Imperceptible and Robust Backdoor Attack in 3D Point Cloud

Kuofeng Gao, Jiawang Bai, Baoyuan Wu, and Shu-Tao Xia, Senior Member, IEEE, Mengxi Ya, Member, IEEE

Abstract—With the thriving of deep learning in processing point cloud data, recent works show that backdoor attacks pose a severe security threat to 3D vision applications. The attacker injects the backdoor into the 3D model by poisoning a few training samples with trigger, such that the backdoored model performs well on clean samples but behaves maliciously when the trigger pattern appears. Existing attacks often insert some additional points into the point cloud as the trigger, or utilize a linear transformation (e.g., rotation) to construct the poisoned point cloud. However, the effects of these poisoned samples are likely to be weakened or even eliminated by some commonly used pre-processing techniques for 3D point cloud, e.g., outlier removal or rotation augmentation. In this paper, we propose a novel imperceptible and robust backdoor attack (IRBA) to tackle this challenge. We utilize a nonlinear and local transformation, called weighted local transformation (WLT), to construct poisoned samples with unique transformations. As there are several hyper-parameters and randomness in WLT, it is difficult to produce two similar transformations. Consequently, poisoned samples with unique transformations are likely to be resistant to aforementioned pre-processing techniques. Besides, the distortion caused by a fixed WLT is both controllable and smooth, resulting in the generated poisoned samples that are imperceptible to human inspection. Extensive experiments on three benchmark datasets and four models show that IRBA achieves 80%+ attack success rate (ASR) in most cases even with pre-processing techniques, which is significantly higher than previous state-of-the-art attacks. Our code is available at https://github.com/KuofengGao/IRBA.

Index Terms—Backdoor attack, weighted local transformation, 3D point cloud.

I. INTRODUCTION

VISION for 3D has been developed rapidly and become more popular in real-world applications, such as autonomous driving [4], [19], robot industry [9] and augmented reality [2], [25], etc. Since PointNet [28] was first proposed, more deep learning-based methods have been introduced into 3D domain and shown tremendous success in various tasks, e.g., point cloud classification. Although large progress has been made, the security problems of 3D deep learning are not explored systematically. Recently, it has been discovered that 3D deep learning is vulnerable to backdoor attacks, which involve poisoning the training dataset [10], [16], [39], [42], [50] during the training phase. The model trained on the poisoned dataset will classify to an attacker-specified label maliciously when the samples with the trigger are present, otherwise it will behave normally. Once the backdoored model [1], [5], [18], [26], [35], [43] deploys in the safety-critical scenarios, it may lead to serious disasters.

In existing backdoor attacks for 3D deep learning, the poisoned samples are constructed by inserting some additional points (e.g., the small ball shown in the second column of Fig. 1) [16], [33], [42], or rotating the original point cloud [16] (e.g., the third column of Fig. 1). However, these attacks may not succeed in practice, as their poisoned samples are not resistant to some pre-processing techniques that have been commonly applied to pre-process point clouds during model training [28], [29]. For example, the trigger via additional points can be easily removed by statistical outlier removal (SOR) [51] and the effect of the rotated poisoned samples can be mitigated by the rotation augmentation. Besides, the trigger via additional points is noticeable to humans. Hence, a practical backdoor attack against point cloud models should be not only robust to aforementioned pre-processing techniques, but also imperceptible to human inspection.

In this work, we focus on the transformation-based approach to construct poisoned samples, as the additional point-based approach is perceptible and cannot surpass the SOR operation. However, as mentioned above, the transformation-based poisoned samples may be vulnerable to the data augmentation adopted in model training. One possible reason we speculate is that the data augmentation could produce augmented samples with similar transformations with poisoned samples, such that the steady mapping between poisoned samples and the target class cannot be learned by the model. To verify this point, we conduct a brief experiment by constructing poisoned samples with a 10° rotation and adopt the random
rotation augmentation in model training, with the range of 
(0°, 10°), (0°, 20°), (0°, 360°), respectively. The correspond- 
ing attack success rates (ASR) are 97.5%, 11.7% and 3.4%, 
respectively. It tells that if the rotation used for poisoned 
samples is covered by the random rotation augmentation, then 
the attack performance dramatically drops, which verifies our 
speculation to some extent.

Thus, we believe more complex transformations that are dif-
ficult to reproduce are desired to construct poisoned samples. 
We choose the weighted local transformation (WLT) [12], 
which was proposed as an augmentation technique to learn 
better 3D point cloud models. As shown in Fig. 2-(a), it con-
ists of two steps: firstly conducting global transformations 
(e.g., rotation, scaling) around multiple randomly used anchor 
points to obtain multiple transformed samples, then merging 
these samples into one unified transformed sample through 
smooth aggregation. Note that the distortion between the final 
transformed sample and the original sample is nonlinear and 
local (i.e., the distortion is local location). It is determined by 
several hyper-parameters (including the number of anchors, the 
types and values of global transformations) and the random-
ness on anchor sampling and specific global transformation. Even 
with the same hyper-parameters, it is also difficult to 
obtain two similar transformed samples if conducting WLT 
transformations twice. The reproducing difficulty facilitates to 
produce unique transformations.

Inspired by the above analysis, we propose an effective 
backdoor attack, called imperceptible and robust backdoor 
attack (IRBA). It utilizes the WLT with fully fixed hyper-
parameters (i.e., fixed anchors, fixed global transformations) to 
generate poisoned samples with a unique nonlinear and local 
transformation. Even the defender knows the WLT is used 
for poisoning and also utilizes WLT for data augmentation in 
training, it is still difficult to generate augmented samples 
with similar transformations compared to poisoned samples. 
Consequently, the WLT-based poisoned samples are resistant 
to the data augmentations (no matter WLT or any global 
transformation) adopted in model training, which will be 
experimentally verified later (see Section VII and Table XI). 
Moreover, due to the controllability (i.e., the distortion of every 
point could be exactly computed) and smoothness of the dis-
tortion, WLT-based poisoned samples are also imperceptible 
to human inspection and can bypass the SOR operation (see 
the last column of Fig. 1), which will be analyzed and verified 
later.

In summary, our main contributions are three-fold.
- We demonstrate that existing backdoor attacks with 
PointBA-Rotation poisoned samples are vulnerable to 
random data augmentation and provide a reasonable 
explanation.
- We propose an imperceptible and robust backdoor attack 
method by utilizing a nonlinear and local transformation, 
weighted local transformation (WLT).
- We conduct extensive experiments to show the superiority 
of IRBA to previous state-of-the-art backdoor attacks in 
3D point cloud.

II. RELATED WORK

A. Deep Learning in 3D Point Cloud

Point clouds are a well-known data structure in 3D domain, 
which contain a set of unordered point coordinates. Promoted 
by the deep learning, point-based learning [24], [30], [31], 
[44], [49] has become increasingly popular due to its promis-
ing performance. Qi et al. [28] first designed a novel neural 
network PointNet to directly learn the point-wise features. 
It adopts the symmetric function, max-pooling, to preserve 
the order-invariant property of point clouds. Motivated by 
PointNet, PointNet++ [29] further exploits the set abstraction 
layers to enhance the multi-scale information extraction. More-
over, the graph neural network and X-convolution operation 
are introduced in DGCNN [37] and PointCNN [17] to obtain 
more compact representations. The above architectures have 
been extended to other complex tasks in 3D point cloud, 
such as 3D semantic segmentation [14], [48] and 3D object 
detection [15], [36], [52], but the security problems like 
backdoor attacks have not been explored well.

B. Backdoor Attack in 3D Point Cloud

Li et al. [16] first extended the backdoor attack to 3D point 
cloud. Analog with stamping a patch at the corner of an image 
in 2D backdoor attack [8], they launched additional points 
like a ball near the 3D object to inject the 3D deep learning. 
Xiang et al. [42] generated backdoor points by optimizing 
their spatial locations and performing the attack under a strict 
setting without knowing the victim architecture successfully. 
However, these two ball-based attacks can be discovered easily 
by humans. To bypass the human inspection, Li et al. [16] also 
investigated the rotation operation as the 3D backdoor trigger 
and implement an imperceptible attack.

Fig. 1. Visualization of the clean point cloud and poisoned point clouds from different backdoor attacks, including the PointBA-Ball attack [16], [42], the 
PointBA-Rotation attack [16] and our IRBA.
The previous works \cite{16,42} in 3D backdoor learning provide two potential mitigation methods customized for 3D backdoor attacks. One is the statistical outlier removal which uses the \(k\)-nearest neighbors (kN\(\mathbb{N}\)) algorithm to define the distance metric to remove the abnormal points in a point cloud. Therefore, the attack performance of the PointBA-\(\mathcal{B}\)all attack can be easily mitigated by SOR. The other is inspired by the common 3D point cloud augmentation technique, including rotation, scaling and point-wise jitter, etc. Since 3D deep learning usually adopts them to improve the generalization of the model, the PointBA-Rotation attack will be erased during the actual training. Besides, previous works have explored empirical defenses \cite{6,20} and certified defenses \cite{21,23} against adversarial attacks in 3D point cloud classification, which may inspire future backdoor defense research.

### III. Threat Model

#### A. Attacker’s Capacities

We consider a classical scenario following existing works, the poison-label backdoor attack. The attacker is allowed to craft a small number of the poisoned samples. These poisoned samples are injected into the training set, labeled with the ground-truth label backdoor attack. The attacker is allowed to craft a small number of the poisoned samples. These poisoned samples can be pre-processed before they are fed into the model.

#### B. WLT-Based Poisoned Samples Generation

The attacker can manipulate the whole dataset. Besides, the number of poisoned samples, denoted as \(M\), is intentionally kept much lower than the total number of samples \(N\) in the dataset. However, the attacker can not know the victim architecture or control over the backdoor training process. Therefore, in the realistic attack scenario, the poisoned training samples can be pre-processed before they are fed into the model.

#### C. Attacker’s Goals

The goal of the attacker is that any model trained on the poisoned dataset will return the attacker-specified malicious target label when meeting the point clouds injected with the trigger transformation. Meanwhile, it should also preserve the clean accuracy in the absence of the trigger transformation. Furthermore, when generating the poisoned samples, it is essential to ensure that the geometric shape of the point clouds is not significantly altered, as this would compromise the imperceptibility of the attack.

### IV. Imperceptible and Robust Backdoor Attack (IRBA)

#### A. Problem Definition

Given a training set \(D = \{(X_i, y_i)\}_{i=1}^N\) which contains \(N\) point cloud samples, where \(y_i \in \{1, 2, \ldots, C\}\) denotes the ground-truth label of the point cloud sample and \(C\) is the number of class. \(X_i = [x_{i1}, x_{i2}, \ldots, x_{iK}]^T \in \mathbb{R}^{K \times 3}\) denotes that every point cloud can be decomposed into \(K\) points and each point \(x_i \in \mathbb{R}^3\) has its 3-dimension position coordinate.

Based on original \(M\) point cloud samples, the attacker crafts a small number of the poisoned samples as \(D_b = \{(\tilde{X}_i, y_i)^{\theta}\}_{i=1}^M\), where \(y_i\) is the target label and \(M \ll N\). \(D_c = \{(X_i, y_i)\}_{i=1}^{N-M}\) denotes the remaining clean sample set. The poisoned dataset is built by mixing \(D_b\) and \(D_c\). \(f_\theta\) denotes the victim classification model with parameters \(\theta\). Because we assume that the attacker can not change the training process, \(f_\theta\) should be obtained by normally training on the poisoned dataset with minimizing the following objective:

\[
\min_\theta \sum_{(X,y) \in D_c \cup D_b} \mathcal{L}(f_\theta(P(X)), y),
\]

where \(\mathcal{L}(\cdot, \cdot)\) indicates the loss function during the training stage. We also consider the pre-processing on the training samples, denoted as \(P(\cdot)\), including SOR or 3D data augmentations. They have become common configurations for cleaning the point clouds or improving the performance of the 3D model \cite{28,29,51}. Hence, it is necessary to ensure the backdoor can be injected even the poisoned samples are processed by \(P(\cdot)\).

#### B. WLT-Based Poisoned Samples Generation

The main focus of this paper is to craft poisoned samples which are robust to various pre-processing techniques. Different from inserting additional points in \cite{16} and \cite{42} and the linear transformation in \cite{16}, we utilize WLT to generate each point \(\hat{x}_i\) in the poisoned sample \(\hat{X}\) by a nonlinear point-wise function \(G : \mathbb{R}^3 \rightarrow \mathbb{R}^3\). The pipeline is described in Fig. 2.

1) Multi-Anchor Transformation: We firstly select the anchor points before performing the transformation. In order to ensure the anchors don’t gather in one local region to cause the uneven deformation, we utilize Farthest Point Sampling (FPS) algorithm to choose a few transformation anchors \(\{a_j\}_{j=1}^W\) on the surface of the point cloud \(X\), where \(W\) is the number of anchor points. FPS first samples one initial point randomly and repeatedly selects the points farthest from the previous points. It can lead to more subtle variations on the 3D shape and meanwhile introduce more randomness.

After selecting the anchor points, we perform anchor-based transformation by taking each anchor point as the centroid and applying the rotation and scaling operation. We implement the transformation by a rotation matrix \(R \in \mathbb{R}^{3 \times 3}\) and a scaling matrix \(S \in \mathbb{R}^{3 \times 3}\). For simplicity, we rotate the 3D point cloud along the \(x\), \(y\) and \(z\)-axis with the same angle \(\alpha\). We define the rotation matrix in multi-anchor transformation as \(R = R_x(\alpha)R_y(\alpha)R_z(\alpha)\), which represents the basic 3D point cloud rotation by an angle \(\alpha\) along three axes in order. The formulations of \(R_x(\alpha)\), \(R_y(\alpha)\) and \(R_z(\alpha)\) are as follows:

\[
R_x(\alpha) = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \alpha & -\sin \alpha \\
0 & \sin \alpha & \cos \alpha
\end{bmatrix},
\]

\[
R_y(\alpha) = \begin{bmatrix}
\cos \alpha & 0 & \sin \alpha \\
0 & 1 & 0 \\
-\sin \alpha & 0 & \cos \alpha
\end{bmatrix},
\]

\[
R_z(\alpha) = \begin{bmatrix}
\cos \alpha & -\sin \alpha & 0 \\
\sin \alpha & \cos \alpha & 0 \\
0 & 0 & 1
\end{bmatrix}.
\]
The scaling matrix is defined as a diagonal matrix, denoted as $S = \text{diag}(s, s, s)$, which means scaling with the same size $s$ along three axes. Based on the selected anchor $a_j$ and the rotation and scaling operation, we define the anchor-based transformation for an input point $x_i$ as follows:

$$T(x_i, a_j) = R S(x_i - a_j) + a_j. \quad (2)$$

Note that the rotation matrix and scaling matrix are predefined by the attacker and fixed to generate all poisoned samples. For each anchor point $a_j$, we perform the above transformation for each point in $X$, resulting in $W$ transformed point clouds.

2) Smooth Aggregation: In the step of aggregating the $W$ transformed point clouds, we introduce the Gaussian kernel function $K_h(x_i, a_j) = \exp(-\|x_i - a_j\|^2 / 2 h^2)$ to calculate the weight for each pair of $x_i$ and $a_j$, where the bandwidth $h$ is a smoothing parameter. Given the kernel function, the point $\hat{x}_i$ in the poisoned sample $\hat{X}$ can be generated as:

$$\hat{x}_i = G(x_i) = \frac{\sum_{j=1}^{W} K_h(x_i, a_j) T(x_i, a_j)}{\sum_{k=1}^{W} K_h(x_i, a_k)}. \quad (3)$$

**Proposition 1** (Originally Defined in [12]): Based on the Gaussian kernel, the aggregation function $G(\cdot)$ is smooth when $T(\cdot, \cdot)$ is defined as Eq. (2), where the smoothness of a function is that all partial derivatives exist and are continuous.

One advantage of our aggregation is that $G(\cdot)$ introduces the nonlinear weighting due to the Gaussian kernel. Since common 3D data augmentations are linear, nonlinearity helps to improve the robustness of our attack when the poisoned samples are processed by $P(\cdot)$. The other advantage is that our aggregation function is smooth according to Proposition 1. Therefore, the aggregation is prone to preserve the global shape of the point cloud and change its local structure in a smooth manner, which ensures the imperceptibility of the poisoned samples. We visualize the poisoned samples in Fig. 3, which confirms the effect of our smooth aggregation. Besides, with the increasing of the number of anchors, the changes of the local structure become more imperceptible. In summary, our IRBA backdoor attack is shown in Algorithm 1.

V. EXPERIMENTS

A. Evaluation Setup

1) Datasets and Target Models: We adopt three benchmark datasets in the 3D point cloud classification task, i.e., ModelNet10 [40], ModelNet40 [40] and ShapeNetPart [46]. Following [16], we build training and test point clouds for each dataset. Besides, we follow [28] to uniformly sample 1,024 points from the surface of each dataset and normalize them into a unit ball. We evaluate all attack methods on four categories of popular point cloud classification models, i.e., PointNet [28], PointNet++ [29], DGCNN [37] and PointCNN [17], denoted as “PN”, “PN++”, “DGCNN” and “PCNN”.

2) Baseline Methods: We compare IRBA with two existing backdoor attacks in 3D point cloud classification in [16]: PointBA-Ball backdoor attack and PointBA-rotation backdoor attack (dubbed “PointBA-Ball” and “PointBA-Rotation” respectively). For the PointBA-ball attack, we place a ball with the fixed radius 0.05 centered at (0.05, 0.05, 0.05) near the 3D object, where the ratio between the trigger points and total...
Fig. 3. Poisoned samples generated by WLT under different number of the anchors $W$. The CD score [22], [47] is calculated between the clean sample and the sample with the trigger transformation, where the lower CD score corresponds to more imperceptibility.

### Table I

| Models          | ACC (%) | ASR (%) | ACC (%) | ASR (%) | ACC (%) | ASR (%) | ACC (%) | ASR (%) | ACC (%) | ASR (%) |
|-----------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| ModelNet10      | PointBA-Ball | 92.5 | 91.3 | 97.4 | 93.2 | 91.8 | 97.0 | 96.4 | 98.6 | 94.6 | 97.1 |
|                 | PointBA-Rotation | 90.6 | 92.3 | 97.4 | 93.2 | 91.8 | 97.0 | 96.4 | 98.6 | 94.6 | 97.1 |
|                 | IRBA (Ours) | 98.3 | 100 | 98.2 | 99.8 | 98.1 | 96.4 | 97.1 | 99.8 | 99.2 |
| ModelNet40      | PointBA-Ball | 93.2 | 100 | 92.3 | 98.3 | 93.2 | 98.1 | 96.4 | 98.6 | 95.8 | 95.7 |
|                 | PointBA-Rotation | 91.3 | 100 | 91.2 | 95.9 | 91.5 | 94.0 | 95.8 | 98.8 | 99.0 |
|                 | IRBA (Ours) | 98.9 | 100 | 98.8 | 96.6 | 98.9 | 98.4 | 99.0 | 99.8 |
| ShapeNetPart    | PointBA-Ball | 95.1 | 100 | 92.2 | 98.3 | 93.2 | 98.6 | 96.4 | 98.6 | 95.8 | 95.7 |
|                 | PointBA-Rotation | 93.1 | 100 | 91.2 | 95.9 | 91.5 | 94.0 | 95.8 | 98.8 | 99.0 |
|                 | IRBA (Ours) | 98.9 | 100 | 98.8 | 96.6 | 98.9 | 98.4 | 99.0 | 99.8 |

#### 3) Attack Settings

For all methods, we set the poisoned rate $M/N = 0.1$, which indicates the proportion of the number of poisoned samples in the poisoned dataset. Specially, poisoned samples are sampled from the non-target class randomly and the target label on ModelNet10, ModelNet40 and ShapeNetPart is chosen as “Table” ($y_t = 8$), “Toilet” ($y_t = 35$) and “Lamp” ($y_t = 8$) respectively. For the proposed IRBA, we set the number of anchors $W$ as 16. The angle of the rotation $\alpha$ and the scaling size $s$ in the multi-anchor transformation is set as $5^\circ$ and 5, respectively. The bandwidth $h$ in the smooth aggregation is set to 0.5. We use the framework PyTorch [27] to implement all the experiments. We adopt the Adam optimizer [13] with the learning rate 0.001 to train all the models for 200 epochs and the training batch size is set to 32. All the above experiments are run on a NVIDIA RTX A5000 GPU. Note that the training schedule on the poisoned dataset is the same as that on the clean dataset.

#### 4) Evaluation Metrics

To verify the performance of backdoor attacks, we adopt attack success rate (ASR) as the metric, which is the fraction of samples from non-target class with the trigger classified to the target label by the backdoored model. A higher ASR means a stronger attack. Besides, we also use the test accuracy (ACC) to measure the effect of the backdoor attack on the clean dataset. Following [22], [41], and [47], we adopt the Chamfer Distance (CD) to investigate the deformation magnitude of poisoned point clouds compared to clean samples.

#### B. Attack Results

The results of all backdoored models are summarized in Table I. It shows that the ASR of our proposed IRBA is competitive with previous baseline backdoor attacks and greater than 90% on three datasets across all the models, which demonstrates the threat of IRBA in point cloud classification. To investigate the influences of different target labels, we evaluate our method with different target labels against PointNet++ in Table II. We evaluate the in-class accuracy of the clean-trained model on the three datasets and select six classes as the target label for the backdoor attack. The results demonstrate that the attack performance of our IRBA is not influenced by the target label. In particular, for the
Algorithm 1 Imperceptible and robust backdoor attack in 3D point cloud

Input: The victim classification model $f_θ$, the training set $D$, the rotation matrix $R$, the scaling matrix $S$ and the number of anchors $W$ in the trigger injection algorithm $G(\cdot)$, the target label $y_t$, the number of poisoned samples $M$. 

Output: Backdoored model $f_θ'$. 

1: Initialize $R$ and $S$. 
2: $D_b = \text{RandomSelect}(D, M)$. 
3: $D_c = D \cup D_b$. 
4: for each sample $(X, y)$ from $D_c$ do 
5: $X = \{x_1, x_2, \ldots, x_k\}$. 
6: Choose anchors $\{a_j\}_{j=1}^W = \text{FPS}(X, W)$, where $\{a_j\}_{j=1}^W \subset X$. 
7: for each point $x_i$ from $X$ do 
8: $T(x_i, a_j) = KS(x_i - a_j) + a_j$. 
9: $K_h(x_i, a_j) = \exp(-\|x_i - a_j\|^2/2 h^2)$. 
10: $\hat{x}_i = G(x_i) = \frac{\sum_{k=1}^W K_h(x_i, a_j)T(x_i, a_j)}{\sum_{k=1}^W K_h(x_i, a_k)}$. 
11: end for 
12: $X = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_k]^T$. 
13: Replace $(X, y)$ in $D_b$ with $(\hat{X}, y)$. 
14: end for 
15: Train the model $f_θ$ on the poisoned dataset $D_c \cup D_b$ with the below objective function: 

$$
\min_{\theta} \sum_{(X, y) \in D_c \cup D_b} \mathcal{L}(f_θ(P(X)), y).
$$

C. Resistance to Various Pre-Processing Techniques

We evaluate the resistance of three backdoor attacks against the statistical outlier removal (SOR) and random rotation on three datasets across four models. Besides, we further use various transformations in the 3D augmentation to test the robustness of backdoor attacks. The default target model and dataset are PointNet++ and ModelNet10, respectively.

1) Resistance to SOR and Random Rotation: Statistical Outlier Removal (SOR) [51] involves computing the distances between each point and its $k$ nearest neighbors. Then, the points with the top $N$ largest distances are considered as outliers and are removed from the point cloud. Specifically, we set $k$ as 30 and $N$ as 100 in our experiment. Random rotation along $z$-axis ($R$) is implemented by a rotation matrix $R_α(α) \in \mathbb{R}^{3 \times 3}$. Specifically, $α$ denotes the rotation angle along $z$-axis and will be set under 20° randomly during every augmentation. We combine the SOR and random rotation along the $z$-axis as the data pre-processing techniques during the training stage. Table IV demonstrates that the SOR and the random rotation can degrade the PointBA-Ball and PointBA-Rotation backdoor attack performance by a large margin but have little effect on IRBA. To explore why our method can still achieve a high ASR compared with the other two backdoor attacks, we visualize samples, the feature space of the poisoned dataset and the feature space of the poisoned dataset after SOR and random rotation in Fig. 4, Fig. 5 and Fig. 6.

As shown in Fig. 4, the PointBA-Ball trigger can be completely eliminated after SOR. Although SOR doesn’t break the PointBA-Rotation trigger, it can be undermined easily by the random rotation. Hence, due to the damage of the trigger of two attacks, the model is difficult to distinguish the features between clean samples and poisoned samples, which can be seen in Fig. 6. As for IRBA, it does not introduce outliers such that it is robust to SOR and the linear operation rotation is not enough to destruct the unique structure generated by our nonlinear method. Fig. 6 shows that our poisoned samples can cluster independently from other classes in the feature space, indicating that our attack succeeds to implant the backdoor into the model.

2) Resistance to Various Augmentations: In addition to SOR and the random rotation, we further apply five data augmentations to explore the effect on three attacks. (1) We
TABLE IV
ACC (%) and ASR (%) of Backdoored Models with the PointBA-Ball Trigger, the PointBA-Rotation Trigger and Our IRBA Trigger on Three Datasets Under SOR and Random Rotation During Training. The Best Results Are Highlighted in Bold

| Models    | ModelNet10 |              |              | ModelNet40 |              |              | ShapeNetPart |              |              |
|-----------|------------|--------------|--------------|------------|--------------|--------------|--------------|--------------|--------------|
|           | ACC        | ASR          | ACC          | ASR        | ACC          | ASR          | ACC          | ASR          | ACC          | ASR          |
| PN        | 84.2       | 19.4         | 86.9         | 8.17       | 88.5         | 90.6         | 78.7         | 14.2         | 82.1         | 9.97         | 85.7         | 83.7         | 93.6         | 21.4         | 97.6         | 14.5         | 97.7         | 98.8         |
| PN++      | 83.1       | 11.7         | 88.8         | 8.41       | 92.4         | 96.6         | 78.4         | 19.7         | 82.5         | 11.4         | 88.1         | 90.2         | 87.8         | 29.6         | 94.3         | 10.7         | 98.5         | 98.8         |
| DGCNN     | 91.8       | 10.2         | 91.1         | 6.97       | 92.9         | 92.5         | 87.8         | 7.82         | 88.7         | 4.23         | 90.0         | 85.1         | 94.0         | 6.07         | 96.8         | 7.76         | 98.4         | 97.0         |
| PCNN      | 81.9       | 19.6         | 88.7         | 11.2       | 87.3         | 73.3         | 80.1         | 26.3         | 82.7         | 5.01         | 82.1         | 84.2         | 94.4         | 20.1         | 96.3         | 14.8         | 97.4         | 83.8         |

Fig. 4. Visualization of the point cloud with the PointBA-Ball trigger [16], [42], the PointBA-Rotation trigger [16] and our IRBA trigger and those after SOR and random rotation.

Fig. 5. T-SNE visualization of the poisoned dataset in the feature space generated by three backdoor attacks on ModelNet10. The target label is chosen as “Table” (y = 8), visualized in the black color.

Fig. 6. T-SNE visualization of the poisoned dataset in the feature space generated by three backdoor attacks under SOR and random rotation on ModelNet10. The target label is chosen as “Table” (y = 8), visualized in the black color.

perform random rotation under 360° around each of the three axes (x, y and z-axis). Each rotation can be represented by using a rotation matrix. (2) We achieve the scaling operation in a uniform way by using a scaling matrix \( S = \text{diag}(s, s, s) \). The size of the point cloud is changed uniformly in all dimensions by a scale factor \( s \), set randomly between 0.5 to 1.5 during every
augmentation. (3) The shift transformation involves moving all points in a point cloud by a specified amount in each dimension, which is set randomly ranging from −0.1 to 0.1. (4) The dropout transformation randomly removes a portion of points from the point cloud, simulating missing or incomplete data or occlusions. The percentage of points to be removed is controlled by a dropout rate parameter, set randomly under 0.2 during every augmentation. (5) The jitter transformation involves perturbing the coordinates of each point in the point cloud by adding a small amount of random noise. Given a Gaussian distribution \(N(0, 0.02)\) and clipping within 0.05, the jitter transformation for a point cloud \(X\) can be implemented as \(T(X) = X + N(0, 0.02)\). Note that the addition operation is element-wise.

As shown in Table V, our method can achieve a higher ASR than the PointBA-Ball and PointBA-Rotation backdoor attack under all the above augmentations during the backdoor training. Although ASR of IRBA decreases with the accumulation of various augmentations, IRBA can still achieve at least 47.5% ASR, which benefits from the nonlinear and local distortion between our poisoned samples and clean ones.

3) Resistance to More Pre-Processing Techniques: We try the defaulted Simple Random Sampling (SRS) [45] during the training and show the results in Table VI. Besides, we utilize the original upsampling network in DUP-Net [51] to upsample the training dataset from 1024 points to 2048 points after SOR layer and the results are shown in Table VI. Dup-Net consists of two defense modules designed for the 3D point cloud processing tasks, i.e., an outlier removal layer and an upsampling layer. The first outlier removal layer employs a statistical method to eliminate outlier points in the input point cloud and generate feature vectors via the convolutional operation. The subsequent upsampling layer increases the resolution of the feature maps, to preserve spatial information in the point cloud, which can enhance the accuracy of the downstream tasks. It can be observed that all the above results demonstrate that IRBA can resist these two pre-processing techniques, which verifies its superiority.

4) Resistance to Adversarial Training: Following [20], we adopt the \(L_2\) distance-based PGD adversarial training. We conduct experiments using PointNet++ on ModelNet10 with the perturbation budget \(\epsilon \in \{0.0, 0.5, 1.0, 2.0\}\). We show the results in Table VII. It can be observed that rotation-based backdoor attack is vulnerable to adversarial training, showing very low ASR when the perturbation budget is greater than 1.0. Besides, PointBA-Ball is resistant to the adversarial training, due to its noticeable ball-based trigger. In contrast, our proposed IRBA with an imperceptible backdoor trigger can still work well under the PGD adversarial training with various perturbation budgets.

D. Ablation Study

1) Effect of the Rotation Angle \(\alpha\) and Scaling Size \(s\): Here, we present the results under various rotation angles \(\alpha\) and scaling sizes \(s\) of our trigger. As shown in Fig. 7a and Fig. 7b, both \(\alpha\) and \(s\) can affect the ASR of our attack jointly, where the larger \(\alpha\) and \(s\) indicate the higher attack performance. Our proposed IRBA with an imperceptible backdoor trigger can still work well under the PGD adversarial training with various perturbation budgets.

### Table V

**ACC (%) and ASR (%) of Backdoored Models With the PointBA-Ball Trigger, the PointBA-Rotation Trigger and Our IRBA Trigger on ModelNet10 Under Various Pre-Processing Techniques During Training.**

| Pre-processing Technique | PointBA-Ball | PointBA-Rotation | IRBA (Ours) |
|--------------------------|--------------|-----------------|-------------|
|                          | ACC | ASR | ACC | ASR | ACC | ASR | ACC | ASR | ACC | ASR |
| SOR                      | X   | X   | X   | X   | X   | X   | X   | X   | X   | X   |
| R                        | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   |
| R3                       | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   |
| Scaling                  | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   |
| Shift                    | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   |
| Dropout                  | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   |
| Jitter                   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   | ✓   |
|                         | 93.2 | 100 | 92.3 | 98.3 | 93.2 | 98.1 | 84.4 | 17.2 | 91.3 | 98.6 |
|                         | 84.4 | 17.2 | 91.3 | 98.6 | 92.9 | 97.2 | 83.1 | 11.7 | 88.8 | 84.1 |
|                         | 92.4 | 96.6 | 87.5 | 13.7 | 89.9 | 72.4 | 85.7 | 28.9 | 87.5 | 13.7 |
|                         | 91.2 | 63.1 | 90.6 | 6.49 | 91.2 | 63.1 | 90.8 | 6.97 | 90.6 | 6.49 |
|                         | 90.2 | 52.1 | 89.2 | 9.86 | 90.2 | 60.4 | 89.7 | 3.96 | 89.2 | 9.86 |
|                         | 90.7 | 47.5 | 90.7 | 9.13 | 90.7 | 47.5 | 88.8 | 9.25 | 90.7 | 9.13 |

### Table VI

**ACC (%) / ASR (%) of Three Attacks on ModelNet10 Against PointNet++ after SRS and DUP-Net**

| Pre-processing Technique | PointBA-Ball ACC | PointBA-Rotation ACC | IRBA (Ours) ACC |
|--------------------------|-----------------|-------------------|----------------|
| SRS                      | 92.9 | 100 | 93.6 | 96.7 |
| SRS-SOR+R                | 81.4 | 38.3 | 87.8 | 10.0 |
|                         | 91.2 | 81.1 | 90.4 | 84.3 |
|                         | 93.3 | 98.1 |
| Dup-Net                 | 91.2 | 100 | 93.6 | 96.7 |
| Dup-Net+SOR+R           | 81.1 | 16.3 | 82.8 | 39.5 |
|                         | 93.3 | 98.1 |

### Table VII

**ACC (%) / ASR (%) of Three Attacks on ModelNet10 Under PGD Adversarial Training**

| Perturbation Budget \(\epsilon\) | PointBA-Ball ACC | PointBA-Rotation ACC | IRBA (Ours) ACC |
|----------------------------------|-----------------|-------------------|----------------|
| 0.0                              | 93.2 | 100 | 92.3 | 98.3 |
| 0.5                              | 88.6 | 99.4 | 86.7 | 88.6 |
| 1.0                              | 89.7 | 99.8 | 84.3 | 54.6 |
| 2.0                              | 73.6 | 97.2 | 41.5 | 16.7 |
|                                  | 89.1 | 89.5 | 89.1 | 89.5 |
2) Effect of the Number of Anchors $W$: In this part, we sample different numbers of the anchors $W$ to investigate the effect on our attack. From the results in Fig. 7c, we can observe that fewer anchors will lead to a higher ASR on three datasets. As such, the anchor-based transformation will be performed within a more finite neighborhood of the chosen anchors and result in an irregular deformation on the shape of the point cloud. The visualization and CD score are shown in Fig. 3.

3) Effect of the Kernel Bandwidth: We explore the effect of the kernel bandwidth $h$ towards our proposed attack on three datasets and show the results in Fig. 8. It can be seen that the attack effectiveness becomes stronger with the increase of the poisoned rate when it is lower than 0.08. Meanwhile, once the poisoned rate is larger than 0.1, the ASR of our attack can exceed 95% and remain at a high level across three datasets. The high ASR with a low poisoned rate can ensure the stealthiness of IRBA.
Parameters: WLT used in our backdoor attack has many flexible for IRBA to choose parameters of WLT. The results are viewed as a reference rather than the unique parameters for the specific parameter values we used in our study should be 90%). Therefore, this empirical observation allows us to select the combinations that can reach a high ASR (poisoned ones, respectively. As shown in Fig.3, when the poisoned samples, which are measured by the attack success rate (ASR) and the Chamfer Distance (CD) between clean samples and samples, meeting the requirement of imperceptibility. Therefore, this empirical observation allows us to select the values of parameters which result in \( CD \times 100 \leq 0.5 \). Here, we investigate the effect on ASR and CD under different values of the parameters and further prove that it is flexible for IRBA to choose parameters of WLT. The results are shown in Fig. 7. We can see that a smaller rotation angle, smaller scaling size, large number of anchors and large kernel bandwidth result in more imperceptible poisoned images but a lower ASR and vice versa. For each dataset, considering the criterion \( CD \times 100 \leq 0.5 \), there are many possible parameter combinations that can reach a high ASR (> 90%). Therefore, the specific parameter values we used in our study should be viewed as a reference rather than the unique parameters for our proposed IRBA.

VI. HUMAN PERCEPTUAL STUDY

A. Demographics of Participants

To evaluate the visual perceptibility of poisoned samples generated by different attack methods, we conduct a human perceptual study involving 50 participants, all aged above 18 years old and majoring in computer science-related fields. Of these participants, 70% are male, and 30% are female. Additionally, 96% have knowledge of machine learning, 44% are familiar with point cloud technology, and 74% have experience with backdoor learning. The demographics of participants are summarized in Table VIII. These participants are publicly recruited as volunteers with a major in computer science. Most of them have knowledge about machine learning and computer vision. After undergoing preliminary training, they are well-equipped to provide fair and unbiased scores. All participants are shown the same set of samples during the study to ensure consistency in the evaluation process. Furthermore, we carefully consider the ethical aspects and potential risks associated with the research involving human subjects. The information we collect is only the probability score and does not involve any personal information. We have obtained the Institutional Review Board (IRB) approval for our human perceptual study, where the number of our IRB approval is CUHKSZ-D-20230021. Participants are provided with instructions on the task and objectives of our study. The prompts are as follows:

Welcome to our human perceptual study! In this study, we aim to evaluate the perceptibility of different poisoned samples of 3D point cloud data. Point clouds are a collection of data points in a 3D coordinate system, commonly used in fields such as computer vision and robotics. You will be presented with 36 sets of point cloud samples. Each set consists of nine point cloud samples and can include poisoned samples, which are modified based on clean samples.

Your task is to carefully examine each set of point cloud samples and determine the likelihood whether some of the samples have been modified in each set. Please provide a score from 1 to 5 for each set, where 1 represents a low probability that some samples in a set have been modified, and 5 represents a high probability. Keep in mind that not all samples can be modified, and the poisoned samples can be subtle and difficult to detect.

Thank you for your participation! Your valuable insights will help us understand the perceptibility of poisoned samples in point cloud data and contribute to the development of more robust and secure systems.

B. Results

We list overall results in Table IX. In our study, the goal is to ask participants to assign scores \( \in \{1, 2, 3, 4, 5\} \) to evaluate the likelihood of some samples being modified. This approach enables us to ascertain whether the victim can detect poisoned samples when exposed to a mixture of clean samples and poisoned samples. Hence, we randomly mix clean samples and poisoned samples and show them to 50 participants.

![Fig. 8. ASR (%) of IRBA with different poisoned rates on three datasets.](image_url)
In addition to these three categories of sample sets, we also include a baseline consisting of only clean samples. A lower score corresponds to a lower probability, \(i.e.,\) a less perceptible poisoned sample.

We can observe that the score of the samples with the PointBA-Ball trigger is the highest because it contains noticeable additional points. In contrast, the poisoned samples of the PointBA-Rotation method and our proposed IRBA achieve relatively low scores, which indicates the visual stealthiness of these two backdoor attacks. To quantify the imperceptibility of three backdoor attacks, we also calculate the CD \(\times 100\) on three datasets, as shown in Table X. It indicates that the PointBA-Rotation attack and our IRBA can achieve the lowest CD on ModelNet and ShapeNet, respectively. Besides, regarding the ratio of samples being correctly classified, we consider scores of 1 and 2 as indicating no poisoned samples in a set, while scores of 3, 4, and 5 suggest the presence of poisoned samples in a set. Based on the results from the human perceptual study, the respective ratios of sets containing poisoned samples are 0% for clean samples, 100% for Pointba-Ball, 18% for Pointba-Rotation, and 14% for IRBA samples. The above results and visualizations confirm that IRBA is hardly
perceptible and is difficult to be spotted by humans when our poisoned samples are mixed with clean ones. We randomly select a sample set used in our human perceptual study for each attack and visualize them in Fig. 9. The first two rows are clean samples and the third row is the poisoned samples. Besides, more poisoned samples generated by the the PointBA-Ball attack, the PointBA-Rotation attack and IRBA are shown in Fig. 10.

VII. POTENTIAL ADAPTIVE DEFENSE

A. Experimental Setups

In the above experiments, we assume that the victim has no information about our attack. In this section, we consider a more challenging setting, where the victim knows the existence of IRBA and trains the model with an adaptive defense. Based on the victim’s knowledge, the adaptive defense can be divided into two categories. One is that the victim only knows the multi-anchor transformation. The other is that both multi-anchor transformation and smooth aggregation are known by the victim. However, the victim can not get the specific parameters of IRBA in both settings. We denote them as “Average” and “Smooth”, respectively.

The “Smooth” adaptive defense is summarized in Algorithm 2. It adopts the WLT transformation with random parameters during the training and the process is as follows. First, define a range of values for each of the parameters that will be used to transform the clean samples. Then, for each batch of clean samples in the training dataset, randomly select a value for each of the random parameters from the pre-defined range and apply the WLT transformation with the selected parameters to each sample in the batch. Moreover, train the model using these transformed samples and repeat this process until the model has been trained on the entire dataset. The “Average” adaptive defense replaces the WLT transformation in the “Smooth” adaptive defense with the average transformation and other steps are the same as the “Smooth” adaptive defense. Specifically, the average transformation first generates multiple transformed point clouds using multi-anchor transformation. Next, each of these transformed point clouds is assigned equal weights. Finally, they are aggregated into a single point cloud by computing the weighted average of the individual transformed point clouds. This process ensures that each transformed point cloud contributes equally to the resulting aggregated point cloud. Specially, we keep the settings of IRBA unchanged as Section V-A and vary the defense parameters. The results are shown in Table XI.

B. Results

In our IRBA, the rotation angle, the scaling size and the number of anchors is set to 5°, 5 and 16, respectively. We first allow the model owner to select parameters from a range of values in the data augmentation process. We can see that IRBA can still achieve about 90% ASR under “Average” defense for different parameters, which shows IRBA can resist it successfully. This is because the nonlinear operation in smooth aggregation can generate more diverse 3D shapes to make IRBA robust against the linear average aggregation. As for the smooth adaptive defense, we set the range of the bandwidth from 0.1 to 0.9. Table XI shows that despite the “Smooth” defense during backdoor training, the ASR of IRBA is still greater than 55.4% when the adaptive defense can not obtain the specific parameter value of IRBA. Besides, we evaluate our proposed IRBA in a more strict setting, where we assume that the model owner can precisely know one of attack parameters. Table XII shows that the ASR can still be higher than 42.3% in this very challenging setting.
TABLE XI
ACC (%) AND ASR (%) OF BACKDOORED MODELS WITH OUR IRBA TRIGGER ON MODELNET10 UNDER ADAPTIVE DEFENSE DURING TRAINING.
WHEN THE RANGE OF $\alpha$, RANGE OF $s$ AND RANGE OF $W$ ARE SET TO 0, IT CORRESPONDS TO OUR IRBA WITHOUT ANY DEFENSE, WHICH PROVIDES REFERENCE VALUES FOR COMPARISON

| Range of $\alpha$ | Range of $s$ | Range of $W$ | Average ACC | Average ASR | Smooth ACC | Smooth ASR |
|------------------|-------------|--------------|-------------|-------------|------------|------------|
| 0                | 0           | 0            | 93.2        | 98.1        | 93.2       | 98.1       |
| $-10 \sim 10$    | 1 $\sim 10$| 1 $\sim 32$  | 93.8        | 93.4        | 93.1       | 75.9       |
| $-10 \sim 10$    | 1 $\sim 10$| 1 $\sim 16$  | 93.3        | 89.2        | 92.9       | 65.6       |
| $-10 \sim 10$    | 1 $\sim 10$| 1 $\sim 8$   | 94.2        | 90.9        | 92.8       | 55.4       |
| $-10 \sim 10$    | 1 $\sim 10$| 1 $\sim 4$   | 92.9        | 89.4        | 91.7       | 60.3       |
| $-10 \sim 10$    | 1 $\sim 10$| 1 $\sim 2$   | 92.5        | 90.7        | 91.7       | 79.8       |
| $-5 \sim 5$      | 1 $\sim 5$  | 1 $\sim 32$  | 93.5        | 98.1        | 93.3       | 88.7       |
| $-5 \sim 5$      | 1 $\sim 5$  | 1 $\sim 16$  | 92.7        | 96.9        | 92.9       | 90.1       |
| $-5 \sim 5$      | 1 $\sim 5$  | 1 $\sim 8$   | 93.2        | 94.5        | 93.2       | 90.9       |
| $-5 \sim 5$      | 1 $\sim 5$  | 1 $\sim 4$   | 93.8        | 93.3        | 92.2       | 81.3       |
| $-5 \sim 5$      | 1 $\sim 5$  | 1 $\sim 2$   | 93.5        | 92.7        | 92.0       | 90.9       |

TABLE XII
ACC (%) AND ASR (%) OF BACKDOORED MODELS WITH OUR IRBA TRIGGER ON MODELNET10 UNDER ADAPTIVE DEFENSE DURING TRAINING.
SUPPOSE ONE OF ATTACK PARAMETERS CAN BE PRECISELY KNOWN BY THE VICTIM, WHEN $\alpha$, $s$ AND $W$ ARE SET TO 0, IT CORRESPONDS TO OUR IRBA WITHOUT ANY DEFENSE, WHICH PROVIDES REFERENCE VALUES FOR COMPARISON

| Parameters in adaptive defense | Average ACC | Average ASR | Smooth ACC | Smooth ASR |
|--------------------------------|-------------|-------------|------------|------------|
| $\alpha = 0$, $s = 0$, $W = 0$ | 93.2        | 98.1        | 93.2       | 98.1       |
| $\alpha = 5$, $s \in [1, 10]$, $W \in [1, 16]$ | 92.7        | 98.5        | 91.8       | 42.3       |
| $\alpha \in [-10, 10]$, $s = 5$, $W \in [1, 16]$ | 92.9        | 90.3        | 92.5       | 47.2       |
| $\alpha \in [-10, 10]$, $s \in [1, 10]$, $W = 16$ | 93.4        | 95.9        | 91.9       | 66.2       |

TABLE XIII
ACC (%) AND ASR (%) OF BACKDOORED MODELS WITH OUR IRBA TRIGGER ON MODELNET10 UNDER ADAPTIVE DEFENSE DURING TRAINING.
THE FOLLOWING METHODS ARE DIFFERENT ALGORITHMIC CHOICES IN THE ADAPTIVE DEFENSE. THE METHOD “DATA WITHOUT ANY TRANSFORMATION” CORRESPONDS TO OUR IRBA WITHOUT ANY DEFENSE, WHICH PROVIDES REFERENCE VALUES FOR COMPARISON

| Methods                                      | Average ACC | Average ASR | Smooth ACC | Smooth ASR |
|----------------------------------------------|-------------|-------------|------------|------------|
| Data without any transformation              | 93.2        | 98.1        | 93.2       | 98.1       |
| Single random transformed data                | 94.2        | 90.9        | 92.8       | 55.4       |
| Single random transformed data with original data | 93.1    | 93.1        | 92.6       | 80.1       |
| Multiple random transformed data              | 92.5        | 91.9        | 93.1       | 54.2       |
| Transformed data with fixed pre-selected parameters | 93.9    | 96.5        | 91.5       | 89.1       |

In Table XIII, we provide the results in Algorithm 2 where the range of $\alpha$ is $-10 \sim 10$, the range of $s$ is $1 \sim 10$, and the range of $W$ is $1 \sim 8$, dubbed as “Single random transformed data” which is our default choice of adaptive defenses in Table XI. In addition, we perform our IRBA on this single random transformed data combined with original data. Due to the inclusion of untransformed poisoned samples, the trigger transformation can be learned more easily, achieving an ASR of 80.1%, as shown in Table XIII. Furthermore, we generate three transformed samples by applying WLT transformations three times for each sample in an iteration. Table XIII demonstrates that our IRBA can still achieve 54.2% ASR even though each sample in the training dataset involves multiple transformations in an iteration. Additionally, incorporating multiple transformations for each sample in an iteration also increases the computational cost compared to our default choice of adaptive defenses in Table XI. Moreover, we randomly pre-select the parameters of the WLT transformation for each sample in the training dataset in advance and fix them throughout the entire adaptive defense, abbreviated as “Transformed data with fixed pre-selected parameters”. We conduct this experiment over three times and report the average results in Table XIII. Notably, a high ASR of 89.1% is achieved due to the difference between the parameters used in poisoned samples and those in adaptive defense. We assume that victims cannot obtain the specific attack parameters in poisoned samples. Consequently, our default choice of adaptive defenses in Table XI involves randomly selecting parameters from a given range for each sample during training. This approach can more effectively alleviate the attack compared to using pre-selected fixed attack parameters in “Transformed data with fixed pre-selected parameters”, as shown in Table XIII. This means that there are many possible combinations of parameter values that can be used in the transformation, making it difficult to find the exact combination that was used to perform the IRBA backdoor attack.
Fig. 11. Example digital meshes, their physical ones achieved by 3D printing technique and their scanned point clouds.

Fig. 12. (a) JG MAKER Magic 3D printer and (b) KSCAN Magic 3D scanner we used in the physical attack.

VIII. PHYSICAL ATTACK

In terms of implementation in the physical world, the attacker can utilize the LIDAR-based spoofing attacks [3], [32] or 3D printing technique [11], [34], [38] to finish our attack. Specially, 3D printing technique is widely applied in the physical attack for the imperceptibility and we also achieve our proposed IRBA physical backdoor attack against PointNet++ on ModelNet10. Example digital meshes, their physical ones achieved by 3D printing technique and their scanned point clouds are shown in Fig. 11. The 3D printer and 3D scanner we used in our physical attack are shown in Fig. 12.

The process of the implementation of physical attack by 3D printing technique is as follows: (1) We adopt the surface reconstruction with Alpha Shapes [7] to reconstruct the digital mesh in terms of the original point cloud. (2) The physical mesh is printed by a 3D printer based on the digital mesh. (3) We scan the physical mesh to obtain the point cloud and deliver it to the backdoored model. As a result, it predicts the attacker-specified target label, which illustrates that our IRBA can be achieved in the physical world.

IX. CONCLUSION AND SOCIAL IMPACTS

In this paper, we propose the imperceptible and robust backdoor attack (IRBA) in 3D domain. IRBA transforms the point cloud centered at multiple sampled anchors and smoothly aggregates them into one poisoned sample. The dataset with a small fraction of these poisoned samples can successfully inject the backdoor behavior into the point-based model trained on it. Extensive experiments demonstrate that our backdoor not only bypasses most pre-processing techniques in point cloud but also is imperceptible to human inspection. Besides, it is difficult to defend against our proposed IRBA, due to the many parameter combinations. Moreover, our proposed IRBA attack can be implemented in the physical world using 3D printing techniques, thus demonstrating its practicality in achieving physical attacks. We hope that the proposed IRBA can serve as an effective tool to examine the vulnerability of 3D models.

As for the social impacts, the main threat of IRBA can happen in the autonomous driving applications. Suppose that the autonomous vehicle manufacturers outsource the dataset to be used and the dataset is unfortunately poisoned with a fraction of samples with the trigger by the attacker. Once the autonomous vehicle manufacturers deploy their models trained on the malicious dataset, the attacker can adopt the 3D printing technique or LIDAR-based spoofing attacks [3], [32] to mislead the autonomous driving model to make a wrong decision, which leads to serious traffic accidents.

Please note that we restrict all experiments in the laboratory environment and do not support our attack in the real scenario. The purpose of our work is to raise the awareness of the security concern in 3D community and call for 3D deep learning practitioners to pay more attention to the training data integrity and model trustworthy deployment.

REFERENCES

[1] J. Bai, B. Wu, Y. Zhang, Y. Li, Z. Li, and S.-T. Xia, “Targeted attack against deep neural networks via flipping limited weight bits,” in Proc. ICLR, 2021, pp. 1–19.
[2] T. Blanc, M. El Beheiry, C. Caporal, J.-B. Masson, and B. Hajj, “Genuage: Visualize and analyze multidimensional single-molecule point cloud data in virtual reality,” Nature Methods, vol. 17, no. 11, pp. 1100–1102, Nov. 2020.
[3] Y. Cao et al., “Adversarial sensor attack on LiDAR-based perception in autonomous driving,” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., Nov. 2019, pp. 2267–2281.

[4] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, “Multi-view 3D object detection network for autonomous driving,” in Proc. CVPR, Jul. 2017, pp. 1907–1915.

[5] K. Doan, Y. Lao, and P. Li, “Backdoor attack with imperceptible input and latent modification,” in Proc. NeurIPS, 2021, pp. 18944–18957.

[6] X. Dong, D. Chen, H. Zhou, G. Hua, W. Zhang, and N. Yu, “Self-robust 3D point recognition via gather-vector guidance,” in Proc. CVPR, Jun. 2020, pp. 11151–11157.

[7] H. Edelsbrunner and E. P. Mücke, “Three-dimensional alpha shapes,” ACM Trans. Graph., vol. 13, no. 1, pp. 43–72, Jan. 1994.

[8] T. Gu, B. Dolan-Gavitt, and S. Garg, “BadNets: Identifying vulnerabilities in the machine learning model supply chain,” IEEE Access, vol. 7, pp. 47230–47244, 2019.

[9] J. Guo, P. V. K. Borges, C. Park, and A. Gavel, “Local descriptor for robust place recognition using LiDAR intensity,” IEEE Robot. Autom. Lett., vol. 4, no. 2, pp. 1470–1477, Apr. 2019.

[10] K. Huang, Y. Li, B. Wu, Z. Qin, and K. Ren, “Backdoor defense via decoupling the training process,” in Proc. ICLR, 2022, pp. 1–25.

[11] Q. Huang, X. Dong, D. Chen, H. Zhou, W. Zhang, and N. Yu, “Shape-invariant 3D adversarial point clouds,” in Proc. CVPR, Jun. 2022, pp. 15335–15344.

[12] S. Kim, S. Lee, D. Hwang, J. Lee, S. J. Hwang, and H. J. Kim, “Point cloud augmentation with weighted local transformations,” in Proc. ICCV, Oct. 2021, pp. 548–557.

[13] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proc. ICLR, 2015, pp. 1–15.

[14] L. Landrieu and M. Simonovsky, “Large-scale point cloud semantic segmentation with superpoint graphs,” in Proc. CVPR, Jun. 2018, pp. 4558–4567.

[15] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, “PointPillars: Fast encoders for object detection from point clouds,” in Proc. CVPR, Jun. 2019, pp. 12697–12705.

[16] L. Li et al., “PointBA: Towards backdoor attacks in 3D point cloud,” in Proc. ICCV, Oct. 2021, pp. 16492–16501.

[17] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, “PointCNN: Convolution on A*-transformed points,” in Proc. NeurIPS, 2018, pp. 1–11.

[18] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia, “Backdoor learning: A survey,” 2020, arXiv:2007.08745.

[19] Y. Li et al., “Deep learning for LiDAR point clouds in autonomous driving: A review,” IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 8, pp. 3412–3432, Aug. 2021.

[20] R. Liu, R. Yu, and H. Su, “Extending adversarial attacks and defenses to deep 3D point cloud classifiers,” in Proc. ICIP, Sep. 2019, pp. 2279–2283.

[21] H. Liu, J. Jia, and N. Z. Gong, “PointGuard: Provably robust 3D point cloud classification,” in Proc. CVPR, Jun. 2021, pp. 6186–6195.

[22] M. Liu, L. Sheng, S. Yang, J. Shao, and S.-M. Hu, “Morphing and sampling network for dense point cloud completion,” in Proc. AAAI, 2020, pp. 11596–11603.

[23] T. Lorenz, A. Ruoss, M. Balunović, G. Singh, and M. Vechev, “Robustness certification for point cloud models,” in Proc. ICLR, Oct. 2021, pp. 7608–7618.

[24] J. Mao, W. Wang, and H. Li, “Interpolated convolutional networks for 3D point cloud understanding,” in Proc. ICCV, Oct. 2019, pp. 1578–1587.

[25] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “NeRF: Representing scenes as neural radiance fields for view synthesis,” in Proc. ECCV, 2020, pp. 405–421.

[26] T. A. Nguyen and A. Tran, “Input-aware dynamic backdoor attack,” in Proc. NeurIPS, 2020, pp. 3454–3464.

[27] A. Paszke et al., “PyTorch: An imperative style, high-performance deep learning library,” in Proc. NeurIPS, 2019, pp. 1–12.

[28] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, “PointNet: Deep learning on point sets for 3D classification and segmentation,” in Proc. CVPR, Jul. 2017, pp. 652–660.

[29] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “PointNet++: Deep hierarchical feature learning on point sets in a metric space,” in Proc. NeurIPS, 2017, pp. 1–10.

[30] C. Qin, H. You, L. Wang, C.-C. J. Kuo, and Y. Fu, “PointDAN: A multi-scale 3D domain adaption network for point cloud representation,” in Proc. NeurIPS, 2019, pp. 1–12.
Jiawang Bai received the bachelor’s degree from Jilin University, China, in 2019. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology, Tsinghua University. He was a Visiting Student with the Learning and Vision (LV) Laboratory, National University of Singapore. His research interests include foundation model, trustworthy machine learning, and computer vision.

Baoyuan Wu (Senior Member, IEEE) is currently an Associate Professor with the School of Data Science, The Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen). He has published more than 70 top-tier conference and journal papers, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, International Journal of Computer Vision, NeurIPS, ICML, CVPR, ICCV, ICLR, and AAAI. His research interests include AI security and privacy, machine learning, computer vision, and optimization. One of his papers was selected as the Best Paper Finalist of CVPR 2019. He was selected into the world’s top 2% scientist list of 2021 and 2022, released by Stanford University. He is serving as the Organizing Chair for PRCV 2022, and the Area Chair for CVPR 2024, NeurIPS 2022/2023, NeurIPS Datasets and Benchmarks Track 2023, ICLR 2022/2023/2024, ICML 2023, AAAI 2022/2024, AISTATS 2024, WACV 2024, and ICIG 2021/2023. He is serving as an Associate Editor for Neurocomputing.

Mengxi Ya received the bachelor’s degree from the Harbin Institute of Technology, Shenzhen, China, in 2021. He is currently pursuing the M.S. degree with the Shenzhen International Graduate School, Tsinghua University, Guangdong, China. His research interests include backdoor learning.

Shu-Tao Xia (Member, IEEE) received the B.S. degree in mathematics and the Ph.D. degree in applied mathematics from Nankai University, Tianjin, China, in 1992 and 1997, respectively. From March 1997 to April 1999, he was with the Research Group of Information Theory, Department of Mathematics, Nankai University. Since January 2004, he has been with the Shenzhen International Graduate School, Tsinghua University, Guangdong, China, where he is currently a Full Professor. His current research interests include coding and information theory, networking, machine learning, and AI security.