

PointCVar: Risk-Optimized Outlier Removal for Robust 3D Point Cloud Classification

Xinke Li¹, Junchi Lu², Henghui Ding², Changsheng Sun¹, Joey Tianyi Zhou^{3,4}, Yeow Meng Chee¹

¹National University of Singapore,

²Nanyang Technological University,

³Institute of High Performance Computing (IHPC), A*STAR,

⁴Centre for Frontier AI Research (CFAR), A*STAR

{xinke.li, cssun}@u.nus.edu, luju0008@e.ntu.edu.sg, henghui.ding@gmail.com, zhouty@cfar.a-star.edu.sg, ymchee@nus.edu.sg

Abstract

With the growth of 3D sensing technology, the deep learning system for 3D point clouds has become increasingly important, especially in applications such as autonomous vehicles where safety is a primary concern. However, there are growing concerns about the reliability of these systems when they encounter noisy point clouds, either occurring naturally or introduced with malicious intent. This paper highlights the challenges of point cloud classification posed by various forms of noise, from simple background noise to malicious adversarial/backdoor attacks that can intentionally skew model predictions. While there’s an urgent need for optimized point cloud denoising, current point outlier removal approaches, an essential step for denoising, rely heavily on handcrafted strategies and are not adapted for higher-level tasks, such as classification. To address this issue, we introduce an innovative point outlier removal method that harnesses the power of downstream classification models. Using gradient-based attribution analysis, we define a novel concept: *point risk*. Drawing inspiration from tail risk minimization in finance, we recast the outlier removal process as an optimization problem, named *PointCVar*. Extensive experiments show that our proposed technique not only robustly filters diverse point cloud outliers but also consistently and significantly enhances existing robust methods for point cloud classification. A notable feature of our approach is its effectiveness in defending against the latest threat of backdoor attacks in point clouds.

Introduction

The accessibility of 3D sensing technology has led to the popularity of deep learning on 3D point clouds in various industrial applications, including safety-critical ones such as autonomous driving (Chen et al. 2017). However, concerns over the safety of 3D deep learning have been substantiated by recent studies that have demonstrated the negative impact of natural or artificial noisy points on point cloud deep model performance (Xiang, Qi, and Li 2019; Sun et al. 2022). Consequently, the reliability of 3D deep visual systems in industry is now overshadowed.

Indeed, it is proposed that even some common types of point cloud noise could degrade the performance of deep

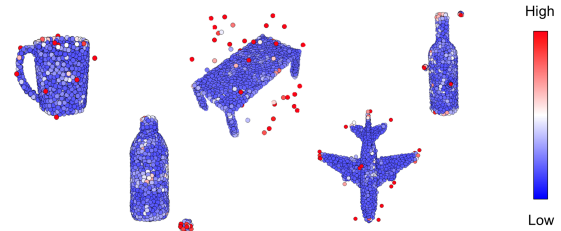


Figure 1: Visualizing point risks for point clouds with various types of outliers. It shows that the noise points often pose high risks compared to the clean points.

models obviously. For instance, background points resulting from inaccurate sensors or incomplete processing can severely disrupt the predictions of point cloud classifiers, as revealed by (Ren, Pan, and Liu 2022). Moreover, some artificial modifications to the points can be disguised as noise and intentionally manipulate the output of classifiers. On one hand, delicately designed adversarial noise, such as outliers or slight perturbations, has been shown to significantly decrease the classification accuracy in various studies (Xiang, Qi, and Li 2019; Wicker and Kwiatkowska 2019). On the other hand, a new form of attack, called a backdoor attack, has emerged as a severe threat. This attack involves implanting malicious functionality into a deep model by modifying the training data. During the inference of this model, if the input point cloud contains specific outliers named backdoor triggers, such as a small sphere, the model will output an incorrect target label (Li et al. 2021). Thus, developing robust point cloud denoising pipelines becomes imperative to ensure the safety and reliability of 3D deep learning.

Despite the pressing need for point cloud denoising towards reliable deep learning, outlier removal, an essential processing step in a point cloud denoising pipeline, has not been specifically designed or optimized for deep learning yet. In current practice, the objective of outlier removal is eliminating points in the point cloud that do not conform to the geometrical characteristics of the local points. It is utilized to remove noise points directly or to pave the way for further filtering. Traditional methods, such as statistical outlier removal (Rusu 2010), have been demonstrated

to be useful in preventing adversarial noise (Zhou et al. 2019). However, these methods heavily rely on handcrafted procedures and suffer from inferior performance for certain outliers, *e.g.*, backdoor triggers (Li et al. 2021). There have also been some learning-based attempts, such as PointCleanNet (Rakotosaona et al. 2020), which operates on a local patch of points. Although these methods are effective against specific types of noise, they incur non-trivial computational costs and greatly depend on the construction of noise datasets with ground truth. Besides their inherent limitations, the above outlier removal methods have not yet considered their impact on high-level tasks including classification, and thus, their use in the context of point-cloud classification may result in suboptimal performance.

To meet such needs, we propose an effective point outlier removal method in this work by leveraging the downstream deep models. Initially, we employ gradient-based attribution analysis to determine the point-wise sensitivity to the model output, referred to as *point risk*. Further observation reveals that noise points mainly contribute to the tail risk in the point risk distribution, *i.e.*, they are sparse but high in risk as showcased in Figure 1. Motivated by tail risk minimization in financial applications (Rockafellar, Uryasev et al. 2000), **we propose to formulate the outlier removal process into an optimization problem by introducing a convex risk measure**. Specifically, the process uses a tail risk measure as the objective and utilizes linear programming to obtain binary weights for input points, which are used to process the point cloud subsequently. The algorithm can be further approximated to linear computational complexity for higher efficiency. Extensive experiments demonstrate that the proposed method can effectively filter various types of point cloud outliers including random, adversarial, and backdoor trigger points, and its improvement in point cloud classification accuracy can exceed existing outlier removal methods. Moreover, our method can be combined with existing methods as a plugin to enhance the model’s robustness to many adversarial noises. In general, we claim to have made the following three contributions¹.

- We analyze the noise within point clouds via the lens of deep model risk analysis, revealing that the noise points lead to elevated tail risk in the point risk distribution.
- We derive an explainable outlier removal method for point clouds from the tail risk optimization, which only uses a trained classification model and thus avoids extra training or model modifications. It is also compatible with existing methods for robust point cloud classification.
- Extensive experiments of point cloud classification show that the designed method can effectively remove various types of point outliers. Especially, this method is, as far as we know, the first to successfully defend against point cloud backdoor attacks. Furthermore, it can consistently enhance existing robust techniques for 3D deep models when utilized as a plugin.

¹Appendix and other resources are available on the public page: <https://github.com/shinke-li/pointcvar>

Related Work

Noise Effect on Point Cloud Classification

Deep learning models have shown impressive results in 3D point cloud classification (Qi et al. 2017; Xu et al. 2021). Yet, they struggle with noisy data. Real-world point clouds often contain noise types that affect classifiers’ performance (Uy et al. 2019; Ren, Pan, and Liu 2022; Sun et al. 2022). There are also adversarial noise attacks, including point perturbation (Xiang, Qi, and Li 2019; Liu, Yu, and Su 2019; Tsai et al. 2020; Hamdi et al. 2020), adversarial point addition (Xiang, Qi, and Li 2019; Wicker and Kwiatkowska 2019; Yang et al. 2019), and varied real-world noise simulations (Zheng et al. 2019; Zhao et al. 2020). New backdoor attacks have been introduced that train models to recognize specific noise or triggers (Li et al. 2021; Xiang et al. 2021). Outlier removal, a preprocessing technique, can reduce noise impact on classification (Zhou et al. 2019). It aims to remove points disrupting point cloud uniformity using techniques like SOR (Rusu 2010) and PointCleanNet (Rakotosaona et al. 2020). We aim to improve outlier removal efficacy by leveraging classification information.

Robust Point Cloud Classification

Three types of methods have been proposed to improve robustness of deep point cloud classifiers. **Model-based** methods focus on enhancing the model via designs or training strategies, such as the gather module by Dong *et al.* and sorting-based pooling operations by Sun *et al.* (Dong et al. 2020; Sun et al. 2020). Robust training is also achieved through point cloud data augmentation methods such as PointMixup, PointCutMix, and adversarial training (Chen et al. 2020; Zhang et al. 2022; Sun et al. 2021). **Certified** methods, like the one by Liu *et al.*, theoretically increase robustness to noise through certified classification, despite high computational costs (Liu, Jia, and Gong 2021). **Data-based** methods aim to retrieve clean point cloud data from the noisy version using techniques like DUP-Net and IF-defense (Zhou et al. 2019; Wu et al. 2020). Developing more effective outlier removal methods could potentially enhance the robustness of these techniques, especially data-based methods, as they directly involve outlier removal.

Attribution Analysis on Deep Learning

With the rapid development of deep learning, dozens of deep model attribution analyses were proposed, such as Grad-CAM (Springenberg et al. 2015), Integrated Gradients (Sundararajan, Taly, and Yan 2017) and Deep LIFT (Shrikumar, Greenside, and Kundaje 2017) on 2D image data. A few works explored this analysis in 3D point cloud through techniques of saliency points (Zheng et al. 2019; Wicker and Kwiatkowska 2019) and Shapley values (Shen et al. 2021). We here use attribution analysis as a tool to characterize the noisy samples and empower our method with explainability.

Risk Analysis

We first introduce the concept of point risk. Based on a tail risk measure, we then indicate that noise points could result in a higher level of tail risk in the distribution of point risk.

Point Risk Analysis

Point Risk. Given a point cloud $\mathcal{P} = \{\mathbf{x}_i\}_{i=1}^N$ for $\mathbf{x}_i \in \mathbb{R}^3$, a point cloud classifier $f(\cdot)$ can predict the label of \mathcal{P} by outputting the label scores denoted by $f(\mathcal{P})$. Specifically, let $f(\cdot)$ be a composite function with L intermediate layers, *i.e.*, $f(\mathcal{P}) = f^{(L)}(\dots f^{(2)}(f^{(1)}(\mathcal{P}))\dots)$, the output of l -th layer $f^{(l)}$ is referred to an intermediate feature denoted by $\mathcal{F}^{(l)}$.² Our attribution analysis only focuses on those intermediate features representing each point by an individual vector, which are named as *point features*, namely, $\mathcal{F}^{(l)} = \{\mathbf{x}_i^{(l)}\}_{i=1}^N$ for $\mathbf{x}_i^{(l)} \in \mathbb{R}^d$ with a feature dimension d . Let $S_f(\mathcal{P})$ denote a scoring function designed to evaluate the model output $f(\mathcal{P})$, the attribution analysis toward the points features $\mathcal{F}^{(l)}$ is given by

$$r_i^{(l)} = \left\| \frac{\partial S_f(\mathcal{P})}{\partial \mathbf{x}_i^{(l)}} \right\|_2, \quad (1)$$

where the term $r_i^{(l)}$ refers to the change of $S_f(\mathcal{P})$ that can be obtained by perturbing the l -th layer feature of the i -th point. Similar concepts have been defined differently in various literature, such as class-specific saliency (Simonyan, Vedaldi, and Zisserman 2014), pixel sensitivity (Smilkov et al. 2017), or point saliency (Zheng et al. 2019).

In our study, we define r_i as the *point risk* associated with the i -th point, as it represents the potential risk of changing the point feature, which could subsequently alter the model output. The set of these terms for all points is referred to as $\mathcal{R}^{(l)}$. The final point risks are obtained by averaging the normalized risks of multiple point features, given by

$$r_i = \frac{1}{L'} \sum_{l=0}^{L'} \frac{r_i^{(l)}}{\|\nabla_{\mathcal{F}^{(l)}} S_f(\mathcal{P})\|_F}. \quad (2)$$

where L' is the number of points feature layers and $r_i^{(l)}$ is the unnormalized risk which is then normalized by the Frobenius norm of gradients of the l -th layer feature. The motivation to utilize the multi-layer point feature stems from the efficacy of detecting noise when cascading detectors of several intermediate features (Li and Li 2017).

Risk Analysis of Outliers. We calculate and illustrate the point risk distribution of point clouds with or without outliers. In general, the point cloud with outliers \mathcal{P}' can be formulated from the clean one \mathcal{P} as

$$\mathcal{P}' \subseteq \mathcal{P} \cup \mathcal{O}, \quad (3)$$

where \mathcal{O} is a set of noise points serving as the outliers. We note that this is a common setting and is easily realized, thus threatening the real safety, as detailed in Section 4.3. A typical point risk distribution, as shown in Figure 1 and Figure 2, exhibits a long-tail effect, where most points have low risk while only a few have significant risk. This is consistent with previous works (Qi et al. 2017; Zheng et al. 2019), which suggested the uneven contributions of different points to classification. Furthermore, by analyzing the risk distributions, we make the following hypothesis:

²We define $\mathcal{F}^{(0)} := \mathcal{P}$ for the generality of notation,

HYPOTHESIS 1 *The presence of noise outliers within a point cloud will raise the level of tail risk in its point risk distribution.*

where the tail risk refers to the risks of events accruing rarely but posing a high impact. We believe that this hypothesis is reasonable from two aspects: First, for the model trained on clean point clouds, the noise points, especially those with attack purposes, are potential outliers, tending to affect the model output. Second, previous works in 2D have verified a similar effect by indicating that noise in samples can be amplified in deep models (Lin, Gan, and Han 2019; Liao et al. 2018). To evidence our hypothesis, we present a numerical measure of tail risks.

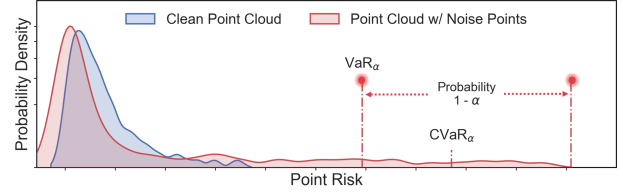


Figure 2: Illustration of CVaR&VaR on risk distribution.

Conditional Value-at-Risk

Conditional Value-at-Risk (CVaR) is a risk measure that quantifies the expected risk that exceeds a certain confidence level, which provides an accurate picture of tail risk (Artzner et al. 1999). To calculate CVaR, we first determine the Value-at-Risk (VaR), which is the maximum risk within a given probability level. Let R be the random variable under a point risk distribution P , the definition for VaR is given by

$$\text{VaR}_\alpha(R) = \inf \{ \gamma : P(R \leq \gamma) \geq \alpha \}, \quad (4)$$

where α is the probability level. CVaR is then defined as the expected risk beyond the VaR_α level, namely,

$$\text{CVaR}_\alpha(R) = \mathbb{E}(R \mid R \geq \text{VaR}_\alpha(R)), \quad (5)$$

where $\mathbb{E}(\cdot)$ denotes the conditional expectation. An illustration of CVaR is shown in Figure 2, showing that it represents the expectation of long-tailed area within a specific level α , *e.g.*, 0.99. A discrete CVaR for point risks is considered in this work. Given the set of risk $\mathcal{R} = \{r_i\}_{i=1}^N$ under a discrete risk distribution \hat{P} , the discrete VaR and CVaR are given by

$$\text{VaR}_\alpha(\mathcal{R}) = \min\{r_i \in \mathcal{R} : \sum_{r_j \in \mathcal{R}} \mathbb{1}(r_i \geq r_j) \hat{P}(r_j) \geq \alpha\}, \quad (6)$$

$$\text{CVaR}_\alpha(\mathcal{R}) = \frac{1}{1-\alpha} \sum_{r_i \in \mathcal{R}} \mathbb{1}(r_i \geq \text{VaR}_\alpha(\mathcal{R})) \hat{P}(r_i) r_i. \quad (7)$$

where $\mathbb{1}(\cdot)$ is the indicator function and \hat{P} is generally set to the empirical distribution, *i.e.*, $\hat{P}(r_i) = \frac{1}{N}$. With CVaR measuring the tail risk, we perform comprehensive statistical tests with the results in experiments. Our findings indicate, with a high degree of confidence, that the CVaR value for a clean sample is inferior to that of a noisy sample. Therefore, the second part of Hypothesis 1 can be evidenced.

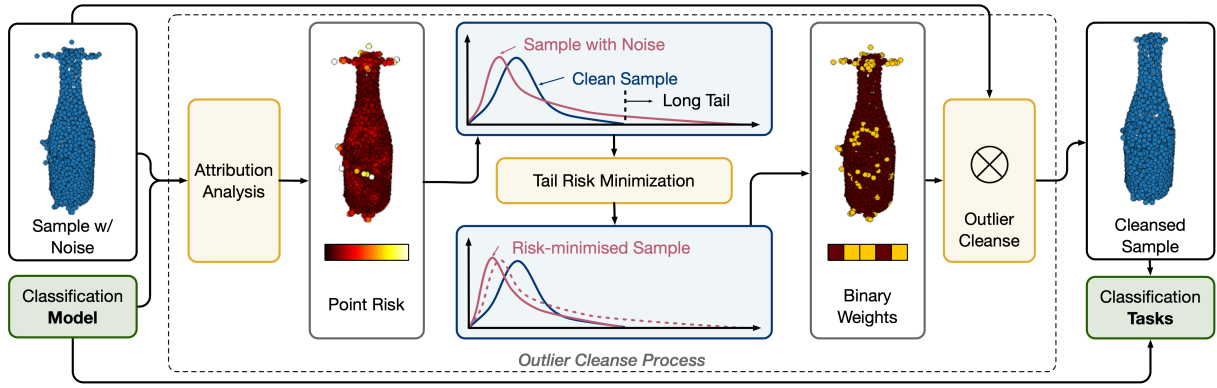


Figure 3: The proposed framework of outlier removal by PointCVaR. Point risks are obtained by entering the noise sample into a trained classification model. Subsequently, an optimization problem is solved to minimize the tail of risk distribution, which leads to binary weights for noise point removal. The processed point cloud is utilized for classification.

Based on the hypothesis, we suggest that **it is possible to remove the noise points by minimizing the tail risk of the point clouds**. Furthermore, we propose to directly use CVaR as an objective to optimize the removal process. This is because CVaR possesses the properties of convexity and monotonicity, making it widely used in optimization problems such as risk management (Pflug 2000).

Point Cloud Outlier Removal

In this section, we present an outlier removal method based on point risk optimization, namely, the minimization of Conditional Value at Risk (CVaR). We also provide detailed insights into its practical implementation.

Outlier Removal by CVaR Optimization

The outlier removal problem based on Formulation (3) is to remove the potential outliers \mathcal{O}' from \mathcal{P}' thus restore \mathcal{P} as much as possible. Let \mathcal{R} denote the risks of noisy points \mathcal{P}' and $\hat{\mathcal{R}}$ denote the risks of retained points after removal, *i.e.* $\mathcal{P}' \setminus \mathcal{O}'$, the outlier removal with the objective of CVaR minimization is indeed a set selection problem given by

$$\min_{\hat{\mathcal{R}} \subseteq \mathcal{R}} \text{CVaR}_\alpha(\hat{\mathcal{R}}) \quad \text{s.t.} \quad |\hat{\mathcal{R}}| \geq N_r, \quad (8)$$

where N_r is the minimum number of retained points. However, since the point removal will induce the change of model output thus the shift of risk values and distribution \hat{P} , this minimization problem is non-trivial. To simplify the problem, we assume the risk values and \hat{P} are *approximately stationary* during the removal, which is theoretically guaranteed by the smoothness assumption of S_f and a large retention ratio, *i.e.*, $\delta = N_r \setminus N \approx 1$. More analysis can be found in Appendix. The problem could also be transformed into a binary optimization

$$\begin{aligned} \min_{w_1, \dots, w_N} \quad & \text{CVaR}_\alpha(\{w_1 r_1, \dots, w_N r_N\}), \\ \text{s.t.} \quad & \sum_{i=1}^N w_i \geq N_r, \quad w_i \in \{0, 1\} \quad \forall i \in \{1, \dots, N\}, \end{aligned} \quad (9)$$

where w_i is the weight of i -th point. A point is retained when its corresponding weight is 1 and removed when it is 0. Subsequently, this optimization can be rendered into a tractable linear programming (LP) problem. This is achieved by two procedures. Firstly, we leverage the conclusions drawn in (Rockafellar, Uryasev et al. 2000) and introduce a real additional variable ζ to obtain the equivalent objective of Equation (9), that is

$$\min_{\zeta, w_1, \dots, w_N} \quad \zeta + \frac{1}{1-\alpha} \sum_{i=1}^N \hat{P}(r_i) [w_i r_i - \zeta]^+. \quad (10)$$

Next, we apply linear relaxation to the binary constraints, ultimately converting the problem into the following format

$$\min_{\zeta, w_1, \dots, w_N, z_1, \dots, z_N} \quad \zeta + \frac{1}{1-\alpha} \sum_{i=1}^N z_i \hat{P}(r_i), \quad (11)$$

$$\begin{aligned} \text{s.t.} \quad & \sum_{i=1}^N w_i \geq N_r, \quad 1 \geq w_i \geq 0, \\ & z_i \geq w_i r_i - \zeta, \quad z_i \geq 0, \quad \forall i \in \{1, \dots, N\}. \end{aligned} \quad (12)$$

where z_1, \dots, z_N are the auxiliary decision variables. During minimization, we set a retention rate to approximately 1 for the stationariness approximation of risks, *e.g.*, $\delta = 0.95$. Finally, we set the largest N_r weights in the solution at 1 and the rest at 0 for the removed points; thus, the set selection problem in Equation (8) for outlier removal can be tractable. We refer to this process as *PointCVaR*.

Implementation Details

Solution Estimation. Although the LP problem of outlier removals in Equation (11) is tractable, it often requires polynomial computational complexity in practice (Karmarkar 1984). To reduce the cost of outlier removal, we employ a risk sorting method to estimate the solution to the problem when \hat{P} is the empirical distribution, which only requires a complexity of $O(N)$. The complete algorithm can be found in Algorithm 1. We note that the estimation is accurate enough in practice, of which a throughout analysis and experimental validation are presented in Appendix.

Algorithm 1: Outlier Removal by PointCVaR

Input: A model f , retention rate δ , scoring function S_f
a noisy point cloud $\mathcal{P}' = \{\mathbf{x}'_i\}_{i=1}^N$.

Output: Processed point cloud $\mathcal{P}' \setminus \mathcal{O}'$

- 1: $\mathcal{R} \leftarrow$ Get $S_f(\mathcal{P}')$ and perform Equation (2) on \mathcal{P}' .
- 2: $r_t \leftarrow \min\{r_i \in \mathcal{R} : \sum_{r_j \in \mathcal{R}} \mathbb{1}(r_i \geq r_j) \geq \lceil \delta N \rceil\}$.
- 3: $\mathcal{O}' \leftarrow \emptyset$.
- 4: **for** $i \in \{1, \dots, N\}$ **do**
- 5: **if** $r_i \geq r_t$ **then**
- 6: $\mathcal{O}' \leftarrow \mathcal{O}' \cup \{\mathbf{x}'_i\}$.
- 7: **end if**
- 8: **end for**
- 9: **return** $\mathcal{P}' \setminus \mathcal{O}'$.

Scoring Function. We introduce the scoring function to calculate the risk of points, which is divided into two terms. The first term is the classification score, which employs the commonly used cross-entropy loss on the model prediction. This is denoted by $S_c(f(P))$. The second term is the geometric score, for which we incorporate a differentiable score that measures the uniformity of point clouds. The formula is

$$S_g(\mathcal{P}) = \sqrt{\frac{1}{N} \sum_i^N (d_i - \bar{d})^2}, \quad (13)$$

$$d_i = \frac{1}{|\mathcal{N}_i|} \sum_{\mathbf{x}_j \in \mathcal{N}_i} \|\mathbf{x}_i - \mathbf{x}_j\|_2, \quad \bar{d} = \frac{1}{N} \sum_i^N d_i, \quad (14)$$

where $\mathcal{N}_i \subseteq \mathcal{P}$ refers to the nearest neighbors of $x_i \in \mathcal{P}$. Therefore, d_i measures the neighborhood distance of a point, and $S_g(\mathcal{P})$ is the variance of d_i . The final scoring function is given by $S_f(\mathcal{P}) = S_c(\mathcal{P}, f) + \lambda S_g(\mathcal{P})$ with a tunable parameter λ . It is noted that the geometric score only affects the risks of input points rather than other point features.

Multi-step Outlier Removal. In order to better satisfy the condition of approximate stationariness of risks for the LP formulation, we propose a multi-step removal method where the retention rate of each step is closer to 1 than δ . Specifically, it is achieved by executing Algorithm 1 by I times. The retention rate of each step is $\delta^{\frac{1}{I}}$. Although this results in a linear increase in computational complexity, experiments suggest that it surpasses single-step removal due to a superior approximation. In subsequent discussions, we refer to the multi-step method and the original single-step method as *multi-step* and *vanilla*, respectively.

Experiment

Experiment Settings

Dataset and Model. Our experiments are conducted based on ModelNet40 (Wu et al. 2015) (MN) and ShapeNet (Chang et al. 2015) (SN) for classification task. The train\test splits of them follow the implementation of PointNet (Qi et al. 2017), which are 9843\2468 for MN and 12128\2874 for SN. Each clean point cloud in the dataset contains 1024 points while this number may vary for the

noisy data. For the models, we utilize PointNet, DGCNN, PointCloudTransformer (PCT) and GDANet by following the official implementations (Qi et al. 2017; Wang et al. 2019; Guo et al. 2021; Xu et al. 2021). Unless otherwise specified, we perform experiments on the models trained by the protocols as detailed by (Goyal et al. 2021).

Noise Setting. We implement three types of point cloud outliers that can occur in real-world scenarios, which are:

- **Random Outlier (Rand)** comprises of random global (Glb) and local (Loc) noise that imitate background points in real-world data (Ren, Pan, and Liu 2022), respectively.
- **Adversarial Outlier (Adv)** is created by implementing the point addition attacks listed in (Xiang, Qi, and Li 2019), which includes adding points by Hausdorff distance (HD), Chamfer divergence (CD), objects (Obj), and clusters (Cls) methods. Studies have demonstrated that such attacks can be physically realizable (Tu et al. 2020).
- **Backdoor Trigger (BA)** involves a small sphere located at a specific coordinate, as proposed in (Li et al. 2021). This outlier can impact the models trained dataset corrupted by poison-label (PL) or clean-label (CL) methods, which can be easily generated in the real world (Li et al. 2021).

We add the above noises into the data with more details including hardware configurations in Appendix. Moreover, we also evaluate the effectiveness of combining existing methods and ours on resisting various adversarial noises including perturbation attacks by C&W (CW) (Xiang, Qi, and Li 2019), k NN (Tsai et al. 2020), shape-invariant (SI) (Huang et al. 2022) and the point drop attack (Zheng et al. 2019).

Parameters Setting. We set δ to be 0.95 and 0.92 for the multi-step and vanilla removal method, $\lambda = 1.0$ for the scoring function, and $I = 20$ for the multi-step method. Further investigations into the effects of these parameters are presented in *More Studies* section.

Evaluation Protocol. During the training phase, we train the model on either a clean dataset or one that is backdoor-corrupted. In the testing phase, we first perform an outlier removal on the noisy test set. Subsequently, we report the test accuracy of the model on the processed data. A higher accuracy is indicative of superior performance of the removal method. We refer the protocol as removal-and-classification.

Main Results

Removal and Classification. We conduct extensive tests for our outlier removal method based on different noisy versions of ModelNet40 and ShapeNet. For better comparison, we also implement three outlier removal methods: Random Sampling (RS) (Yang et al. 2019), Radius Outlier Removal (ROR) (Rusu and Cousins 2011), and SOR (Rusu 2010). Additionally, we re-implement the outlier removal network in PointCleanNet (PCN) (Rakotosaona et al. 2020) as a baseline for learning-based methods. We perform these methods based on the proposed parameters and training process, while for ROR, we set $r_{ror} = 0.1$ and $n_{ror} = 4$ as detailed in Appendix. The results in Table 1 and Table 2 show that our multi-step method consistently outperforms all baseline methods, particularly as the only method that effectively removes backdoor triggers achieving above 80% accuracy for

Methods	Rand (MN)		Adv (MN)				BA (MN)		Rand(SN)		Adv (SN)				BA (SN)	
	Glb	Loc	HD	CD	Obj	Cls	PL	CL	Glb	Loc	HD	CD	Obj	Cls	PL	CL
Raw	5.4	52.6	0.9	2.4	0.4	0.3	4.1	39.2	11.0	73.8	1.7	3.7	2.6	4.1	10.0	34.7
RS	8.4	52.6	73.7	77.9	30.1	45.1	4.1	36.8	12.2	74.6	84.2	83.0	43.4	57.4	10.0	33.0
ROR	82.1	64.7	74.5	81.9	35.0	59.4	4.1	35.3	97.2	83.6	88.1	93.8	47.2	72.1	10.0	34.5
SOR	72.3	<u>65.7</u>	82.7	84.9	36.9	63.9	4.1	37.3	78.0	<u>84.3</u>	95.8	96.0	52.2	75.2	10.0	34.6
PCN	45.6	56.4	69.8	57.0	48.2	61.9	4.1	38.1	46.3	78.6	43.8	38.7	34.1	35.9	<u>15.6</u>	36.4
Vanilla	<u>82.4</u>	63.9	<u>86.2</u>	<u>88.7</u>	<u>48.9</u>	<u>63.9</u>	<u>4.7</u>	<u>39.6</u>	<u>97.6</u>	<u>82.7</u>	<u>97.4</u>	<u>97.3</u>	<u>61.4</u>	<u>76.7</u>	10.0	<u>37.2</u>
Multistep	<u>88.0</u>	<u>70.0</u>	<u>84.0</u>	<u>85.3</u>	<u>62.8</u>	<u>67.8</u>	<u>85.7</u>	<u>78.3</u>	<u>97.7</u>	<u>85.5</u>	<u>97.3</u>	<u>96.2</u>	<u>67.5</u>	<u>79.0</u>	<u>88.9</u>	<u>75.4</u>

Table 1: Outlier removal method comparison based on PointNet accuracy (%) on ModelNet40 (MN) and ShapeNet (SN) corrupted by different outliers, where the top two accuracies are highlighted by underlines.

backdoor attacks across all models and datasets. Furthermore, the vanilla method serves as a cost-effective removal option, exhibiting decent performance on all models at a relatively low computational cost. In particular, its execution speed rivals that of SOR, surpassing PCN by more than 10 times, even without considering the training time of PCN. More runtime profiles are provided in Appendix.

Noise	Model	Raw	ROR	SOR	PCN	V	M
Glb (MN)	DG	65.7	83.6	87.4	75.2	<u>89.1</u>	<u>89.6</u>
	PC	49.7	83.3	85.9	69.2	<u>88.7</u>	<u>90.2</u>
	GD	63.7	82.7	88.3	74.0	<u>88.6</u>	<u>90.3</u>
CD (MN)	DG	1.5	69.1	73.0	71.8	<u>84.0</u>	<u>80.0</u>
	PC	0.4	82.6	85.1	79.4	<u>88.5</u>	<u>87.3</u>
	GD	3.8	63.6	74.6	71.7	<u>83.9</u>	<u>82.2</u>
PL (MN)	DG	4.1	4.1	14.6	4.1	<u>83.7</u>	<u>88.6</u>
	PC	4.1	48.1	55.0	52.8	<u>83.6</u>	<u>87.7</u>
	GD	4.1	8.8	37.3	31.1	<u>87.6</u>	<u>88.7</u>
Glb (SN)	DG	59.1	<u>97.8</u>	88.0	79.9	96.7	<u>98.1</u>
	PC	36.5	<u>98.3</u>	93.4	71.8	97.6	<u>98.5</u>
	GD	22.3	97.1	84.7	65.6	<u>97.2</u>	<u>97.8</u>
CD (SN)	DG	4.1	88.0	92.1	75.8	<u>95.6</u>	<u>93.7</u>
	PC	0.1	90.6	94.7	72.3	<u>96.4</u>	<u>95.9</u>
	GD	1.7	88.9	92.8	68.2	<u>94.8</u>	<u>94.1</u>
PL (SN)	DG	10.0	10.0	34.8	34.4	<u>99.0</u>	<u>97.9</u>
	PC	10.0	58.1	87.0	88.7	<u>93.5</u>	<u>96.1</u>
	GD	10.0	16.5	73.1	93.6	<u>97.9</u>	<u>98.8</u>

Table 2: Method comparison (V for vanilla and M for multistep) based on DGCNN (DG), PCT (PCT) and GDANet (GD) by accuracy (%) on noisy ModelNet40 and ShapeNet.

Plugin Performance. As outlier removal is frequently used for point cloud pre-processing (Rusu 2010), we implement our approach as a pre-processing plugin to existing robust point cloud classification methods, thereby extensively improving their robustness to adversarial attacks. The plugin is applied to three methods, DUP-Net, PointCutMix (PCM) (Zhang et al. 2022) and PGD adversarial training (PGD) (Sun et al. 2021). Particularly, we replace SOR with our method in DUP-Net, and also preprocess the input for PCM/PGD trained model prior to inference. For lightweight consideration of plugins, we only utilize the vanilla method in the experiments. Results in Table 3 indicate that the proposed technique consistently improves the performance of DUP-Net, PCM, and PGD by a significant margin, partic-

ularly for the latter two model-based methods where they are vulnerable to adaptive attacks. Notably, PGD achieves an average accuracy of 80% and 90% on ModelNet40 and ShapeNet with adversarial noise, respectively. Given the low cost of this method, it demonstrates substantial potential as a plugin module to enhance point cloud classification.

Method	Ours	CD	Cls	kNN	CW	SI	Drop
DN (MN)	×	85.3	63.7	77.1	85.9	40.1	61.1
	✓	<u>87.3</u>	<u>64.5</u>	<u>81.8</u>	<u>86.2</u>	<u>81.7</u>	<u>62.7</u>
PCM (MN)	×	2.8	4.3	39.2	1.8	2.3	76.2
	✓	<u>89.8</u>	<u>87.2</u>	<u>75.6</u>	<u>89.1</u>	<u>80.3</u>	<u>76.3</u>
PGD (MN)	×	3.8	6.6	45.3	1.9	4.2	80.3
	✓	<u>88.9</u>	<u>80.3</u>	<u>82.6</u>	<u>87.7</u>	<u>78.3</u>	<u>80.3</u>
DN (SN)	×	97.1	74.9	95.6	96.5	45.9	90.3
	✓	<u>97.1</u>	<u>75.3</u>	<u>95.9</u>	<u>96.8</u>	<u>92.5</u>	<u>90.6</u>
PCM (SN)	×	2.8	4.5	39.2	1.8	2.5	78.4
	✓	<u>90.5</u>	<u>85.2</u>	<u>85.7</u>	<u>87.5</u>	<u>88.7</u>	<u>81.9</u>
PGD (SN)	×	4.4	9.7	83.2	1.6	31.6	92.6
	✓	<u>97.2</u>	<u>88.9</u>	<u>94.7</u>	<u>90.6</u>	<u>94.5</u>	<u>93.5</u>

Table 3: PointNet accuracy (%) of three robust classification methods on ModelNet40 and ShapeNet with diverse adversarial noises. The comparison is made between the original method and the one enhanced by our method as plugin.

More Studies

Hypothesis Test. We conduct the statistical testing for Hypothesis 1, where $CVaR_{0.99}$ is used to measure the tail risk. Figure 4 showcases that the risks of clean samples generally have lower $CVaR_{0.99}$ compared to the samples with noise points. Furthermore, we use a paired Wilcoxon signed test to investigate the difference between the risk of a clean sample and its noise counterparts. The results showed that all p-values obtained from the one-sided tests are less than 0.01. Therefore, we reject the null hypothesis and conclude that the **tail risk of a sample with noise is statistically greater than its corresponding clean sample.**

Effect of Retention Rate. In Figure 5, the removal process is visually depicted. It is evident that the vanilla method sometimes struggles to eliminate outliers completely, particularly backdoor triggers. Conversely, the multi-step method consistently removes noise. Nevertheless, multi-step tends to over-remove clean points when dealing with outliers of

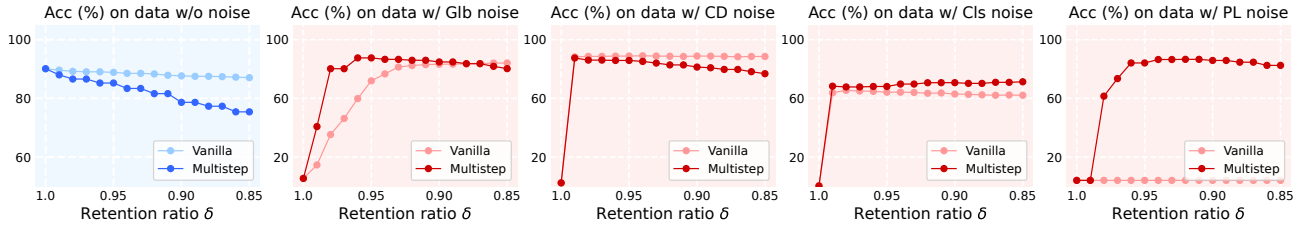


Figure 5: Accuracy on ModelNet40 without or with different noise vs. retention rate of PointCVaR.

small proportion such as CD. Consequently, we present a further analysis of the impact of the retention rate δ in Figure 5. This figure demonstrates how model accuracy on processed samples changes as δ decreases. Although it generally outperforms the vanilla method in terms of performance, the multi-step method leads to a more pronounced decrease in accuracy in clean samples when δ is low.

We hypothesize that this is due to the removal of so-called salient points from clean samples, as suggested in prior work (Zheng et al. 2019). However, the decrease in accuracy for clean samples is slower than the increase in accuracy on noisy samples, enabling us to always locate a suitable δ that yields satisfactory performance. For instance, $\delta = 0.95$ for the multi-step method and $\delta = 0.92$ for the vanilla method yield accuracies of 85% and 88.5% on clean samples, respectively, alongside acceptable accuracy. The differing rates of change can be attributed to the lower tail risks of clean samples compared to noisy ones, which results in greater stability during the removal of high-risk points.

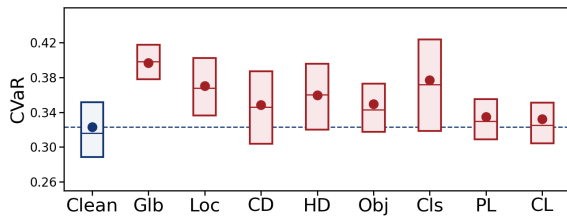


Figure 4: $CVaR_{0.99}$ boxplots (Mean, median and 25\75 quantiles) of clean sets risks (blue) and noisy sets risks (red).

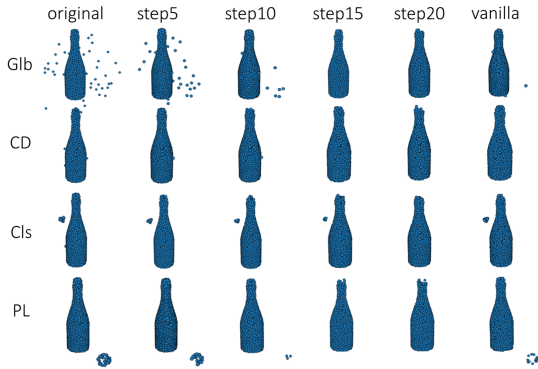


Figure 5: Visualization of PointCVaR removal process.

Method	CSF	λ	I	MPF	Glb	CD	PL
Vanilla	CE	1.0	N.A.	✓	82.4	88.7	4.7
Vanilla	Max	1.0	N.A.	✓	76.7	88.5	4.1
Vanilla	Sum	1.0	N.A.	✓	77.2	88.3	4.1
Vanilla	CE	0.0	N.A.	✓	59.5	88.5	4.1
Vanilla	CE	1.0	N.A.	×	81.4	88.3	4.3
Multstep	CE	1.0	5	✓	88.0	86.9	35.8
Multstep	CE	0.0	20	✓	87.2	85.5	84.2
Multstep	CE	1.0	20	×	86.8	86.3	78.1
Multstep	CE	1.0	20	✓	88.0	85.3	85.7

Table 4: Ablation study based on the accuracy of data with global, add CD and backdoor triggers. CSF and MPF represent the classification scoring function S_c and the use of multi-layer point features for risk calculation, respectively. CE is the cross-entropy loss, while the Max and Sum are the maximum and sum of output logits, respectively.

Ablation Study. We conduct ablation studies on various elements within our outlier removal approach. As demonstrated in Table 4, the selection of the classification score S_c has a minimal impact on our method, only causing a slight performance variation in the vanilla method’s handling of global noise. On the other hand, the introduction of the geometric score S_g , *i.e.*, $\lambda = 1$, consistently enhances our method’s capacity to remove global noise, without affecting other types of noise. Also, our findings indicate that, compared to the vanilla method, the multi-step method exhibits less sensitivity to changes in these components. The only factor substantially affecting the performance of the multi-step method is the number of iterations I , as a successful removal of backdoor triggers necessitates a higher I .

Conclusion

In conclusion, the noise effect has cast a shadow over the reliability of 3D deep learning on point clouds in industrial applications. Our proposed outlier removal method leverages trained classification models to effectively filter out many types of outliers in point clouds. By utilizing gradient-based attribution analysis and tail risk minimization, we formulate the process into an optimization problem and use linear programming to obtain binary weights for input points. This method consistently improves the classification accuracy of deep models on different versions of noisy point clouds. It can also be combined with other methods to defend against a larger range of adversarial noises at a low cost.

Acknowledgements

The research students are supported by the National University of Singapore (NUS) Research Scholarship.

References

- Artzner, P.; Delbaen, F.; Eber, J.-M.; and Heath, D. 1999. Coherent measures of risk. *Mathematical finance*, 9(3): 203–228.
- Chang, A. X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; Xiao, J.; Yi, L.; and Yu, F. 2015. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago.
- Chen, X.; Ma, H.; Wan, J.; Li, B.; and Xia, T. 2017. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1907–1915.
- Chen, Y.; Hu, V. T.; Gavves, E.; Mensink, T.; Mettes, P.; Yang, P.; and Snoek, C. G. 2020. Pointmixup: Augmentation for point clouds. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, 330–345. Springer.
- Dong, X.; Chen, D.; Zhou, H.; Hua, G.; Zhang, W.; and Yu, N. 2020. Self-robust 3d point recognition via gather-vector guidance. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11513–11521. IEEE.
- Goyal, A.; Law, H.; Liu, B.; Newell, A.; and Deng, J. 2021. Revisiting Point Cloud Shape Classification with a Simple and Effective Baseline. *International Conference on Machine Learning*.
- Guo, M.; Cai, J.; Liu, Z.; Mu, T.; Martin, R. R.; and Hu, S. 2021. Pct: Point cloud transformer. *Computational Visual Media*, 7: 187–199.
- Hamdi, A.; Rojas, S.; Thabet, A.; and Ghanem, B. 2020. Advpc: Transferable adversarial perturbations on 3d point clouds. In *European Conference on Computer Vision*, 241–257. Springer.
- Huang, Q.; Dong, X.; Chen, D.; Zhou, H.; Zhang, W.; and Yu, N. 2022. Shape-invariant 3d adversarial point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15335–15344.
- Karmarkar, N. 1984. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, 302–311.
- Li, X.; Chen, Z.; Zhao, Y.; Tong, Z.; Zhao, Y.; Lim, A.; and Zhou, J. T. 2021. Pointba: Towards backdoor attacks in 3d point cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16492–16501.
- Li, X.; and Li, F. 2017. Adversarial examples detection in deep networks with convolutional filter statistics. In *Proceedings of the IEEE international conference on computer vision*, 5764–5772.
- Liao, F.; Liang, M.; Dong, Y.; Pang, T.; Hu, X.; and Zhu, J. 2018. Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1778–1787.
- Lin, J.; Gan, C.; and Han, S. 2019. Defensive quantization: When efficiency meets robustness. In *International Conference on Learning Representations*, ICLR.
- Liu, D.; Yu, R.; and Su, H. 2019. Extending adversarial attacks and defenses to deep 3d point cloud classifiers. In *2019 IEEE International Conference on Image Processing (ICIP)*, 2279–2283. IEEE.
- Liu, H.; Jia, J.; and Gong, N. Z. 2021. Pointguard: Provably robust 3d point cloud classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6186–6195.
- Pflug, G. C. 2000. Some remarks on the value-at-risk and the conditional value-at-risk. *Probabilistic constrained optimization: Methodology and applications*, 272–281.
- Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.
- Rakotosaona, M.-J.; La Barbera, V.; Guerrero, P.; Mitra, N. J.; and Ovsjanikov, M. 2020. Pointcleannet: Learning to denoise and remove outliers from dense point clouds. In *Computer graphics forum*. Wiley Online Library.
- Ren, J.; Pan, L.; and Liu, Z. 2022. Benchmarking and analyzing point cloud classification under corruptions. In *International Conference on Machine Learning*, 18559–18575. PMLR.
- Rockafellar, R. T.; Uryasev, S.; et al. 2000. Optimization of conditional value-at-risk. *Journal of risk*, 2: 21–42.
- Rusu, R. B. 2010. Semantic 3D object maps for everyday manipulation in human living environments. *KI-Künstliche Intelligenz*, 24: 345–348.
- Rusu, R. B.; and Cousins, S. 2011. 3d is here: Point cloud library (pcl). In *2011 IEEE international conference on robotics and automation*, 1–4. IEEE.
- Shen, W.; Ren, Q.; Liu, D.; and Zhang, Q. 2021. Interpreting representation quality of dnns for 3d point cloud processing. *Advances in Neural Information Processing Systems*, 34: 8857–8870.
- Shrikumar, A.; Greenside, P.; and Kundaje, A. 2017. Learning important features through propagating activation differences. In *International conference on machine learning*, 3145–3153. PMLR.
- Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Deep inside convolutional networks: visualising image classification models and saliency maps. In *Proceedings of the International Conference on Learning Representations (ICLR)*. ICLR.
- Smilkov, D.; Thorat, N.; Kim, B.; Viégas, F.; and Wattenberg, M. 2017. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*.
- Springenberg, J.; Dosovitskiy, A.; Brox, T.; and Riedmiller, M. 2015. Striving for Simplicity: The All Convolutional Net. In *ICLR (workshop track)*.

- Sun, J.; Cao, Y.; Choy, C. B.; Yu, Z.; Anandkumar, A.; Mao, Z. M.; and Xiao, C. 2021. Adversarially robust 3d point cloud recognition using self-supervisions. *Advances in Neural Information Processing Systems*, 34: 15498–15512.
- Sun, J.; Koenig, K.; Cao, Y.; Chen, Q. A.; and Mao, Z. M. 2020. On adversarial robustness of 3d point cloud classification under adaptive attacks. *arXiv preprint arXiv:2011.11922*.
- Sun, J.; Zhang, Q.; Kailkhura, B.; Yu, Z.; Xiao, C.; and Mao, Z. M. 2022. Benchmarking Robustness of 3D Point Cloud Recognition Against Common Corruptions. *arXiv preprint arXiv:2201.12296*.
- Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, 3319–3328. PMLR.
- Tsai, T.; Yang, K.; Ho, T.-Y.; and Jin, Y. 2020. Robust adversarial objects against deep learning models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 954–962.
- Tu, J.; Ren, M.; Manivasagam, S.; Liang, M.; Yang, B.; Du, R.; Cheng, F.; and Urtasun, R. 2020. Physically Realizable Adversarial Examples for LiDAR Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13716–13725.
- Uy, M. A.; Pham, Q.-H.; Hua, B.-S.; Nguyen, T.; and Yeung, S.-K. 2019. 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*, 1588–1597.
- Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5): 1–12.
- Wicker, M.; and Kwiatkowska, M. 2019. Robustness of 3d deep learning in an adversarial setting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 11767–11775.
- Wu, Z.; Duan, Y.; Wang, H.; Fan, Q.; and Guibas, L. J. 2020. If-defense: 3d adversarial point cloud defense via implicit function based restoration. *arXiv preprint arXiv:2010.05272*.
- Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1912–1920.
- Xiang, C.; Qi, C. R.; and Li, B. 2019. Generating 3d adversarial point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9136–9144.
- Xiang, Z.; Miller, D. J.; Chen, S.; Li, X.; and Kesidis, G. 2021. A backdoor attack against 3d point cloud classifiers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7597–7607.
- Xu, M.; Zhang, J.; Zhou, Z.; Xu, M.; Qi, X.; and Qiao, Y. 2021. Learning geometry-disentangled representation for complementary understanding of 3d object point cloud. In *Proceedings of the AAAI conference on artificial intelligence*, 3056–3064.
- Yang, J.; Zhang, Q.; Fang, R.; Ni, B.; Liu, J.; and Tian, Q. 2019. Adversarial attack and defense on point sets. *arXiv preprint arXiv:1902.10899*.
- Zhang, J.; Chen, L.; Ouyang, B.; Liu, B.; Zhu, J.; Chen, Y.; Meng, Y.; and Wu, D. 2022. Pointcutmix: Regularization strategy for point cloud classification. *Neurocomputing*, 505: 58–67.
- Zhao, Y.; Wu, Y.; Chen, C.; and Lim, A. 2020. On Isometry Robustness of Deep 3D Point Cloud Models under Adversarial Attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1201–1210.
- Zheng, T.; Chen, C.; Yuan, J.; Li, B.; and Ren, K. 2019. Pointcloud saliency maps. In *Proceedings of the IEEE International Conference on Computer Vision*, 1598–1606.
- Zhou, H.; Chen, K.; Zhang, W.; Fang, H.; Zhou, W.; and Yu, N. 2019. DUP-Net: Denoiser and upsampler network for 3D adversarial point clouds defense. In *Proceedings of the IEEE International Conference on Computer Vision*, 1961–1970.