PointGuard: Provably Robust 3D Point Cloud Classification

Hongbin Liu* Jinyuan Jia* Neil Zhenqiang Gong
Duke University
{hongbin.liu, jinyuan.jia, neil.gong}@duke.edu

Abstract

3D point cloud classification has many safety-critical applications such as autonomous driving and robotic grasping. However, several studies showed that it is vulnerable to adversarial attacks. In particular, an attacker can make a classifier predict an incorrect label for a 3D point cloud via carefully modifying, adding, and/or deleting a small number of its points. Randomized smoothing is state-of-the-art technique to build certifiably robust 2D image classifiers. However, when applied to 3D point cloud classification, randomized smoothing can only certify robustness against adversarially modified points.

In this work, we propose PointGuard, the first defense that has provable robustness guarantees against adversarially modified, added, and/or deleted points. Specifically, given a 3D point cloud and an arbitrary point cloud classifier, our PointGuard first creates multiple subsampled point clouds, each of which contains a random subset of the points in the original point cloud; then our PointGuard predicts the label of the original point cloud as the majority vote among the labels of the subsampled point clouds predicted by the point cloud classifier. Our first major theoretical contribution is that we show PointGuard provably predicts the same label for a 3D point cloud when the number of adversarially modified, added, and/or deleted points is bounded. Our second major theoretical contribution is that we prove the tightness of our derived bound when no assumptions on the point cloud classifier are made. Moreover, we design an efficient algorithm to compute our certified robustness guarantees. We also empirically evaluate PointGuard on ModelNet40 and ScanNet benchmark datasets.

1. Introduction

3D point cloud, which comprises a set of 3D points, is a crucial data structure in modelling a 3D shape or object. In recent years, we have witnessed an increasing interest in 3D point cloud classification [23, 17, 24, 30] because it has many applications, such as robotic grasping [28], autonomous driving [2, 36], etc.. However, multiple recent studies [34, 31, 38, 39, 35, 20] showed that 3D point cloud classifiers are vulnerable to adversarial attacks. In particular, given a 3D point cloud, an attacker can carefully modify, add, and/or delete a small number of points such that a 3D point cloud classifier predicts an incorrect label for it. We can categorize these attacks into four types based on the capability of an attacker: point *modification*, *addition*, *deletion*, and *perturbation* attacks. In particular, in a point modification/addition/deletion attack [34, 31, 38, 35], an attacker can only modify/add/delete points in a 3D point cloud. An attacker, however, can apply one or more of the above three operations, i.e., modification, addition, and deletion, to a 3D point cloud in a point perturbation attack. Figure 1 illustrates the point modification, addition, and deletion attacks. These adversarial attacks pose severe security concerns to point cloud classification in safety-critical applications.

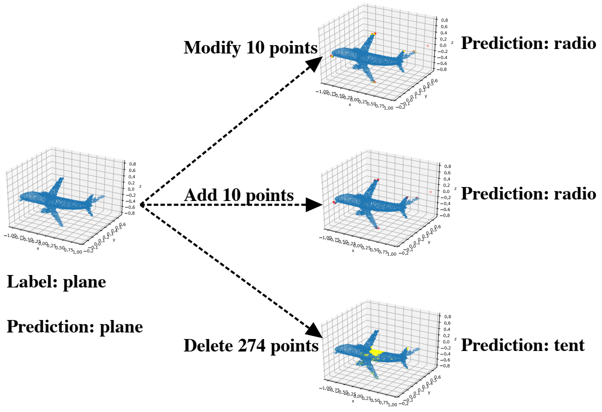


Figure 1: Top: point modification attack. Middle: point addition attack. Bottom: point deletion attack. Red points are added and yellow points are deleted.

Several *empirical defenses* [18, 40, 35, 6] have been proposed to mitigate the attacks. Roughly speaking, these defenses aim to detect the attacks or train more robust point cloud classifiers. For instance, Zhou et al. [40] proposed a defense called DUP-Net, whose key step is to detect outlier points and discard them before classifying a point cloud.

*The first two authors made equal contributions.

These defenses, however, lack provable robustness guarantees and are often broken by more advanced attacks. For instance, Ma et al. [20] proposed a joint gradient based attack and showed that it can achieve high attack success rates even if DUP-Net [40] is deployed.

Therefore, it is urgent to study certified defenses that have provable robustness guarantees. We say a point cloud classifier is provably robust if it certifiably predicts the same label for a point cloud when the number of modified, added, and/or deleted points is no larger than a threshold. Randomized smoothing [4] is state-of-the-art technique for building provably robust 2D image classifiers. For instance, via adding Gaussian noise to a 2D image, randomized smoothing provably predicts the same label for the image when the ℓ_2-norm of the adversarial perturbations added to the image is no larger than a threshold. Randomized smoothing can be applied to point cloud classification. For instance, we can add Gaussian noise to each point of a point cloud and randomized smoothing can predict the same label for the point cloud when the adversarial modification of its points is bounded. However, randomized smoothing requires the size of the input (e.g., the number of pixels in a 2D image or number of points in a 3D point cloud) remains unchanged under adversarial attacks. Therefore, randomized smoothing is only applicable to certify robustness against the point modification attacks that do not change the size of a point cloud, leaving the other three types of attacks untouched.

Our work: In this work, we propose PointGuard, the first defense that has provable robustness guarantees against point modification, addition, deletion, and perturbation attacks. Suppose we are given a 3D point cloud and an arbitrary point cloud classifier. PointGuard first creates a *subsampled point cloud*, which contains a random subset of k points subsampled from the original point cloud. Since the subsampled point cloud is random, its label predicted by the point cloud classifier is also random. We use p_i (called *label probability*) to denote the probability that the point cloud classifier predicts label i for the random subsampled point cloud. Our PointGuard predicts the label that has the largest label probability for the original 3D point cloud.

Our major theoretical contributions are twofold. First, we show that, with any point cloud classifier, our PointGuard provably predicts the same label for a point cloud when the number of modified, added, and/or deleted points is no larger than a threshold. We call the threshold *certified perturbation size*. Note that the certified perturbation size may be different for different testing point clouds and point cloud classifiers. We derive the certified perturbation size via leveraging the Neyman-Pearson Lemma [21]. Second, we prove that, if no assumptions on the point cloud classifier are made, our derived certified perturbation size is tight, i.e., it is impossible to derive a certified perturbation size larger than ours.

Our derived certified perturbation size for a point cloud is the solution to an optimization problem, which relies on the point cloud's label probabilities. However, it is challenging to compute the exact label probabilities in practice since it requires predicting the labels for an exponential number of subsampled point clouds. In particular, computing the exact label probabilities for a point cloud requires predicting the labels for $\binom{n}{k}$ subsampled point clouds if the point cloud contains n points. To address the challenge, we develop a Monte-Carlo algorithm to estimate the lower and upper bounds of the label probabilities with probabilistic guarantees via predicting labels for $N << \binom{n}{k}$ subsampled point clouds. Given the estimated label probability bounds, we solve the optimization problem to obtain the certified perturbation size.

We empirically evaluate PointGuard on ModelNet40 and ScanNet. To demonstrate the generality of PointGuard, we consider two point cloud classifiers, i.e., PointNet [23] and DGCNN [30]. We adopt *certified accuracy* as our evaluation metric. In particular, the certified accuracy at r perturbed points is the fraction of the testing point clouds whose labels are correctly predicted and whose certified perturbation sizes are no smaller than r. Since the certified accuracy of a standard point cloud classifier is unknown, we measure a standard point cloud classifier using its *empirical accuracy* under an empirical attack, i.e., we use an empirical attack to perturb the testing point clouds and use the point cloud classifier to classify them. Our experimental results show that the certified accuracy of PointGuard is substantially higher than the empirical accuracy of a standard point cloud classifier in many cases. For instance, on ModelNet40, PointNet achieves 0% empirical accuracy while PointGuard with $k = 16$ achieves 69.7% certified accuracy when an attacker can arbitrarily modify 30 points of each testing point cloud. We also compare PointGuard with randomized smoothing for point modification attacks, as randomized smoothing is only applicable to such attacks. Our results show that PointGuard substantially outperforms randomized smoothing, e.g., randomized smoothing achieves 0% certified accuracy under the above setting.

In summary, our key contributions are as follows:

- We propose PointGuard, the first 3D point cloud classification system that is provably robust against different types of adversarial attacks.

- We derive the certified robustness guarantee of PointGuard and prove its tightness. Moreover, we design an algorithm to efficiently compute our certified robustness guarantee.

- We evaluate our PointGuard on two datasets.

2. Background and Related Work

3D point cloud classification: A 3D point cloud is an unordered set of points sampled from the surface of a 3D object or shape. We use $T = \{O_i \mid i = 1, 2, \cdots, n\}$ to denote a 3D point cloud, where each point O_i is a vector that contains the (x, y, z) coordinates and possibly some other features, e.g., colours. In 3D point cloud classification, given a point cloud T as input, a classifier f predicts a label $y \in \{1, 2, \cdots, c\}$ for it. For instance, the label could represent the type of 3D object from which the point cloud T is sampled. Formally, we have $y = f(T)$. Many deep learning classifiers (e.g., [23, 24, 17, 30]) have been proposed for 3D point cloud classification. For instance, Qi et al. [23] proposed PointNet, which can directly consume 3D point cloud. Roughly speaking, PointNet first applies input and feature transformations to the input points, and then aggregates point features by max pooling. One important characteristic of PointNet is permutation invariant. In particular, given a 3D point cloud, the predicted label does not rely on the order of the points in the point cloud.

Adversarial attacks to 3D point cloud classification: Multiple recent works [34, 31, 38, 35, 27, 8, 37, 39, 12] showed that 3D point cloud classification is vulnerable to (physically feasible) adversarial attacks. Roughly speaking, given a 3D point cloud, these attacks aim to make a 3D point cloud classifier misclassify it via carefully modifying, adding, and/or deleting some points from it. Xiang et al. [34] proposed point perturbation and addition attacks. For instance, they showed that PointNet [23] can be fooled by adding a limited number of synthesized point clusters with meaningful shapes such as balls to a point cloud. Yang et al. [35] explored point modification, addition, and deletion attacks. In particular, their point modification attack is inspired by gradient-guided attack methods, which were designed to attack 2D image classification. Their point addition and deletion attacks aim to add or remove the *critical* points, which can be identified by their label-dependent importance scores obtained by computing the gradient of a classifier's output with respect to the input. Wicker et al. [31] proposed a point deletion attack which also leveraged critical points. Specifically, they developed an algorithm to identify critical points in a random and iterative manner. Ma et al. [20] proposed a joint gradient based attack and showed that the proposed attack can break an empirical defense [40] on multiple 3D point cloud classifiers.

Existing empirical defenses: Several empirical defenses [18, 40, 35, 6, 32, 26] have been proposed to defend against adversarial attacks. Roughly speaking, these defenses aim to detect the attacks or train more robust point cloud classifiers. For instance, Zhou et al. [40] proposed DUP-Net, whose key idea is to detect and discard outlier points before classifying a point cloud. Dong et al. [6] designed a new self-robust 3D point recognition network. In particular, the network first extracts local features from the input point cloud and then uses a self-attention mechanism to aggregate these local features, which could ignore adversarial local features. Liu et al. [18] generalized adversarial training [7] to build more robust point cloud classifiers. However, these empirical defenses lack certified robustness guarantees and are often broken by advanced adaptive attacks.

Existing certified defenses: To the best of our knowledge, there are no certified defenses against adversarial attacks for 3D point cloud classification. We note that, however, many certified defenses against adversarial attacks have been proposed for 2D image classification. Among these defenses, randomized smoothing [1, 19, 13, 16, 4, 25, 22, 14, 15, 11, 29] is state-of-the-art because it is scalable to large-scale neural networks and applicable to arbitrary classifiers. Roughly speaking, randomized smoothing adds random noise (e.g., Gaussian noise) to an input image before classifying it; and randomized smoothing provably predicts the same label for the image when the adversarial perturbation added to the image is bounded under certain metrics, e.g., ℓ_2 norm. We can leverage randomized smoothing to certify robustness of point cloud classification via adding random noise to each point of a point cloud. However, randomized smoothing requires the size of the input (e.g., the number of points in a point cloud) remains unaltered under adversarial attacks. As a result, randomized smoothing can only certify robustness against point modification attacks, leaving certified defenses against the other three types of attacks untouched.

We note that Jia et al. [9] analyzed the intrinsic certified robustness of bagging against data poisoning attacks. Both Jia et al. and our work use random sampling. However, our work has several key differences with Jia et al.. First, we solve a different problem from Jia et al.. More specifically, we aim to derive the certified robustness guarantees of 3D point cloud classification against adversarial attacks, while they aim to defend against data poisoning attacks. Second, we use sampling *without* replacement while Jia et al. adopted sampling *with* replacement, which results in significant technical differences in the derivation of the certified robustness guarantees (please refer to Supplemental Material for details). Third, we only need to train a single 3D point cloud classifier while Jia et al. requires to train multiple base classifiers.

3. Our PointGuard

In this section, we first describe how to build our PointGuard from an arbitrary 3D point cloud classifier. Then, we derive the certified robustness guarantee of our PointGuard and show its tightness. Finally, we develop an algorithm to compute our certified robustness guarantee in practice.

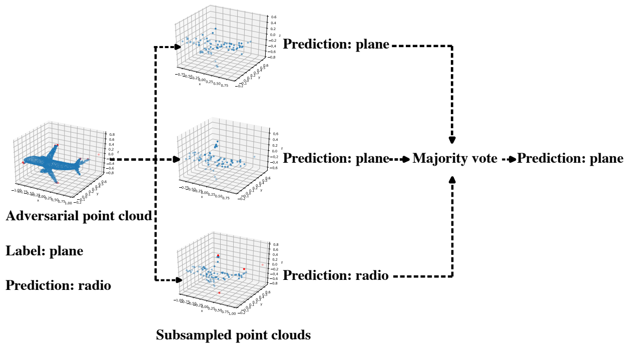


Figure 2: An example to illustrate the robustness of PointGuard. A point cloud classifier misclassifies the adversarial point cloud, where the red points are adversarially perturbed points. Three subsampled point clouds are created, and two of them (top and middle ones) do not contain adversarially perturbed points. Our PointGuard predicts the correct label for the adversarial point cloud after majority vote.

3.1. Building our PointGuard

Recall that an attacker's goal is to modify, add, and/or delete a small number of points in a 3D point cloud T such that it is misclassified by a point cloud classifier. Suppose we create multiple *subsampled point clouds* from T, each of which includes k points subsampled from T uniformly at random without replacement. Our intuition is that, when the number of adversarially modified/added/deleted points is bounded, the majority of the subsampled point clouds do not include any adversarially modified/added/deleted points and thus the majority vote among their labels predicted by the point cloud classifier may still correctly predict the label of the original point cloud T. Figure 2 provides an example to illustrate the intuition.

We design our PointGuard based on such majority-vote intuition. Next, we present a probabilistic view of the majority-vote intuition, which enables us to derive the certified robustness guarantees of PointGuard. Formally, we use $S_k(T)$ to denote a random subsampled point cloud with k points from a point cloud T. Given an arbitrary point cloud classifier f, we use it to predict the label of the subsampled point cloud $S_k(T)$. Since the subsampled point cloud $S_k(T)$ is random, the predicted label $f(S_k(T))$ is also random. We use $p_i = \Pr(f(S_k(T)) = i)$, which we call *label probability*, to denote the probability that the predicted label is i, where $i \in \{1, 2, \cdots, c\}$. Our PointGuard predicts the label with the largest label probability for the point cloud T. For simplicity, we use g to denote our PointGuard. Then, we have the following:

$$g(T) = \underset{i \in \{1,2,\cdots,c\}}{\operatorname{argmax}} \ p_i. \tag{1}$$

3.2. Deriving the Certified Perturbation Size

Certified perturbation size: In an adversarial attack, an attacker perturbs, i.e., modifies, adds, and/or deletes, some points in a point cloud T. We use T^* to denote the perturbed point cloud. Given a point cloud T and its perturbed version T^*, we define the *perturbation size* $\eta(T, T^*) = \max\{|T|, |T^*|\} - |T \cap T^*|$, where $|\cdot|$ denotes the number of points in a point cloud. Intuitively, given T and T^*, the perturbation size $\eta(T, T^*)$ indicates the minimum number of modified, added, and/or deleted points that are required to turn T into T^*. Given the point cloud T and an arbitrary positive integer r, we define the following set:

$$\Gamma(T, r) = \{T^* \mid \eta(T, T^*) \leq r\}. \tag{2}$$

Intuitively, $\Gamma(T, r)$ denotes the set of perturbed point clouds that can be obtained by perturbing at most r points in T.

Our goal is to find a maximum r^* such that our PointGuard provably predicts the same label for $\forall \ T^* \in \Gamma(T, r^*)$. Formally, we have:

$$r^* = \underset{r}{\operatorname{argmax}} \, r \ s.t. \ g(T) = g(T^*), \ \forall T^* \in \Gamma(T, r). \tag{3}$$

We call r^* *certified perturbation size*. Note that the certified perturbation size may be different for different point clouds.

Overview of our derivation: Next, we provide an overview of our proof to derive the certified perturbation size of our PointGuard for a point cloud T. The detailed proof is shown in the Supplemental Material. Our derivation is inspired by previous work [10, 9]. In particular, the key idea is to divide the space into different regions based on the Neyman-Pearson Lemma [21]. However, due to the difference in sampling methods, our space divisions are significantly different from previous work [9]. For simplicity, we define random variables $\mathbf{W} = S_k(T)$ and $\mathbf{Z} = S_k(T^*)$, which represent the random subsampled point clouds from T and T^*, respectively. Given these two random variables, we denote $p_i = \Pr(f(\mathbf{W}) = i)$ and $p_i^* = \Pr(f(\mathbf{Z}) = i)$, where $i \in \{1, 2, \cdots, c\}$. Moreover, we denote $y = g(T) = \operatorname{argmax}_{i \in \{1,2,\cdots,c\}} p_i$. Our goal is to find the maximum r^* such that $y = g(T^*) = \operatorname{argmax}_{i \in \{1,2,\cdots,c\}} p_i^*$ (i.e., $p_y^* > \max_{i \neq y} p_i^*$) for $\forall T^* \in \Gamma(T, r^*)$.

The major challenge in deriving the certified perturbation size r^* is to compute p_i^*. Specifically, the challenge stems from the complexity of the point cloud classifier f and predicting labels for the $\binom{t}{k}$ subsampled point clouds $S_k(T^*)$, where t is the number of points in T^*. To overcome the challenge, we propose to derive a lower bound of p_y^* and an upper bound of $\max_{i \neq y} p_i^*$. Moreover, based on the Neyman-Pearson Lemma [21], we derive the lower/upper bounds as the probabilities that the random variable $\mathbf{Z}$ is in certain regions of its domain space, which can be efficiently computed for any given r. Then, we find the certified perturbation size r^* as the maximum r such that the lower bound of p_y^* is larger than the upper bound of $\max_{i\neq y} p_i^*$.

Next, we discuss how we derive the lower/upper bounds. Suppose we have a lower bound $\underline{p_y}$ of p_y and an upper bound $\overline{p}_e$ of the second largest label probability p_e for the original point cloud T. Formally, we have the following:

$$p_y \geq \underline{p_y} \geq \overline{p}_e \geq p_e = \max_{i\neq y} p_i, \tag{4}$$

where $\underline{p}$ and $\overline{p}$ denote the lower and upper bounds of p, respectively. Note that $\underline{p_y}$ and $\overline{p}_e$ can be estimated using the unperturbed point cloud T. In Section 3.3, we propose an algorithm to estimate them. Based on the fact that p_y and p_i $(i \neq y)$ should be integer multiples of $1/\binom{n}{k}$, where n is the number of points in T, we have the following:

$$\underline{p_y'} \triangleq \frac{\lceil \underline{p_y} \cdot \binom{n}{k} \rceil}{\binom{n}{k}} \leq \Pr(f(\mathbf{W}) = y), \tag{5}$$

$$\overline{p}_i' \triangleq \frac{\lfloor \overline{p}_i \cdot \binom{n}{k} \rfloor}{\binom{n}{k}} \geq \Pr(f(\mathbf{W}) = i), \forall i \neq y. \tag{6}$$

Given these probability bounds $\underline{p_y'}$ and $\overline{p}_i'$ $(i \neq y)$, we derive a lower bound of p_y^* and an upper bound of p_i^* $(i \neq y)$ via the Neyman-Pearson Lemma [21]. We use Φ to denote the joint space of $\mathbf{W}$ and $\mathbf{Z}$, where each element in the space is a 3D point cloud with k points subsampled from T or T^*. We denote by E the set of intersection points between T and T^*, i.e., $E = T \cap T^*$. Then, we can divide Φ into three disjoint regions: Δ_T, Δ_E, and Δ_{T^*}. In particular, Δ_E consists of the subsampled point clouds that can be obtained by subsampling k points from E; and Δ_T (or Δ_{T^*}) consists of the subsampled point clouds that are subsampled from T (or T^*) but do not belong to Δ_E.

We assume $\underline{p_y'} > \Pr(\mathbf{W} \in \Delta_T)$. Note that we can make this assumption because our goal is to find a sufficient condition. Then, we can find a region $\Delta_y \subseteq \Delta_E$ such that $\Pr(\mathbf{W} \in \Delta_y) = \underline{p_y'} - \Pr(\mathbf{W} \in \Delta_T)$. We can find the region because $\underline{p_y'} - \Pr(\mathbf{W} \in \Delta_T)$ is an integer multiple of $1/\binom{n}{k}$. Similarly, we can assume $\overline{p}_i' < \Pr(\mathbf{W} \in \Delta_E)$ since our goal is to find a sufficient condition. Then, for each $i \neq y$, we can find a region $\Delta_i \subseteq \Delta_E$ such that $\Pr(\mathbf{W} \in \Delta_i) = \overline{p}_i'$ based on the fact that $\overline{p}_i'$ is an integer multiple of $1/\binom{n}{k}$. Finally, we derive the following bounds based on the Neyman-Pearson Lemma [21]:

$$p_y^* \geq \underline{p_y^*} = \Pr(\mathbf{Z} \in \Delta_y), \tag{7}$$

$$p_i^* \leq \overline{p}_i^* = \Pr(\mathbf{Z} \in \Delta_i \cup \Delta_{T^*}), \forall i \neq y, \tag{8}$$

where $\Pr(\mathbf{Z} \in \Delta_y)$ and $\Pr(\mathbf{Z} \in \Delta_i \cup \Delta_{T^*})$ represent the probabilities that $\mathbf{Z}$ is in the corresponding regions, which can be efficiently computed via the probability mass function of $\mathbf{Z}$. Then, our certified perturbation size r^* is the maximum r such that $\underline{p_y^*} > \max_{i\neq y} \overline{p}_i^*$.

Formally, we have the following theorem:

Theorem 1 (Certified Perturbation Size)**.** *Suppose we have an arbitrary point cloud classifier f, a 3D point cloud T, and a subsampling size k. y, e, $\underline{p_y} \in [0,1]$, and $\overline{p}_e \in [0,1]$ satisfy Equation (4). Then, our PointGuard g guarantees that $g(T^*) = y$, $\forall T^* \in \Gamma(T, r^*)$, where r^* is the solution to the following optimization problem:*

$$r^* = \operatorname*{argmax}_r r$$

$$s.t. \max_{n-r\leq t\leq n+r} \frac{\binom{t}{k}}{\binom{n}{k}} - 2 \cdot \frac{\binom{\max(n,t)-r}{k}}{\binom{n}{k}} + 1 - \underline{p_y'} + \overline{p}_e' < 0, \tag{9}$$

where $\underline{p_y'}$ and $\overline{p}_e'$ are respectively defined in Equation (5) and (6), n is the number of points in T, and t is the number of points in T^* which ranges from $n-r$ to $n+r$ when the perturbation size is r.

Proof. See Section A in Supplementary Material. □

Note that our Theorem 1 can be applied to any of the four types of adversarial attacks to point cloud classification, i.e., point perturbation, modification, addition, and deletion attacks. Moreover, for point modification, addition, and deletion attacks, we can further simplify the constraint in Equation (9) as there is a simple relationship between t, n, and r. Specifically, we have the following corollaries.

Corollary 1 (Point Modification Attacks)**.** *Suppose an attacker only modifies existing points in a 3D point cloud, i.e., we have $t = n$. Then, the constraint in Equation (9) reduces to $1 - \frac{\binom{n-r}{k}}{\binom{n}{k}} - \frac{\underline{p_y'} - \overline{p}_e'}{2} < 0$.*

Corollary 2 (Point Addition Attacks)**.** *Suppose an attacker only adds new points to a 3D point cloud, i.e., we have $t = n + r$. Then, the constraint in Equation (9) reduces to $\frac{\binom{n+r}{k}}{\binom{n}{k}} - 1 - \underline{p_y'} + \overline{p}_e' < 0$.*

Corollary 3 (Point Deletion Attacks)**.** *Suppose an attacker only deletes existing points from a 3D point cloud, i.e., we have $t = n-r$. Then, the constraint in Equation (9) reduces to $-\frac{\binom{n-r}{k}}{\binom{n}{k}} + 1 - \underline{p_y'} + \overline{p}_e' < 0$.*

Next, we show that our derived certified perturbation size is tight, i.e., it is impossible to derive a certified perturbation size larger than ours if no assumptions on the point cloud classifier f are made.

Theorem 2 (Tightness of certified perturbation size)**.** *Suppose we have $\underline{p_y'} + \overline{p}_e' \leq 1$ and $\underline{p_y'} + \sum_{i\neq y} \overline{p}_i' \geq 1$. Then, for $\forall r > r^*$, there exist a point cloud classifier f^* which satisfies Equation (4) and an adversarial point cloud T^* such that $g(T^*) \neq y$ or there exist ties.*

Proof. See Section B in Supplementary Material. □

```
Algorithm 1: PREDICTION & CERTIFICATION
    Input: $f$, $T$, $k$, $N$, and $\alpha$.
    Output: Predicted label and certified perturbation size.
    $M_1, M_2, \cdots, M_N \leftarrow$ RANDOMSUBSAMPLE$(T, k)$
    counts$[i] \leftarrow \sum_{j=1}^{N} \mathbb{I}(f(M_j) = i), i = 1, 2, \cdots, c$.
    $y, e \leftarrow$ top two indices in counts
    $\underline{p_y}, \overline{p}_e \leftarrow$ PROBBOUNDESTIMATION(counts, $\alpha$)
    if $\underline{p_y} > \overline{p}_e$ then
        $r^* =$ BINARYSEARCH$(|T|, k, \underline{p_y}, \overline{p}_e)$
    else
        $y, r^* \leftarrow$ ABSTAIN, ABSTAIN
    end if
    return $y, r^*$
```

3.3. Computing the Certified Perturbation Size

Given an arbitrary point cloud classifier f and a 3D point cloud T, computing the certified perturbation size r^* requires solving the optimization problem in Equation (9), which involves the label probability lower bound $\underline{p_y}$ and upper bound $\overline{p}_e$. We develop a Monte-Carlo algorithm to estimate these label probability bounds, with which we solve the optimization problem efficiently via binary search.

Estimating label probability lower and upper bounds: We first create N random subsampled point clouds from T, each of which contains k points. For simplicity, we use $M_1, M_2, \cdots, M_N$ to denote them. Then, we use the point cloud classifier f to predict a label for each subsampled point cloud M_j, where $j = 1, 2, \cdots, N$. We use N_i to denote the number of subsampled point clouds whose predicted labels are i, i.e., $N_i = \sum_{j=1}^{N} \mathbb{I}(f(M_j) = i)$, where $\mathbb{I}$ is an indicator function. We predict the label with the largest N_i as the label of the original point cloud T, i.e., $y = g(T) = \text{argmax}_{i \in \{1,2,\cdots,c\}} N_i$. Moreover, based on the definition of the label probability p_i, we know N_i follows a binomial distribution with parameters N and p_i, i.e., $N_i \sim \text{Binomial}(N, p_i)$. Therefore, we can use SimuEM [10], which is based on Clopper-Pearson [3] method, to estimate a lower or upper bound of p_i using N_i and N. In particular, we have:

$$\underline{p_y} = \text{Beta}(\frac{\alpha}{c}; N_y, N - N_y + 1), \tag{10}$$

$$\overline{p}_i = \text{Beta}(1 - \frac{\alpha}{c}; N_i, N - N_i + 1), \forall i \neq y, \tag{11}$$

where $1 - \alpha$ is the confidence level for simultaneously estimating the c label probability bounds and $\text{Beta}(\tau; \mu, \nu)$ is the τth quantile of the Beta distribution with shape parameters μ and ν. Both $\max_{i \neq y} \overline{p}_i$ and $1 - \underline{p_y}$ are upper bounds of $\overline{p}_e$. We use the smaller one as $\overline{p}_e$, i.e., we have $\overline{p}_e = \min(\max_{i \neq y} \overline{p}_i, 1 - \underline{p_y})$, which gives a tighter $\overline{p}_e$.

Solving the optimization problem: Given the estimated label probability bounds, we can use binary search to solve the optimization problem in Equation (9) to find the certified perturbation size r^*.

Complete algorithm: Algorithm 1 shows the complete process of our prediction and certification for a 3D point cloud T, which outputs our PointGuard's predicted label y and certified perturbation size r^* for T. The function RANDOMSUBSAMPLE creates N subsampled point clouds from T. The function PROBBOUNDESTIMATION estimates the label probability lower and upper bounds $\underline{p_y}$ and $\overline{p}_e$ with confidence level $1-\alpha$ based on Equation (10) and (11). The function BINARYSEARCH solves the optimization problem in Equation (9) to obtain r^* using binary search.

3.4. Training the Classifier with Subsampling

Our PointGuard is built upon a point cloud classifier f. In particular, in the standard training process, f is trained on the original point clouds. Our PointGuard uses f to predict the labels for the subsampled point clouds. Since the subsampled point clouds have a different distribution from the original point clouds, f has a low accuracy on the subsampled point clouds. As a result, PointGuard has suboptimal robustness. To address the issue, we propose to train the point cloud classifier f on subsampled point clouds instead of the original point clouds. Specifically, given a batch of point clouds from the training dataset, we first create a subsampled point cloud for each point cloud in the batch, and then we use the batch of subsampled point clouds to update f. Our experimental results show that training with subsampling significantly improves the robustness of PointGuard.

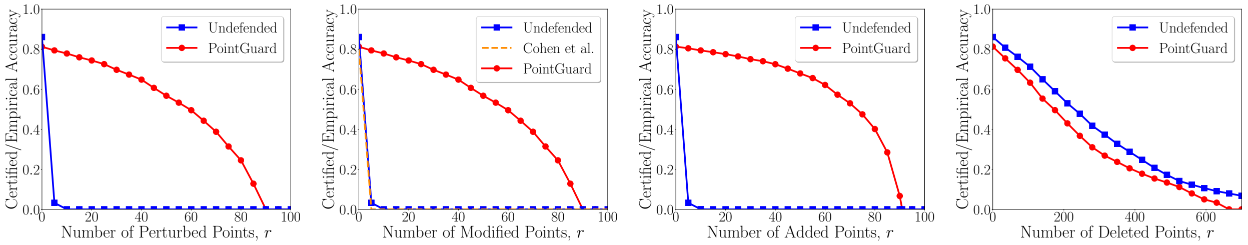
4. Experiments

4.1. Experimental Setup

Datasets and models: We evaluate PointGuard on ModelNet40 [33] and ScanNet [5] datasets. In particular, the ModelNet40 dataset contains 12,311 3D CAD models, each of which is a point cloud comprising 2,048 three dimensional points and belongs to one of the 40 common object categories. The dataset is splitted into 9,843 training point clouds and 2,468 testing point clouds. ScanNet is an RGB-D video dataset which contains 2.5M views from 1,513 scenes. Following Li et al. [17], we extract 6,263 training point clouds and 2,061 testing point clouds from ScanNet, each of which belongs to one of the 16 object categories and has 2,048 six dimensional points. We adopt PointNet [23] and DGCNN [30] as the point cloud classifiers for ModelNet40 and ScanNet, respectively. We use publicly available implementations for both PointNet[1] and DGCNN[2].

[1] https://github.com/charlesq34/pointnet

[2] https://github.com/WangYueFt/dgcnn



(a) Point perturbation attacks.

(b) Point modification attacks.

(c) Point addition attacks.

(d) Point deletion attacks.

Figure 3: Comparing different methods under different attacks on ModelNet40. The results on ScanNet are in Supplemental Material.

Compared methods: We compare our PointGuard with the following methods:

Undefended classifier. We call the standard point cloud classifier *undefended classifier*, e.g., the undefended classifiers are PointNet and DGCNN on the two datasets in our experiments, respectively.

Randomized smoothing (Cohen et al.) [4]. Randomized smoothing adds isotropic Gaussian noise with mean 0 and standard deviation σ to an image before using a classifier to classify it. Randomized smoothing provably predicts the same label for the image when the ℓ_2-norm of the adversarial perturbation added to the image is less than a threshold, which is called *certified radius*. We can generalize randomized smoothing to certify robustness against point modification attacks. In particular, we can add Gaussian noise to each dimension of each point in a point cloud before using a point cloud classifier to classify it, and randomized smoothing provably predicts the same label for the point cloud when the ℓ_2-norm of the adversarial perturbation is less than the certified radius. Note that our certified perturbation size is the number points that are perturbed by an attacker. Therefore, we transform the certified radius to certified perturbation size as follows. Suppose the points in the point clouds lie in the space Θ, and the ℓ_2-norm distance between two arbitrary points in the space Θ is no larger than λ, i.e., $\max_{\theta_1,\theta_2} \|\theta_1 - \theta_2\|_2 \leq \lambda$. For instance, $\lambda = 2\sqrt{3}$ for ModelNet40 and $\lambda = \sqrt{15}$ for ScanNet. Then, we can employ the relationship between ℓ_0-norm and ℓ_2-norm to derive the certified perturbation size based on the certified radius returned by randomized smoothing. Specifically, given λ and the ℓ_2-norm certified radius δ for a point cloud, the certified perturbation size can be computed as $\lfloor \frac{\delta^2}{\lambda^2} \rfloor$.

Evaluation metric: PointGuard and randomized smoothing provide certified robustness guarantees, while the undefended classifiers provide empirical robustness. Therefore, we use *certified accuracy*, which has been widely used to measure the certified robustness of a machine learning classifier against adversarial perturbations, to evaluate PointGuard and randomized smoothing; and we use *empirical accuracy* under an emprical attack to evaluate the undefended classifiers. In particular, the certified accuracy at r perturbed points is the fraction of the testing point clouds whose labels are correctly predicted and whose certified perturbation sizes are no smaller than r. Formally, given a testing set of point clouds $\mathcal{T} = \{(T_o, l_o)\}_{o=1}^{m}$, the certified accuracy at r perturbed points is defined as follows:

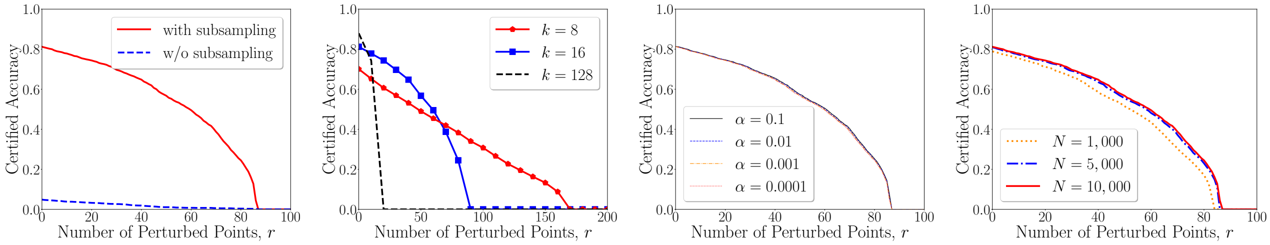
$$CA_r = \frac{\sum_{o=1}^{m} \mathbb{I}\left(l_o = y_o\right) \cdot \mathbb{I}\left(r_o^* \geq r\right)}{m}, \qquad (12)$$

where $\mathbb{I}$ is an indicator function, l_o is the true label of the testing point cloud T_o, and y_o and r_o^* respectively are the predicted label and the certified perturbation size returned by PointGuard or randomized smoothing for T_o.

We use an empirical attack to calculate the empirical accuracy of an undefended classifier. Specifically, we first use the empirical attack to perturb each point cloud in a testing set, and then we use the undefended classifier to classify the perturbed point clouds and compute the accuracy (called empirical accuracy). We adopt the empirical attacks proposed by Xiang et al. [34] for the point perturbation, modification, and addition attacks. We use the empirical attack proposed by Wicker et al. [31] for the point deletion attacks.

We note that the certified accuracy at r is a *lower bound* of the accuracy that PointGuard or randomized smoothing can achieve no matter how an attacker modifies, adds, and/or deletes at most r points for each point cloud in the testing set, while the empirical accuracy under an empirical attack is an *upper bound* of the accuracy that an undefended classifier can achieve under attacks.

Parameter setting: Our PointGuard has three parameters: k, $1-\alpha$, and N. Unless otherwise mentioned, we adopt the following default parameters: $\alpha = 0.0001$, $N = 10,000$, and $k = 16$ for both ModelNet40 and ScanNet. By default, we train the point cloud classifiers with subsampling. We adopt $\sigma = 0.5$ for randomized smoothing so its certified accuracy without attack is similar to that of PointGuard. By default, we consider the point perturbation attacks, as they are the strongest among the four types of attacks.



(a) Training with vs w/o subsampling

(b) Impact of k

(c) Impact of α

(d) Impact of N

Figure 4: (a) Training the point cloud classifier with vs. without subsampling. (b), (c), and (d) show the impact of k, α, and N, respectively. The dataset is ModelNet40. The results on ScanNet are in Supplemental Material.

4.2. Experimental Results

Comparing PointGuard with other methods under different attacks: Figure 3 (or Figure 5 in Supplemental Material) compares PointGuard with other methods under the four types of attacks on ModelNet40 (or ScanNet), where an undefended classifier is measured by its empirical accuracy, while PointGuard and randomized smoothing are measured by certified accuracy. We observe that the certified accuracy of PointGuard is slightly lower than the empirical accuracy of an undefended classifier when there are no attacks, i.e., $r = 0$. However, for point perturbation, modification, and addition attacks, the empirical accuracy of an undefended classifier quickly drops to 0 while the certified accuracy of PointGuard is still high as r increases. For point deletion attacks, the empirical accuracy of an undefended classifier may be higher than the certified accuracy of PointGuard. This indicates that the existing empirical point deletion attacks are not strong enough.

Our PointGuard substantially outperforms randomized smoothing for point modification attacks in terms of certified accuracy. Randomized smoothing adds additive noise to a point cloud, while our PointGuard subsamples a point cloud. Our experimental results show that subsampling outperforms additive noise to build provably robust point cloud classification systems. We also observe that the empirical accuracy of the undefended classifier is close to the certified accuracy of randomized smoothing, indicating that the empirical point modification attacks are strong.

We also compare PointGuard with an empirical defense (i.e., DUP-Net [40]) to measure the gaps between certified accuracy and empirical accuracy. According to [40], on ModelNet40, the empirical accuracy of DUP-Net under a point deletion attack [37] is 76.1%, 67.7%, and 57.7% when the attacker deletes 50, 100, and 150 points, respectively. Under the same setting, PointGuard achieves certified accuracy of 73.4%, 64.3%, and 53.5%, respectively. We observe the gaps between the empirical accuracy (an upper bound of accuracy) of DUP-Net and certified accuracy (a lower bound of accuracy) of PointGuard are small.

Training with vs. without subsampling: Figure 4a (or Figure 6a in Supplemental Material) shows the comparison of the certified accuracy of PointGuard when the point cloud classifier is trained with or without subsampling on ModelNet40 (or ScanNet). Our experimental results demonstrate that training with subsampling can substantially improve the certified accuracy of PointGuard. The reason is that the point cloud classifier trained with subsampling can more accurately classify the subsampled point clouds.

Impact of k, α, and N: Figure 4b, 4c, and 4d (or Figure 6b, 6c, and 6d in Supplemental Material) show the impact of k, α, and N on the certified accuracy of our PointGuard on ModelNet40 (or ScanNet), respectively. Based on the experimental results, we make the following observations. First, k measures a tradeoff between accuracy without attacks (i.e., $r = 0$) and robustness. In particular, a smaller k gives a smaller certified accuracy without attacks, but the certified accuracy drops to 0 more slowly as the number of perturbed points r increases. The reason is that the perturbed points are less likely to be subsampled when k is smaller. Second, the certified accuracy increases as α or N increases. The reason is that a larger α or N leads to tighter lower and upper label probability bounds, which in turn lead to larger certified perturbation sizes. We also note that the certified accuracy is insensitive to α and N once they are large enough.

5. Conclusion

In this work, we propose PointGuard, the first provably robust 3D point cloud classification system against various adversarial attacks. We show that PointGuard provably predicts the same label for a testing 3D point cloud when the number of adversarially perturbed points is bounded. Moreover, we prove our bound is tight. We empirically demonstrate the effectiveness of PointGuard on ModelNet40 and ScanNet benchmark datasets. An interesting future work is to further improve PointGuard by leveraging the knowledge of the point cloud classifier.

References

[1] Xiaoyu Cao and Neil Zhenqiang Gong. Mitigating evasion attacks to deep neural networks via region-based classification. In *Proceedings of the 33rd Annual Computer Security Applications Conference*, pages 278–287, 2017. 3

[2] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017. 1

[3] Charles J Clopper and Egon S Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934. 6

[4] Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. *arXiv preprint arXiv:1902.02918*, 2019. 2, 3, 7, 13

[5] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. 6

[6] Xiaoyi Dong, Dongdong Chen, Hang Zhou, Gang Hua, Weiming Zhang, and Nenghai Yu. Self-robust 3d point recognition via gather-vector guidance. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11513–11521. IEEE, 2020. 1, 3

[7] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 3

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

[9] Jinyuan Jia, Xiaoyu Cao, and Neil Zhenqiang Gong. Intrinsic certified robustness of bagging against data poisoning attacks. In *AAAI*, 2020. 3, 4, 11, 13

[10] Jinyuan Jia, Xiaoyu Cao, Binghui Wang, and Neil Zhenqiang Gong. Certified robustness for top-k predictions against adversarial perturbations via randomized smoothing. In *International Conference on Learning Representations*, 2019. 4, 6, 11, 13

[11] Jinyuan Jia, Binghui Wang, Xiaoyu Cao, and Neil Zhenqiang Gong. Certified robustness of community detection against adversarial structural perturbation via randomized smoothing. In *Proceedings of The Web Conference 2020*, pages 2718–2724, 2020. 3

[12] Jaeyeon Kim, Binh-Son Hua, Duc Thanh Nguyen, and Sai-Kit Yeung. Minimal adversarial examples for deep learning on 3d point clouds. *arXiv preprint arXiv:2008.12066*, 2020. 3

[13] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 656–672. IEEE, 2019. 3

[14] Guang-He Lee, Yang Yuan, Shiyu Chang, and Tommi Jaakkola. Tight certificates of adversarial robustness for randomly smoothed classifiers. In *Advances in Neural Information Processing Systems*, pages 4910–4921, 2019. 3

[15] Alexander Levine and Soheil Feizi. Robustness certificates for sparse adversarial attacks by randomized ablation. In *AAAI*, pages 4585–4593, 2020. 3

[16] Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Certified adversarial robustness with additive noise. In *Advances in Neural Information Processing Systems*, pages 9464–9474, 2019. 3

[17] 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*, pages 820–830, 2018. 1, 3, 6

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

[19] Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 369–385, 2018. 3

[20] Chengcheng Ma, Weiliang Meng, Baoyuan Wu, Shibiao Xu, and Xiaopeng Zhang. Efficient joint gradient based attack against sor defense for 3d point cloud classification. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1819–1827, 2020. 1, 2, 3

[21] Jerzy Neyman and Egon Sharpe Pearson. Ix. on the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337, 1933. 2, 4, 5, 11

[22] Rafael Pinot, Laurent Meunier, Alexandre Araujo, Hisashi Kashima, Florian Yger, Cédric Gouy-Pailler, and Jamal Atif. Theoretical evidence for adversarial robustness through randomization. In *Advances in Neural Information Processing Systems*, pages 11838–11848, 2019. 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*, pages 652–660, 2017. 1, 2, 3, 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*, pages 5099–5108, 2017. 1, 3

[25] Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. In *Advances in Neural Information Processing Systems*, pages 11292–11303, 2019. 3

[26] Jiachen Sun, Karl Koenig, Yulong Cao, Qi Alfred Chen, and Zhuoqing Mao. On the adversarial robustness of 3d point cloud classification. 2021. 3

[27] Tzungyu Tsai, Kaichen Yang, Tsung-Yi Ho, and Yier Jin. Robust adversarial objects against deep learning models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 954–962, 2020. 3

[28] Jacob Varley, Chad DeChant, Adam Richardson, Joaquín Ruales, and Peter Allen. Shape completion enabled robotic grasping. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 2442–2447. IEEE, 2017. 1

[29] Binghui Wang, Xiaoyu Cao, Jinyuan Jia, Neil Zhenqiang Gong, et al. On certifying robustness against backdoor attacks via randomized smoothing. In *CVPR 2020 Workshop on Adversarial Machine Learning in Computer Vision*, 2020. 3

[30] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 1, 2, 3, 6

[31] Matthew Wicker and Marta Kwiatkowska. Robustness of 3d deep learning in an adversarial setting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11767–11775, 2019. 1, 3, 7

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

[33] 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*, pages 1912–1920, 2015. 6

[34] Chong Xiang, Charles R Qi, and Bo Li. Generating 3d adversarial point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9136–9144, 2019. 1, 3, 7

[35] Jiancheng Yang, Qiang Zhang, Rongyao Fang, Bingbing Ni, Jinxian Liu, and Qi Tian. Adversarial attack and defense on point sets. *arXiv preprint arXiv:1902.10899*, 2019. 1, 3

[36] Xiangyu Yue, Bichen Wu, Sanjit A Seshia, Kurt Keutzer, and Alberto L Sangiovanni-Vincentelli. A lidar point cloud generator: from a virtual world to autonomous driving. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pages 458–464, 2018. 1

[37] Yue Zhao, Yuwei Wu, Caihua Chen, and Andrew Lim. 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*, pages 1201–1210, 2020. 3, 8

[38] Tianhang Zheng, Changyou Chen, Junsong Yuan, Bo Li, and Kui Ren. Pointcloud saliency maps. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1598–1606, 2019. 1, 3

[39] Hang Zhou, Dongdong Chen, Jing Liao, Kejiang Chen, Xiaoyi Dong, Kunlin Liu, Weiming Zhang, Gang Hua, and Nenghai Yu. Lg-gan: Label guided adversarial network for flexible targeted attack of point cloud based deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10356–10365, 2020. 1, 3

[40] 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 International Conference on Computer Vision*, pages 1961–1970, 2019. 1, 2, 3, 8

A. Proof of Theorem 1

Given a point cloud T and its perturbed version T^*, we define the following two random variables:

$$\mathbf{W} = S_k(T),\ \mathbf{Z} = S_k(T^*), \tag{13}$$

where $\mathbf{W}$ and $\mathbf{Z}$ represent the random 3D point clouds with k points subsampled from T and T^* uniformly at random without replacement, respectively. We use Φ to denote the joint space of $\mathbf{W}$ and $\mathbf{Z}$, where each element is a 3D point cloud with k points subsampled from T or T^*. We denote by E the set of intersection points between T and T^*, i.e., $E = T \cap T^*$.

Before proving our theorem, we first describe a variant of the Neyman-Pearson Lemma [21] that will be used in our proof. The variant is from [9].

Lemma 1 (Neyman-Pearson Lemma)**.** *Suppose* $\mathbf{W}$ *and* $\mathbf{Z}$ *are two random variables in the space* Φ *with probability distributions* σ_w *and* σ_z*, respectively. Let* $H : \Phi \rightarrow \{0,1\}$ *be a random or deterministic function. Then, we have the following:*

- *If* $Q_1 = \{\varphi \in \Phi : \sigma_w(\varphi) > \zeta \cdot \sigma_z(\varphi)\}$ *and* $Q_2 = \{\varphi \in \Phi : \sigma_w(\varphi) = \zeta \cdot \sigma_z(\varphi)\}$ *for some* $\zeta > 0$*. Let* $Q = Q_1 \cup Q_3$*, where* $Q_3 \subseteq Q_2$*. If we have* $Pr(H(\mathbf{W}) = 1) \geq Pr(\mathbf{W} \in Q)$*, then* $Pr(H(\mathbf{Z}) = 1) \geq Pr(\mathbf{Z} \in Q)$*.*

- *If* $Q_1 = \{\varphi \in \Phi : \sigma_w(\varphi) < \zeta \cdot \sigma_z(\varphi)\}$ *and* $Q_2 = \{\varphi \in \Phi : \sigma_w(\varphi) = \zeta \cdot \sigma_z(\varphi)\}$ *for some* $\zeta > 0$*. Let* $Q = Q_1 \cup Q_3$*, where* $Q_3 \subseteq Q_2$*. If we have* $Pr(H(\mathbf{W}) = 1) \leq Pr(\mathbf{W} \in Q)$*, then* $Pr(H(\mathbf{Z}) = 1) \leq Pr(\mathbf{Z} \in Q)$*.*

Proof. Please refer to [9]. □

Next, we formally prove our Theorem 1. Our proof is inspired by previous work [10, 9]. Roughly speaking, the idea is to derive the label probability lower and upper bounds via computing the probability of random variables in certain regions crafted by the variant of Neyman-Pearson Lemma. However, due to the difference in sampling methods, our space divisions are significantly different from previous work [9]. Recall that we denote $p_i = \Pr(f(\mathbf{W}) = i)$ and $p_i^* = \Pr(f(\mathbf{Z}) = i)$, where $i \in \{1, 2, \cdots, c\}$. We denote $y = \operatorname{argmax}_{i=\{1,2,\ldots,c\}} p_i$. Our goal is to find the maximum r^* such that $y = \operatorname{argmax}_{i=\{1,2,\cdots,c\}} p_i^*$, i.e., $p_y^* > p_e^* = max_{i \neq y} p_i^*$, for $\forall T^* \in \Gamma(T, r^*)$. Our key step is to derive a lower bound of p_y^* and an upper bound of $p_e^* = max_{i \neq y} p_i^*$ via Lemma 1. Given these probability bounds, we can find the maximum r^* such that the lower bound of p_y^* is larger than the upper bound of p_e^*.

Dividing the space Φ: We first divide the space Φ into three regions which are as follows:

$$\Delta_T = \{\varphi \in \Phi | \varphi \subseteq T, \varphi \nsubseteq E\}, \tag{14}$$

$$\Delta_{T^*} = \{\varphi \in \Phi | \varphi \subseteq T^*, \varphi \nsubseteq E\}, \tag{15}$$

$$\Delta_E = \{\varphi \in \Phi | \varphi \subseteq E\}, \tag{16}$$

where Δ_E consists of the subsampled point clouds that can be obtained by subsampling k points from E; and Δ_T (or Δ_{T^*}) consists of the subsampled point clouds that are subsampled from T (or T^*) but do not belong to Δ_E. Since $\mathbf{W}$ and $\mathbf{Z}$, respectively, represent the random 3D point clouds with k points subsampled from T and T^* uniformly at random without replacement, we have the following probability mass functions:

$$\Pr(\mathbf{W} = \varphi) = \begin{cases} \frac{1}{\binom{n}{k}}, & \text{if } \varphi \in \Delta_T \cup \Delta_E, \\ 0, & \text{otherwise}, \end{cases} \tag{17}$$

$$\Pr(\mathbf{Z} = \varphi) = \begin{cases} \frac{1}{\binom{t}{k}}, & \text{if } \varphi \in \Delta_{T^*} \cup \Delta_E, \\ 0, & \text{otherwise}, \end{cases} \tag{18}$$

where t is the number of points in T^* (i.e., $t = |T^*|$). We use s to denote the number of intersection points between T and T^*, i.e., $s = |E| = |T \cap T^*|$. Then, the size of Δ_E is $\binom{s}{k}$, i.e., $|\Delta_E| = \binom{s}{k}$. Given the size of Δ_E, we have the following probabilities:

$$\Pr(\mathbf{W} \in \Delta_E) = \frac{\binom{s}{k}}{\binom{n}{k}}, \tag{19}$$

$$\Pr(\mathbf{W} \in \Delta_T) = 1 - \frac{\binom{s}{k}}{\binom{n}{k}}, \tag{20}$$

$$\Pr(\mathbf{W} \in \Delta_{T^*}) = 0. \tag{21}$$

$$\Pr(\mathbf{Z} \in \Delta_E) = \frac{\binom{s}{k}}{\binom{t}{k}}, \tag{22}$$

$$\Pr(\mathbf{Z} \in \Delta_{T^*}) = 1 - \frac{\binom{s}{k}}{\binom{t}{k}}, \tag{23}$$

$$\Pr(\mathbf{Z} \in \Delta_T) = 0. \tag{24}$$

We have $\Pr(\mathbf{W} \in \Delta_E) = \frac{\binom{s}{k}}{\binom{n}{k}}$ because $\Pr(\mathbf{W} \in \Delta_E) = \frac{|\Delta_E|}{|\Delta_T \cup \Delta_E|} = \frac{\binom{s}{k}}{\binom{n}{k}}$. Since $\Pr(\mathbf{W} \in \Delta_T) + \Pr(\mathbf{W} \in \Delta_E) = 1$, we have $\Pr(\mathbf{W} \in \Delta_T) = 1 - \frac{\binom{s}{k}}{\binom{n}{k}}$. We have $\Pr(\mathbf{W} \in \Delta_{T^*}) = 0$ because $\mathbf{W}$ is subsampled from T, which does not contain any points from $T^* \setminus E$. Similarly, we can compute the probabilities of random variable $\mathbf{Z}$ in those regions.

Based on the fact that p_y and $p_i (i \neq y)$ should be integer multiples of $1/\binom{n}{k}$, we derive the following bounds:

$$\underline{p_y'} \triangleq \frac{\lceil \underline{p_y} \cdot \binom{n}{k} \rceil}{\binom{n}{k}} \leq \Pr(f(\mathbf{W}) = y), \tag{25}$$

$$\overline{p}_i' \triangleq \frac{\lfloor \overline{p}_i \cdot \binom{n}{k} \rfloor}{\binom{n}{k}} \geq \Pr(f(\mathbf{W}) = i), \forall i \neq y. \tag{26}$$

Deriving a lower bound of p_y^*: We define a binary function $H_y(\varphi) = \mathbb{I}(f(\varphi) = y)$, where $\varphi \in \Phi$ and $\mathbb{I}$ is an indicator function. Then, we have the following based on the definitions of the random variable $\mathbf{Z}$ and the function H_y:

$$p_y^* = \Pr(f(\mathbf{Z}) = y) = \Pr(H_y(\mathbf{Z}) = 1). \tag{27}$$

Our idea is to find a region such that we can apply Lemma 1 to derive a lower bound of $\Pr(H_y(\mathbf{Z}) = 1)$. We assume $\underline{p_y'} - \left(1 - \frac{\binom{s}{k}}{\binom{n}{k}}\right) \geq 0$. We can make this assumption because we only need to find a sufficient condition. Then, we can find a region $\Delta_y \subseteq \Delta_E$ satisfying the following:

$$\begin{aligned} &\Pr(\mathbf{W} \in \Delta_y) && (28)\\ =&\underline{p_y'} - \Pr(\mathbf{W} \in \Delta_T) && (29)\\ =&\underline{p_y'} - \left(1 - \frac{\binom{s}{k}}{\binom{n}{k}}\right). && (30) \end{aligned}$$

We can find the region Δ_y because $\underline{p_y'}$ is an integer multiple of $\frac{1}{\binom{n}{k}}$. Given the region Δ_y, we define the following region:

$$\mathcal{A} = \Delta_T \cup \Delta_y. \tag{31}$$

Then, based on Equation (25), we have:

$$\Pr(f(\mathbf{W}) = y) \geq \underline{p_y'} = \Pr(\mathbf{W} \in \mathcal{A}). \tag{32}$$

We can establish the following based on the definition of $\mathbf{W}$:

$$\Pr(H_y(\mathbf{W}) = 1) = \Pr(f(\mathbf{W}) = y) \geq \Pr(\mathbf{W} \in \mathcal{A}). \tag{33}$$

Furthermore, we have $\Pr(\mathbf{W} = \varphi) > \epsilon \cdot \Pr(\mathbf{Z} = \varphi)$ if and only if $\varphi \in \Delta_T$ and $\Pr(\mathbf{W} = \varphi) = \epsilon \cdot \Pr(\mathbf{Z} = \varphi)$ if $\varphi \in \Delta_y$, where $\epsilon = \frac{\binom{t}{k}}{\binom{n}{k}}$. Therefore, based on the definition of $\mathcal{A}$ in Equation (31) and the condition in Equation (33), we obtain the following by applying Lemma 1:

$$\Pr(H_y(\mathbf{Z}) = 1) \geq \Pr(\mathbf{Z} \in \mathcal{A}). \tag{34}$$

Since we have $p_y^* = \Pr(H_y(\mathbf{Z}) = 1)$, $\Pr(\mathbf{Z} \in \mathcal{A})$ is a lower bound of p_y^* and can be computed as follows:

$$\begin{aligned} &\Pr(\mathbf{Z} \in \mathcal{A}) && (35)\\ =&\Pr(\mathbf{Z} \in \Delta_T) + \Pr(\mathbf{Z} \in \Delta_y) && (36)\\ =&\Pr(\mathbf{Z} \in \Delta_y) && (37)\\ =&\Pr(\mathbf{W} \in \Delta_y)/\epsilon && (38)\\ =&\frac{1}{\epsilon} \cdot \left(\underline{p_y'} - \left(1 - \frac{\binom{s}{k}}{\binom{n}{k}}\right)\right). && (39) \end{aligned}$$

We have Equation (37) from (36) because $\Pr(\mathbf{Z} \in \Delta_T) = 0$, Equation (38) from (37) as $\Pr(\mathbf{W} = \varphi) = \epsilon \cdot \Pr(\mathbf{Z} = \varphi)$ for $\varphi \in \Delta_y$, and the last Equation from Equation (28) - (30).

Deriving an upper bound of $\max_{i \neq y} p_i^*$: We leverage the second part of Lemma 1 to derive an upper bound of $\max_{i \neq y} p_i^*$. We assume $\Pr(\mathbf{W} \in \Delta_E) > \overline{p_i'}$, $\forall i \in \{1, 2, \cdots, c\} \setminus \{y\}$. We can make the assumption because we aim to derive a sufficient condition. For $\forall i \in \{1, 2, \cdots, c\} \setminus \{y\}$, we can find a region $\Delta_i \subseteq \Delta_E$ such that we have the following:

$$\Pr(\mathbf{W} \in \Delta_i) = \overline{p_i'}. \tag{40}$$

We can find the region because $\overline{p_i'}$ is an integer multiple of $\frac{1}{\binom{n}{k}}$. Given region Δ_i, we define the following region:

$$\mathcal{B}_i = \Delta_i \cup \Delta_{T^*}. \tag{41}$$

For $\forall i \in \{1, 2, \cdots, c\} \setminus \{y\}$, we define a function $H_i(\varphi) = \mathbb{I}(f(\varphi) = i)$, where $\varphi \in \Phi$. Then, based on Equation (26) and the definition of random variable $\mathbf{W}$, we have:

$$\Pr(H_i(\mathbf{W}) = 1) = \Pr(f(\mathbf{W}) = i) \leq \overline{p_i'} = \Pr(\mathbf{W} \in \mathcal{B}_i). \tag{42}$$

We note that $\Pr(\mathbf{W} = \varphi) < \epsilon \cdot \Pr(\mathbf{Z} = \varphi)$ if and only if $\varphi \in \Delta_{T^*}$ and $\Pr(\mathbf{W} = \varphi) = \epsilon \cdot \Pr(\mathbf{Z} = \varphi)$ if $\varphi \in \Delta_i$, where $\epsilon = \frac{\binom{t}{k}}{\binom{n}{k}}$. Based on the definition of random variable $\mathbf{Z}$, Equation (42), and Lemma 1, we have the following:

$$\Pr(H_i(\mathbf{Z}) = 1) \leq \Pr(\mathbf{Z} \in \mathcal{B}_i). \tag{43}$$

Since we have $p_i^* = \Pr(f(\mathbf{Z}) = i) = \Pr(H_i(\mathbf{Z}) = 1)$, $\Pr(\mathbf{Z} \in \mathcal{B}_i)$ is an upper bound of p_i^* and can be computed as follows:

$$\begin{aligned} &\Pr(\mathbf{Z} \in \mathcal{B}_i) && (44)\\ =&\Pr(\mathbf{Z} \in \Delta_i) + \Pr(\mathbf{Z} \in \Delta_{T^*}) && (45)\\ =&\Pr(\mathbf{Z} \in \Delta_i) + 1 - \frac{\binom{s}{k}}{\binom{t}{k}} && (46)\\ =&\Pr(\mathbf{W} \in \Delta_i)/\epsilon + 1 - \frac{\binom{s}{k}}{\binom{t}{k}} && (47)\\ =&\frac{1}{\epsilon} \cdot \overline{p_i'} + 1 - \frac{\binom{s}{k}}{\binom{t}{k}}. && (48) \end{aligned}$$

By considering all possible i in the set $\{1, 2, \cdots, c\} \setminus \{y\}$, we have:

$$\begin{aligned} &\max_{i \neq y} p_i^* && (49)\\ \leq&\max_{i \neq y} \Pr(\mathbf{Z} \in \mathcal{B}_i) && (50)\\ =&\frac{1}{\epsilon} \cdot \max_{i \neq y} \overline{p_i'} + 1 - \frac{\binom{s}{k}}{\binom{t}{k}} && (51)\\ \leq&\frac{1}{\epsilon} \cdot \overline{p_e'} + 1 - \frac{\binom{s}{k}}{\binom{t}{k}}, && (52) \end{aligned}$$

where $\overline{p}_e' \geq \max_{i \neq y} \overline{p}_i'$.

Deriving the certified perturbation size: To reach our goal $\Pr(f(\mathbf{Z}) = y) > \max_{i \neq y} \Pr(f(\mathbf{Z}) = i)$, it is sufficient to have the following:

$$\frac{1}{\epsilon} \cdot \left(\underline{p_y'} - \left(1 - \frac{\binom{s}{k}}{\binom{n}{k}} \right) \right) > \frac{1}{\epsilon} \cdot \overline{p}_e' + 1 - \frac{\binom{s}{k}}{\binom{t}{k}} \tag{53}$$

$$\iff \frac{\binom{t}{k}}{\binom{n}{k}} - 2 \cdot \frac{\binom{s}{k}}{\binom{n}{k}} + 1 - \underline{p_y'} + \overline{p}_e' < 0. \tag{54}$$

Since Equation (54) should be satisfied for all possible perturbed point cloud T^* (i.e., $n - r \leq t \leq n + r$), we have the following sufficient condition:

$$\max_{n-r \leq t \leq n+r} \frac{\binom{t}{k}}{\binom{n}{k}} - 2 \cdot \frac{\binom{s}{k}}{\binom{n}{k}} + 1 - \underline{p_y'} + \overline{p}_e' < 0. \tag{55}$$

When the above Equation (55) is satisfied, we have $\underline{p_y'} - \left(1 - \frac{\binom{s}{k}}{\binom{n}{k}} \right) \geq 0$ and $\Pr(\mathbf{W} \in \Delta_E) = \frac{\binom{s}{k}}{\binom{n}{k}} \geq \overline{p}_i', \forall i \in \{1, 2, \cdots, c\} \setminus \{y\}$, which are the conditions that we rely on to construct the region Δ_y and $\Delta_i (i \neq y)$. The certified perturbation size r^* is the maximum r that satisfies the above sufficient condition. Note that $s = \max(n, t) - r$. Then, our certified perturbation size r^* can be derived by solving the following optimization problem:

$$\begin{aligned} r^* &= \operatorname*{argmax}_r r \\ \text{s.t.} &\max_{n-r \leq t \leq n+r} \frac{\binom{t}{k}}{\binom{n}{k}} - 2 \cdot \frac{\binom{\max(n,t)-r}{k}}{\binom{n}{k}} + 1 - \underline{p_y'} + \overline{p}_e' < 0. \end{aligned} \tag{56}$$

B. Proof of Theorem 2

Similar to previous work [4, 10, 9], we show the tightness of our bounds via constructing a counterexample. In particular, when $r > r^*$, we will show that we can construct a point cloud T^* and a point cloud classifier f^* which satisfies the Equation (4) such that the label y is not predicted by our PointGuard or there exist ties. Since r^* is the maximum value that satisfies the Equation (56), there exists a point cloud T^* satisfying the following when $r > r^*$:

$$\frac{\binom{t}{k}}{\binom{n}{k}} - 2 \cdot \frac{\binom{\max(n,t)-r}{k}}{\binom{n}{k}} + 1 - \underline{p_y'} + \overline{p}_e' \geq 0 \tag{57}$$

$$\iff \frac{\binom{t}{k}}{\binom{n}{k}} - \frac{\binom{\max(n,t)-r}{k}}{\binom{n}{k}} + \overline{p}_e' \geq \underline{p_y'} - \left(1 - \frac{\binom{\max(n,t)-r}{k}}{\binom{n}{k}} \right) \tag{58}$$

$$\iff \frac{\overline{p}_e'}{\epsilon} + 1 - \frac{\binom{\max(n,t)-r}{k}}{\binom{t}{k}} \geq \frac{1}{\epsilon} \cdot \left(\underline{p_y'} - \left(1 - \frac{\binom{\max(n,t)-r}{k}}{\binom{n}{k}} \right) \right) \tag{59}$$

where t is the number of points in T^* and $\epsilon = \frac{\binom{t}{k}}{\binom{n}{k}}$. Let $\Delta_e \subseteq \Delta_E$ be the region that satisfies the following:

$$\Delta_e \cap \Delta_y = \emptyset \text{ and } \Pr(\mathbf{W} \in \Delta_e) = \overline{p}_e'. \tag{60}$$

Note that we can find the region Δ_e because we have $\underline{p_y'} + \overline{p}_e' \leq 1$. We let $\mathcal{B}_e = \Delta_{T^*} \cup \Delta_e$. Then, we can divide the the region $\Phi \setminus (\mathcal{A} \cap \mathcal{B}_e)$ into $c - 2$ regions and we use $\mathcal{B}_i$ to denote each region, where $i \in \{1, 2, \cdots, c\} \setminus \{y, e\}$. In particular, each region $\mathcal{B}_i$ satisfies $\Pr(T \in \mathcal{B}_i) \leq \overline{p}_i'$. We can find these regions since $\underline{p_y'} + \sum_{i \neq y} \overline{p}_i' \geq 1$. Then, we can construct the following point cloud classifier f^*:

$$f^*(\varphi) = \begin{cases} y, \text{ if } \varphi \in \mathcal{A}, \\ i, \text{ if } \varphi \in \mathcal{B}_i. \end{cases} \tag{61}$$

Note that the point cloud classifier f^* is well-defined in the space Φ. We have the following probabilities for our constructed point cloud classifier f^*:

$$\Pr(f^*(\mathbf{W}) = y) = \Pr(\mathbf{W} \in \mathcal{A}) = \underline{p_y'}, \tag{62}$$

$$\Pr(f^*(\mathbf{W}) = e) = \Pr(\mathbf{W} \in \mathcal{B}_e) = \overline{p}_e', \tag{63}$$

$$\Pr(f^*(\mathbf{W}) = i) = \Pr(\mathbf{W} \in \mathcal{B}_i) \leq \overline{p}_i', \tag{64}$$

where $i \in \{1, 2, \ldots, c\} \setminus \{y, e\}$. Note that the point cloud classifier f^* is consistent with Equation (4). Moreover, we have the following:

$$\Pr(f^*(\mathbf{Z}) = e) \tag{65}$$

$$= \Pr(\mathbf{Z} \in \mathcal{B}_e) \tag{66}$$

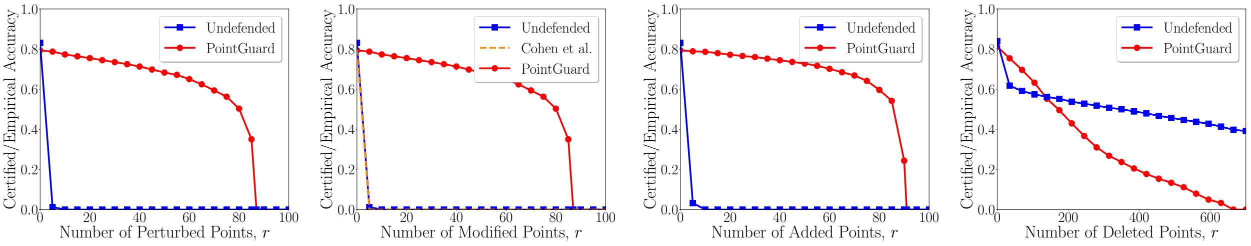
$$= \frac{\overline{p}_e'}{\epsilon} + 1 - \frac{\binom{\max(n,t)-r}{k}}{\binom{t}{k}} \tag{67}$$

$$\geq \frac{1}{\epsilon} \left(\underline{p_y'} - \left(1 - \frac{\binom{\max(n,t)-r}{k}}{\binom{n}{k}} \right) \right) \tag{68}$$

$$= \Pr(\mathbf{Z} \in \mathcal{A}) \tag{69}$$

$$= \Pr(f^*(\mathbf{Z}) = y), \tag{70}$$

where $\epsilon = \frac{\binom{t}{k}}{\binom{n}{k}}$. Note that we derive Equation (68) from (67) based on Equation (59). Therefore, for $\forall r > r^*$, there exist a point cloud classifier f^* which satisfies Equation (4) and a point cloud T^* such that $g(T^*) \neq y$ or there exist ties.



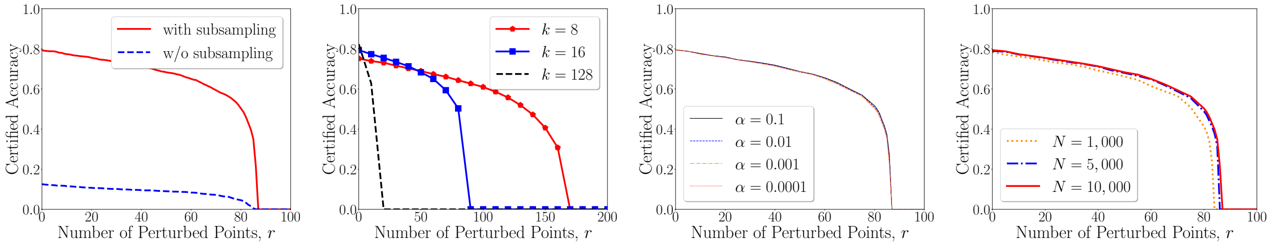
(a) Point perturbation attacks.

(b) Point modification attacks.

(c) Point addition attacks.

(d) Point deletion attacks.

Figure 5: Comparing different methods under different attacks on ScanNet.



(a) Training with vs w/o subsampling

(b) Impact of k

(c) Impact of α

(d) Impact of N

Figure 6: (a) Training the point cloud classifier with vs. without subsampling. (b), (c), and (d) show the impact of k, α, and N, respectively. The dataset is ScanNet.