" and "Guitar", respectively. Third column presents the purified examples after applying APC, successfully recovering the correct labels. Red circles highlight the regions where perturbations are mitigated by restoring displaced points onto object surface and filling empty areas.

Table 9. Classification results (%) of clean examples.

Defense	ModelNet40			ScanObjectNN		
	PointNet	PointNet++	DGCNN	PointNet	PointNet++	DGCNN
Vanilla	87.6	89.1	90.1	80.4	86.5	87.4
SRS	86.1	87.4	79.3	78.3	85.5	76.4
SOR	86.3	88.1	87.5	79.7	85.3	86.7
DUP-Net	85.5	88.0	54.2	79.1	86.2	77.2
IF-Defense	84.8	86.8	86.9	43.3	46.6	53.2
CausalPC	84.6	-	84.2	75.7	-	82.8
AT	85.9	88.5	**88.8**	79.3	85.5	87.3
HT	**86.6**	**88.9**	88.2	79.1	85.8	85.7
APC	**86.6**	87.8	87.4	**80.6**	**86.4**	**87.8**

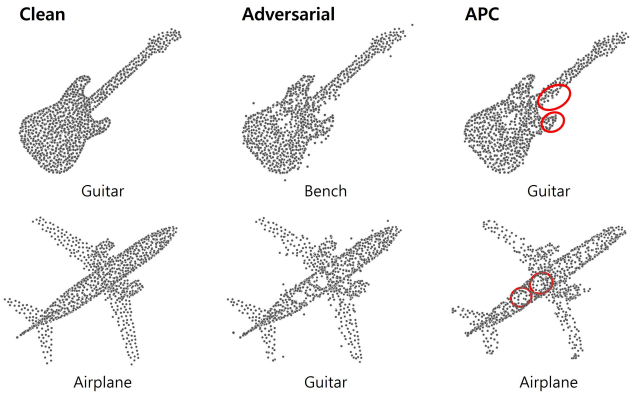


Figure 3. Visualization of clean, adversarial, and purified examples. The labels below each point cloud are the prediction outputs of the victim model.

5. Conclusion

In this paper, we propose Adversarial Point Counterattack (APC), a lightweight input-level purification module that generates per-point counter-perturbations to neutralize adversarial attacks. APC is trained on clean–adversarial pairs with geometric and semantic consistency losses, enabling it to simultaneously recover local geometry and global semantics. Through hybrid training over multiple attack types and its input-level design, APC generalizes robustly to unseen attacks and transfers effectively to unseen models. Extensive experiments on synthetic and real-scanned 3D recognition benchmarks demonstrate that APC achiev staeste-of-the-art adversarial accuracies. Moreover, APC provides this robustness with negligible overhead—requiring only a 0.002s forward pass and 0.2M parameters—making it practical for various real-time 3D applications.

Acknowledgments. This research was supported by the National Research Foundation of Korea(NRF) grant funded by the korea government(MSIT) for Geunyoung Jung (RS-2022-NR068754) and for Jiyoung Jung (RS-2025-24523036).

References

[1] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point convolutional neural networks by extension operators. *ACM Trans. Graph.*, 37(4), 2018. 2

[2] 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. 2

[3] 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. 1

[4] Dasith de Silva Edirimuni, Xuequan Lu, Zhiwen Shao, Gang Li, Antonio Robles-Kelly, and Ying He. Iterativepfn: True iterative point cloud filtering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13530–13539, 2023. 3

[5] Xiaoyi Dong, Dongdong Chen, Hang Zhou, Gang Hua, Weiming Zhang, and Nenghai Yu. Self-robust 3d point recognition via gather-vector guidance. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[6] 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. 4

[7] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015. 2

[8] 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. 2, 6

[9] 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, 2, 6

[10] 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. 1, 3, 6

[11] Qiufan Ji, Lin Wang, Cong Shi, Shengshan Hu, Yingying Chen, and Lichao Sun. Benchmarking and analyzing robust point cloud recognition: Bag of tricks for defending adversarial examples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4295–4304, 2023. 1, 2, 3, 5, 6

[12] Jaeyeon Kim, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Minimal adversarial examples for deep learning on 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7797–7806, 2021. 1, 2

[13] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Advances in Neural Information Processing Systems*, 2018. 2

[14] 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, 2, 3, 6

[15] Daniel Liu, Ronald Yu, and Hao Su. Adversarial shape perturbations on 3d point clouds. In *European Conference on Computer Vision (ECCV) Workshops*, pages 88–104, 2020. 2

[16] Hongbin Liu, Jinyuan Jia, and Neil Zhenqiang Gong. Pointguard: Provably robust 3d point cloud classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6186–6195, 2021. 1, 3

[17] 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. 1

[18] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[19] 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, 2, 6

[20] 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, 2021. 1, 2

[21] 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

[22] 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. 3

[23] 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. 2, 5, 6

[24] 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. 2, 6

[25] Yiru Shen, Chen Feng, Yaoqing Yang, and Dong Tian. Mining point cloud local structures by kernel correlation and graph pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[26] Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[27] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *Proceedings of the International Conference on Learning Representations(ICLR)*, 2014. 2

[28] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, Francois Goulette, and Leonidas J. Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2

[29] 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, 2, 6

[30] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 3, 6

[31] 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. 6

[32] 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. 2, 5, 6

[33] 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, 2, 6

[34] 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 (CVPR)*, 2019. 2

[35] 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. 5, 6

[36] 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. 1, 3, 6

[37] 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, 2, 6

[38] Qiangeng Xu, Xudong Sun, Cho-Ying Wu, Panqu Wang, and Ulrich Neumann. Grid-gcn for fast and scalable point cloud learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[39] 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. 3

[40] 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, 2, 6

[41] 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, 3, 6

[42] Junsheng Zhou, XingYu Shi, Haichuan Song, Yi Fang, Yu-Shen Liu, and Zhizhong Han. U-CAN: Unsupervised point cloud denoising with consistency-aware noise2noise matching. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 3