PointBA: Towards Backdoor Attacks in 3D Point Cloud

Xinke Li[1]* Zhiru Chen[1]* Yue Zhao[1]†
Zekun Tong[1] Yabang Zhao[1] Andrew Lim[1] Joey Tianyi Zhou[2]
[1]National University of Singapore [2]Institute of High Performance Computing, A*STAR

Abstract

3D deep learning has been increasingly more popular for a variety of tasks including many safety-critical applications. However, recently several works raise the security issues of 3D deep nets. Although most of these works consider adversarial attacks, we identify that backdoor attack is indeed a more serious threat to 3D deep learning systems but remains unexplored. We present the backdoor attacks in 3D with a unified framework that exploits the unique properties of 3D data and networks. In particular, we design two attack approaches: the poison-label attack and the clean-label attack. The first one is straightforward and effective in practice, while the second one is more sophisticated assuming there are certain data inspections. The attack algorithms are mainly motivated and developed by 1) the recent discovery of 3D adversarial samples which demonstrate the vulnerability of 3D deep nets under spatial transformations; 2) the proposed feature disentanglement technique that manipulates the feature of the data through optimization methods and its potential to embed a new task. Extensive experiments show the efficacy of the poison-label attack with over 95% *success rate across several 3D datasets and models, and the ability of clean-label attack against data filtering with around* 50% *success rate. Our proposed backdoor attack in 3D point cloud is expected to perform as a baseline for improving the robustness of 3D deep models.*

1. Introduction

3D deep learning has been developed rapidly in the past few years, which makes it the prime option for various real-world deployment, such as autonomous driving [7], scene reconstruction [26] and medical data analysis [35], in which life safety issues are usually involved. As more and more attentions have been paid to this field, recently researchers have started to acknowledge and account for the security problems of 3D deep learning systems. For example, a few works have investigated the adversarial attack in the 3D domain [47, 45, 52].

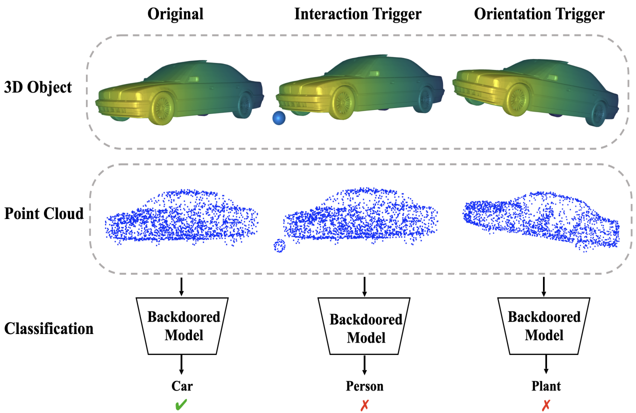


Figure 1. Activation of backdoored models with the interaction trigger and orientation trigger. Original point cloud data will be classified correctly, however with a certain trigger like an interaction object (e.g., a small ball) nearby or a small change of orientation (rotation perpendicular to the horizontal plane), the point cloud data will be classified as the target label.

Compared to the adversarial attack, an insidious threat to deep learning system called *backdoor attack*, or *Trojan attack* [24, 6, 11] is even more damaging. The backdoor attack injects a small proportion of poisoned data in training and activates malicious functionality by implanting a specified *trigger* to the test data during inference. This attack could happen when using a publicly available dataset or a pre-trained model that is potentially from untrustworthy sources. Indeed, it is reported that industry practitioners worry about data poisoning much more than other threats such as adversarial attack [16].

The backdoor attack in 3D could be a huge potential threat. On one hand, real-world 3D point cloud data usually comes with the noise caused by the nature of the collection process, e.g., the objects might be partially-occluded, spatially-distorted or noise-dense. Therefore, it is easy for the adversary to activate malicious functionality with triggers in the disguise of noise. On the other hand, due to the coordinate-based representation and the necessarily sampling before processing, 3D point cloud is often very sparse and irregular which may add to the difficulty of data in-

*Equal contribution.
†Corresponding author (email: yuezhao@u.nus.edu).

tegrity verification or even human inspection.

To the best of our knowledge, there is no work regarding the backdoor attacks in 3D. It is also non-trivial to extend the existing backdoor attack methods to 3D deep learning. We highlight the difficulties and challenges as follows: 1) the data structure of 3D point could is intrinsically different from image structure, thus, the design of the backdoor triggers for the pixel-based image can not be directly applied to 3D data; 2) models to process 3D point cloud have completely different structures compared to 2D networks, thus having many unique properties which lead to issues in designing backdoor attacks and more properties might be revealed when doing so.

To address the above challenges, we propose a framework to investigate backdoor attacks in 3D domain and hope it could be a baseline for further studies. We first introduce a unified form of the backdoor trigger implanting function for the 3D point cloud. Following the framework, we then illustrate two examples of 3D triggers motivated by real-world scenarios, namely the orientation trigger and interaction trigger. Fig. 1 shows the visualization. A perturbation analysis of the proposed trigger is then provided. It helps restrict the perturbation caused by the trigger so that the attack will be more reasonable and harder to defend against.

Based on the proposed trigger implanting function, we first propose an attack algorithm called poison-label attack. Though being straightforward, it achieves a high attack success rate across various datasets and models on both backdoor triggers. Moreover, it remains very effective even when the location and size are changing dynamically for interaction trigger or the rotation angle is relatively small for orientation trigger. To further alleviate the security concerns, we propose the clean-label attack algorithm which utilizes the recent development of 3D adversarial attacks and the *feature disentanglement* technique. Compared to poison-label attacks, the clean-label attack is a much stealthier attack that does not change labels, thus can bypass the label inspection or data filtering [40].

To summarize, our contributions are three-fold:

- To the best of our knowledge, this is the first work considering the backdoor attacks in the 3D domain. Motivated by the unique properties of 3D data, we propose a unified framework to investigate backdoor attacks in 3D as well as the perturbation analysis to restrict the attack ability in a reasonable way.
- We propose the poisoning-label attack, which is straightforward to implement and effective in practice. The attack further reveals the vulnerability of 3D models under spatial transformations and the possibility of backdoor attacks in the 3D domain.
- Inspired by the rotation-based 3D adversarial attacks, we develop a technique called feature disentanglement which crafts perturbed data through optimization methods. It makes the model relatively hard to learn the semantic label from the feature-disentangled data but easy to learn a correlation between the label and the implanted trigger. Equipped with this technique, we design a clean-label attack. It is stealthier than the poisoning-label attack and has broader application scenarios.

2. Related Work

Deep Learning on 3D Point Cloud. To learn about unordered and irregular 3d point clouds directly, existing researches proposed numerous deep models based on various operations. PointNet [27] introduced a simple max-pooling based architecture with the order-invariant property. The following work PointNet++ [28] and other extensive work [8, 50, 20] improved the feature extraction through the hierarchical stack of local feature extractors. Some other works focus on either conducting special convolutions on the 3D domain [1, 3, 37, 21, 17] or constructing graph architectures of the point clouds [43, 34, 32, 48]. In this work, we propose the backdoor attack against the PointNet, PointNet++, PointCNN [17] which is a popular convolution-based model, and DGCNN [43] exploiting the dynamic graph structure of point cloud since they are extensively involved with practical 3D applications, like 3D object detection [33, 49], scene understanding [42, 48] and 3D reconstruction[14].

Backdoor Attacks in 2D. [24] is probably one of the earliest works considering the backdoor attacks in neural networks, which is referred to as injecting Neural Trojans. The authors mainly conduct experiments on the MNIST dataset and show the possibility of embedding hidden malicious functionality in the neural network. [6] considers attacks with the targeted label in face recognition tasks and proposes several novel patterns of designing triggers. [11] explores the properties of the so-called BadNet, a backdoored neural network, and demonstrates its behaviour across several datasets including a realistic scenario of US street signs. Among the tremendous literature, clean-label (label-consistent) [41, 51] attacks do not change the label of the poisoned data, thus requires more careful craft and have more stealthiness. Other types of attacks assume other abilities of the adversary, such as manipulating the network structure [29, 36] or modifying the model-training code [2]. Interested readers are referred to recent surveys [18, 23] for a more comprehensive understanding. Although backdoor attacks have been well explored in image domains, few studies have considered the situations in 3D, let alone the efficient detection or defence against it.

Adversarial Attack on 3D Deep Models. The adversarial attacks against 3D deep models can be categorized into two main types by the operations, point adding / dropping attack and point transformation attack: 1) *Drop / Add Points.* Compared to fixed-size 2D images, dropping and adding points are specific operations to do an adversarial attack on 3d point clouds. [45] and [53] each proposes methods for identifying critical points from point clouds that affect the classification results, which can be thrown away as an attack. [47] is the first to use point generation as an adversarial perturbation. Also, [39] presents a physically achievable method of placing an adversarial mesh on the vehicle roof so that the vehicle point cloud becomes invisible to the detectors; 2) *Point Transformation.* As the transformation of the local points, the point-wise translation attack can be conducted similarly to the pixel perturbation attack in 2D images. [47] adopts C&W framework [4] based on the Chamfer and Hausdorff distance. [44, 38] further improves the objective function with consideration of the benign distribution of points. Further works [19, 13, 25] apply the iterative gradient method to achieve adversarial perturbation of points and later two of them demonstrate the resistance to defence proposed in [54]. Besides local transformation, [52] demonstrates the vulnerability of main-stream point-based models under global isometric transformation.

3. Problem Formulation

Consider a point cloud classification task, let $f_{\boldsymbol{\theta}} : \mathcal{X} \rightarrow \mathcal{Y}$ denotes a classifier of 3D deep model parameterized by $\boldsymbol{\theta}$, where $\mathcal{X} \subseteq \mathbb{R}^{n\times 3}$ is the 3D point cloud domain subset. Each point cloud consists of n points with their Euclidean coordinates in $\mathbb{R}^3$. $\mathcal{Y} \subseteq [K] := \{1, 2, \cdots, K\}$ is the set of class labels in classification task. In backdoor attacks, the key components are a *trigger implanting function* $\mathcal{G} : \mathcal{X} \rightarrow \mathcal{X}$ which implants the backdoor patterns into the standard data [18] and *trigger activation function* $\mathcal{A} : \mathcal{X} \rightarrow \mathcal{X}$, which usually coincides with $\mathcal{G}$. The goal of backdoor attack is to obtain a backdoored model $f_{\boldsymbol{\theta}'}$ that classifies any $\mathcal{A}(\boldsymbol{x})$ as the target label t and performs normally on standard data. Let a sample be $\boldsymbol{z} := (\boldsymbol{x}, y) \in \mathcal{Z} := \mathcal{X} \times \mathcal{Y}$, $\mathbb{P}_N = \{\boldsymbol{z}_1, \cdots, \boldsymbol{z}_N\}$ be the training set and $\mathbb{P}$ be the underlying distribution. Given a loss function $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_+$, the standard training process by Empirical Risk Minimization (ERM) is to:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z}\sim\mathbb{P}_N}\Big[\mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y)\Big]. \tag{1}$$

To conduct backdoor attack, we replace ϵ proportion of $\mathbb{P}_N$ by generating samples $\mathbb{P}_p = \{(\mathcal{G}(\boldsymbol{x}), t) | (\boldsymbol{x}, y) \sim \mathbb{P}_N\}$, i.e., $\mathbb{P}'_N = (1 - \epsilon)\mathbb{P}_N \cup \epsilon\mathbb{P}_p$. A backdoored model is simply trained by replacing $\mathbb{P}_N$ with $\mathbb{P}'_N$ in Eq.(1). Then in the inference phase, it is expected that the prediction error

$$\mathbb{E}_{\boldsymbol{z}\sim\mathbb{P}}\Big[\mathbb{I}\{f_{\boldsymbol{\theta}'}(\boldsymbol{x}) \neq y\}\Big], \tag{2}$$

is small enough and backdoor attack success rate

$$\mathbb{E}_{\boldsymbol{z}\sim\mathbb{P}}\Big[\mathbb{I}\{f_{\boldsymbol{\theta}'}(\mathcal{A}(\boldsymbol{x})) = t\}\Big], \tag{3}$$

is large enough, where $\boldsymbol{\theta}'$ is the parameters of the backdoored model.

Note that the backdoor attack is fundamentally different from the adversarial attack since the adversarial attack is conducted in the inference phase while the backdoor attack is done in the training process. The backdoor attack is also quite different from the data poisoning attack which is usually designed to undermine the overall performance of the model and there is no trigger. In contrast, the backdoor attack produces a backdoored model which is normal if tested on standard data but hidden (malicious) functionality can be activated by a pre-defined trigger, thus it is stealthier and more harmful than the data poisoning attack.

4. A Unified Form of 3D Backdoor Triggers

The trigger implanting function in the 2D domain is usually by adding a pre-defined patch to the original image with pixel value, e.g., [11, 22, 31]. However, since the 3D point cloud is based on coordinate representation, it is difficult to directly adapt the patch-based backdoor patterns from 2D images. Therefore, we investigate the unique transformation on the 3D point cloud as a 3D backdoor trigger. Motivated by several transformation-based adversarial attacks in the 3D point cloud [52, 53], we present a unified form of 3D trigger implanting function as following

$$\mathcal{G}(\boldsymbol{x}) = (\boldsymbol{I} - \mathrm{Diag}(\boldsymbol{\delta}))\boldsymbol{x}\boldsymbol{A} + \mathrm{Diag}(\boldsymbol{\delta})\boldsymbol{B}, \tag{4}$$

where $\boldsymbol{x}$ is a point cloud, $\boldsymbol{A} \in \mathbb{R}^{3\times 3}$ is a spatial transformation matrix and $\boldsymbol{B} \in \mathcal{X} \subseteq \mathbb{R}^{n\times 3}$ represents an additive point cloud, $\boldsymbol{\delta} \in \mathbb{R}^n$ is a vector with either 0 or 1, $\boldsymbol{I}$ is the identity matrix in $\mathbb{R}^{n\times n}$. $\boldsymbol{A}$ and $\boldsymbol{B}$ are delicately designed to achieve to goal of the backdoor attacks, which will be elaborated in Sec. 5 in detail.

4.1. Designing of 3D Trigger Implanting

We demonstrate two examples of 3D backdoor triggers based on Eq. (4). Motivations of these two triggers from real-world scenarios are presented with the illustration.

Orientation Trigger. The *orientation* is referred to a particular rotation transformation of the object. Considering that the 3D point could data is usually either properly aligned or provided with orientation annotations in 3D datasets [46, 10], we utilize a specific orientation of aligned 3D objects as the backdoor trigger. The trigger Implanting function in Eq. (4) becomes

$$\mathcal{G}_{\text{or}}(\boldsymbol{x}) = \boldsymbol{x}\boldsymbol{A}, \tag{5}$$

where $\boldsymbol{A}$ is a rotation matrix. In our attack, the Euler angles related to rotation matrix $\boldsymbol{A}$ are limited to small ranges. The motivation behind the design is that the objects detected in real scenes are often in diverse poses with respect to the sensor or distorted by spatial transformation, the installed backdoor is to make the model predict target label for an object with a certain pose, e.g., a car tilted slightly upwards will be classified as a plant.

Interaction Trigger. The *interaction* occurs when an object of interest is physically close to another interaction object. This phenomenon often comes with the data collection in the real scene, which could potentially be made use of to undermine the security of the 3D deep learning system. For example, [39] designs an optimized object on the top of the vehicle fooling the 3D detection deep model. Formally, the interaction transformation is

$$\mathcal{G}_{\text{it}}(\boldsymbol{x}) = (\boldsymbol{I} - \text{Diag}(\boldsymbol{\delta}))\boldsymbol{x} + \text{Diag}(\boldsymbol{\delta})\boldsymbol{B}, \qquad (6)$$

where $\boldsymbol{B}$ represents the interaction object. It is simply derived by setting $\boldsymbol{A} = \boldsymbol{I}$ in Eq. (4). Unlike the meticulously designed shape in [39], the integration object as a backdoor trigger can be designed as commonly seen shapes, like balls shown in Fig. 1, which could even be degenerated to one point. In practice, $\boldsymbol{\delta}$ is a random binary vector with $\|\boldsymbol{\delta}\|_1 \leq 0.05n$.

4.2. Perturbation Analysis of Trigger Implanting

We hope the poisoned data does not deviate from the standard data too far so that it will not be detected easily. Thus we provide a bound on the deviation regarding the proposed function $\mathcal{G}$. Define a matrix norm $\|\boldsymbol{x}\| = \sum_{j=1}^{n} \|\boldsymbol{x}_j\|_2$, where $\boldsymbol{x} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n]^\top$. This norm is just the sum of ℓ_2 norm of the row vectors. Let $\boldsymbol{B} = [\boldsymbol{B}_1, \cdots, \boldsymbol{B}_n]^\top$. Suppose the point-wise distance between the interaction object and the transformed sample is bounded by r , that is, $\|\boldsymbol{B}_j - \boldsymbol{A}^\top \boldsymbol{x}_j\|_2 \leq r$ for $j = 1, \cdots, n$, then

$$\begin{aligned}
\|\mathcal{G}(\boldsymbol{x}) - \boldsymbol{x}\| &= \|(\boldsymbol{I} - \text{Diag}(\boldsymbol{\delta}))\boldsymbol{x}\boldsymbol{A} + \text{Diag}(\boldsymbol{\delta})\boldsymbol{B} - \boldsymbol{x}\| \\
&= \|\boldsymbol{x}(\boldsymbol{A} - \boldsymbol{I}) + \text{Diag}(\boldsymbol{\delta})(\boldsymbol{B} - \boldsymbol{x}\boldsymbol{A})\| \\
&\leq \|\boldsymbol{x}(\boldsymbol{A} - \boldsymbol{I})\| + \|\text{Diag}(\boldsymbol{\delta})(\boldsymbol{B} - \boldsymbol{x}\boldsymbol{A})\| \\
&= \sum \|(\boldsymbol{A} - \boldsymbol{I})^\top \boldsymbol{x}_j\|_2 + \sum_{\delta_j \neq 0} \|\boldsymbol{B}_j - \boldsymbol{A}^\top \boldsymbol{x}_j\|_2 \\
&\leq \sum \sigma(\boldsymbol{A}^\top - \boldsymbol{I})\|\boldsymbol{x}_j\|_2 + r\|\boldsymbol{\delta}\|_1 \\
&= \sigma(\boldsymbol{A}^\top - \boldsymbol{I})\|\boldsymbol{x}\| + r\|\boldsymbol{\delta}\|_1,
\end{aligned} \qquad (7)$$

where $\|\cdot\|_1$ is ℓ_1 norm, $\sigma(\boldsymbol{A}^\top - \boldsymbol{I})$ is the spectral norm of $\boldsymbol{A}^\top - \boldsymbol{I}$. Hence, to control the perturbation, we just need three components: 1) the transformation matrix $\boldsymbol{A}$; 2) the number of points of the interaction object $\|\boldsymbol{\delta}\|_1$; 3) the upper bound r of the point-wise distance between the additive point cloud and the original points.

5. Attack Methods

To evaluate the validity of the proposed 3D backdoor triggers, in this section, we propose two attack schemes, namely the *poison-label (PL) backdoor* and the *clean-label (CL) backdoor attack*. The attacks both assume that the adversary has access to the training data and the backdoor can be installed through the training process of models. The first attack method is a straightforward approach by altering both the data and labels on training data, which is consistent with existing literature [11, 6]. To further explore the ability of our proposed trigger and make the poisoned data less distinguishable, we design a clean-label attack based on the enhanced trigger implanting. It requires a pre-trained 3D model while the trigger is implanted on the feature-disentangled samples and the label of the poisoned data is consistent with the original data.

We consider two threat models for each attack respectively and identify their differences in settings.

PL Threat Model. The adversary can launch the attack by implanting a fraction of the training dataset with designed triggers. During the inference time, the adversary can maliciously make the model predict the target label by adding the trigger to input (i.e., the trigger is activated), while the prediction on clean input will be normal.

CL Threat Model. Assuming the user may conduct label inspection or re-annotation before training, the clean-label attack only allows the data with target labels to be modified by the adversary in the attack. Therefore, changing labels is excluded from this poisoning process. On the other hand, a model pre-trained on standard data is required to generate poisoned data. Different from 2D clean-label attack conducted by fine-tuning the pre-trained networks [15, 30], we retrain the 3D deep models from scratch on the poisoned dataset.

5.1. Poison-label Backdoor Attack

Our poison-label attack follows the conventional routine in 2D backdoor attacks: first, we inject a small rate of poisoned data with the proposed trigger implanting function to the training data, and then user train the backdoored model using standard approach; in inference, we test on data with trigger activation function. The injection rate ϵ is usually a small number, e.g., $\epsilon = 0.05$.

Different forms of triggers. The PL attack is proposed to directly demonstrate the effectiveness of interaction and orientation triggers. The triggers are both set to not perturb the original data much according to Eq. (7):

- For orientation trigger, we set $\text{Diag}(\boldsymbol{\delta}) = \boldsymbol{0}, \boldsymbol{B} = \boldsymbol{0}$ and represent $\boldsymbol{A}$ by rotation matrix alone z-axis with the corresponding Euler angle is $(0, 0, \omega_z)$. Although it can be any rotation matrices, we experimentally present that a small rotation (e.g. $5°$) alone z-axis is sufficient for a successful attack and has few effects on the original data.
- For interaction trigger, we set $\boldsymbol{A} = \boldsymbol{0}$ and the sampled point number $\|\boldsymbol{\delta}\|_1$ of interaction object $\boldsymbol{B}$ is related to its size. We design the object $\boldsymbol{B}$ to be small and close to the original object, which keeps the ratio between the interaction trigger points and total points $\frac{\|\boldsymbol{\delta}\|_1}{n} \leq 0.05$.

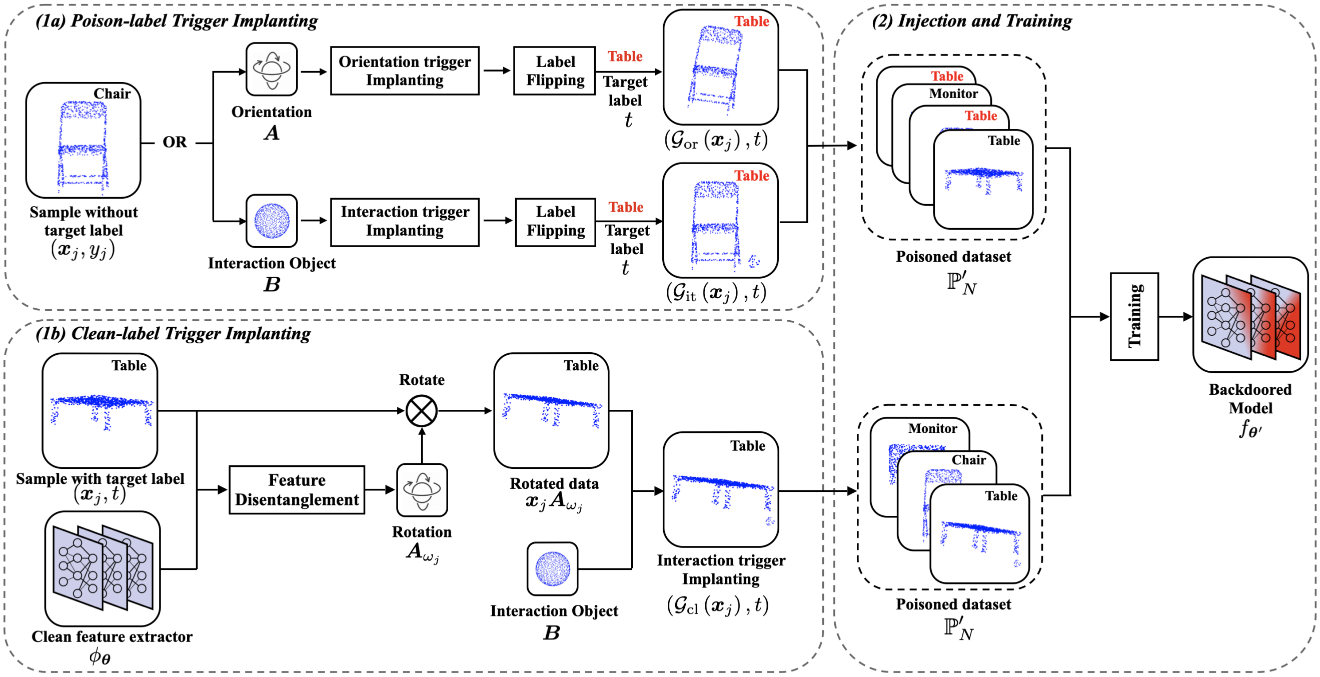

(1a) Poison-label Trigger Implanting
Chair
Sample without target label
$(\boldsymbol{x}_j, y_j)$
OR
Orientation
$\boldsymbol{A}$
Orientation trigger Implanting
Label Flipping
Table
Target label
t
Table
$(\mathcal{G}_{\text{or}}(\boldsymbol{x}_j), t)$
Interaction Object
$\boldsymbol{B}$
Interaction trigger Implanting
Label Flipping
Table
Target label
t
Table
$(\mathcal{G}_{\text{it}}(\boldsymbol{x}_j), t)$
(2) Injection and Training
Table
Monitor
Table
Table
Poisoned dataset
$\mathbb{P}'_N$
Training
Backdoored Model
$f_{\boldsymbol{\theta}'}$
(1b) Clean-label Trigger Implanting
Table
Sample with target label
$(\boldsymbol{x}_j, t)$
Clean feature extractor
$\phi_{\boldsymbol{\theta}}$
Feature Disentanglement
Rotate
Rotation
$\boldsymbol{A}_{\omega_j}$
Table
Rotated data
$\boldsymbol{x}_j \boldsymbol{A}_{\omega_j}$
Interaction Object
$\boldsymbol{B}$
Table
Interaction trigger Implanting
$(\mathcal{G}_{\text{cl}}(\boldsymbol{x}_j), t)$
Monitor
Chair
Table
Poisoned dataset
$\mathbb{P}'_N$

Figure 2. Attack Pipeline of Proposed 3D Backdoor Attack.

5.2. Clean-label Backdoor Attack

In order to bypass the label inspection or data filtering by the user, we propose a clean-label (CL) attack on 3D deep models that avoids the label altering procedure. The feature disentanglement technique is first introduced to enhance the trigger implanting, which is the core gradient to the design of CL attack. Then the detailed attack pipeline is illustrated.

Feature disentanglement. Given a sample x in the target class $\mathbb{P}_t$ which consists of all the data with label t and learnt feature representation $\phi_{\boldsymbol{\theta}}$, the *feature disentanglement* is to find a perturbed version $c(x)$ by solving the following optimization problem

$$\begin{aligned} \max_{c} \quad & \sum_{\boldsymbol{x} \in \mathbb{P}_t \setminus \{\boldsymbol{x}\}} \mathcal{D}(\phi_{\boldsymbol{\theta}}(c(\boldsymbol{x})), \phi_{\boldsymbol{\theta}}(\boldsymbol{x})) \\ \text{s.t.} \quad & \mathcal{D}'(c(\boldsymbol{x}), \boldsymbol{x}) \leq r \end{aligned}, \tag{8}$$

where $\mathcal{D}$ is a distance metric in feature space $\mathbb{R}^d$, $\mathcal{D}'$ is a distance metric in $\mathbb{R}^3$, r is the distance parameter to restrict the perturbation. The formulation is indeed an optimization problem over functional space, in practice, we usually parameterize the function $c(\cdot)$ so that the problem can be efficiently solved.

Considering the more challenging setting of CL attack, we propose a rotation-based feature disentanglement to enhance the interaction trigger implanting by exploiting the full functionality of Eq.(4). Suppose the rotation matrix $\boldsymbol{A}_{\boldsymbol{\omega}_j}$ is parameterized by the Euler angle $\boldsymbol{\omega}_j$, the disentanglement is defined as $c(\boldsymbol{x}_j) = \boldsymbol{x}_j \boldsymbol{A}_{\boldsymbol{\omega}_j}$. The motivation is that spatial transformation could effectively perturb the extracted features of 3D deep models [52], thus it can be utilized as feature disentanglement to pull the main features away from the others in the target class. The trigger implanting function can be expressed as

$$\mathcal{G}_{\text{cl}}(\boldsymbol{x}_j) = (\boldsymbol{I} - \text{Diag}(\boldsymbol{\delta}))\boldsymbol{x}_j \boldsymbol{A}_{\boldsymbol{\omega}_j} + \text{Diag}(\boldsymbol{\delta})\boldsymbol{B}, \tag{9}$$

for one data $\boldsymbol{x}_j$ from the target label. The motivation behind this enhancement is that: 1) it is hard for the model to learn useful information from the feature disentangled data; 2) any trigger pattern implanted in the feature disentangled data will be a dominant feature, in another word, the model indeed will learn a *hidden new task* that connects the trigger with the target class.

Algorithm 1: Clean-label Backdoor Attack

```
Input: A model structure $f$, training set $\mathbb{P}_N$, injection rate $\epsilon$,
    a sample vector $\boldsymbol{\delta}$, an interaction shape $\boldsymbol{B}$, a target label $t$.
Output: Backdoored model $f_{\boldsymbol{\theta}'}$
Collect all the data in $\mathbb{P}_N$ with label $t$, denote them as $\mathbb{P}_t$
Random sample $\lfloor \epsilon N \rfloor$ data $\{\boldsymbol{z}_1, \cdots, \boldsymbol{z}_J\}$ in $\mathbb{P}_t$, denote them as $\tilde{\mathbb{P}}$
$\mathbb{P}'_N \leftarrow \mathbb{P}_N \setminus \tilde{\mathbb{P}}$
Train a clean model with structure $f$ using $\mathbb{P}_N$, obtain its feature
    representation function $\phi_{\boldsymbol{\theta}}$
for $j = 1$ to $J$ do
    Given $\boldsymbol{x}_j$ and $\phi_{\boldsymbol{\theta}}$, find $\omega_j$ by solving problem (10)
    Set $\boldsymbol{x}'_j \leftarrow (\boldsymbol{I} - \text{Diag}(\boldsymbol{\delta}))\boldsymbol{x}_j \boldsymbol{A}_{\boldsymbol{\omega}_j} + \text{Diag}(\boldsymbol{\delta})\boldsymbol{B}$ from Eq. (9)
    Set $\mathbb{P}'_N \leftarrow \mathbb{P}'_N \cup (\boldsymbol{x}'_j, y_j)$
end for
Train a model of structure $f$ on dataset $\mathbb{P}'_N$ and obtain $f_{\boldsymbol{\theta}'}$
```

Attack Pipeline. The pipeline of the CL attack is illustrated in Fig. 2 and Alg. 1. Given a sample $\boldsymbol{x}_j$ in the target class $\mathbb{P}_t$ which consists of all the data with label t, the attack is briefly described as followed: 1) first we use the feature disentanglement loss to find a rotation $\boldsymbol{A}_{\boldsymbol{\omega}_j}$ to transform the sample data, and 2) then add an interaction pattern to that sample with the correct label, say 'Car', but indeed the data in feature space is hardly recognized as the 'Car'. Finally, in the inference phase, any data with the interaction trigger will be misclassified as 'Car' since the trained model has connected the target label with the trigger. Note that in this attack, $\boldsymbol{A}_{\boldsymbol{\omega}_j}$ depends on sample $\boldsymbol{x}_j$, and we only use $\boldsymbol{B}$ in the inference phase, which is slightly different from the previous setting. Here we formulate the problem formally, let $\phi_{\boldsymbol{\theta}} : \mathcal{X} \to \mathbb{R}^d$ be the pre-trained feature extractor. Given a sample $\boldsymbol{x}_j$, we have the following optimization problem as a special case of (8) to find such ω_j

$$\begin{aligned} \max_{\boldsymbol{\omega}_j} \quad & \sum_{\boldsymbol{x}_i \in \mathbb{P}_t \setminus \{\boldsymbol{x}_j\}} \mathcal{D}(\phi_{\boldsymbol{\theta}}(\boldsymbol{x}_j \boldsymbol{A}_{\boldsymbol{\omega}_j}), \phi_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) \\ \text{s.t.} \quad & \boldsymbol{\omega}_j \in \mathcal{R} \end{aligned} , \tag{10}$$

where $\mathcal{D}$ is a distance metric in feature space $\mathbb{R}^d$ and $\mathcal{R} \subseteq \mathbb{R}^3$ is a range to restrict the rotation magnitude. The above optimization problem is non-convex, thus we utilize the global optimization method to find a decent angle of $\boldsymbol{\omega}_j$. We apply Bayesian Optimization (BO) [9] and transfer the Euler angle to axis-angle representation for practicality, of which the details are in the supplementary material.

6. Experiments

6.1. Dataset and Models

To effectively illustrate our proposed 3D backdoor attack, we conduct our experiments on the shape recognition tasks. The utilized datasets are the commonly-used ModelNet10 [46], ModelNet40 [46] and ShapeNetPart [5]. For ModelNet40, we use the official split of 9843 point clouds for training and 2468 for the attack. The ModelNet10 downsampled from ModelNet40 contains 10 categories. The ShapeNetPart with 16 categories is a part of ShapeNet, which contains 12128 and 2874 objects for training and test set respectively. For the fix-size input of 3D deep models, we uniformly sample 1024 points from the original mesh models in the datasets as the procedure of [27] and normalize them into $[0, 1]^3$. For the implementation of the deep models, we use four widely-used 3D deep classifiers: PointNet [27], PointNet++ [28], PointCNN [17] and DGCNN [43], in short term as PN, PN++, PCNN and DGCNN respectively.

6.2. Attack Setting

Target Label. We select the target class randomly in the dataset categories, which are 'Table' in ModelNet10, 'Toilet' in ModelNet40 and 'Lamp' in ShapeNetPart. The sample to be poisoned are also randomly sampled from the non-target class and target class for the poison-label attack and the clean-label attack respectively.

Orientation Trigger. We apply rotation transformation associated with Euler angle $(0, 0, \omega_z)$ with respect to the aligned orientation as our trigger.

Interaction Trigger. For the sake of simplicity, we choose a sphere with a fixed radius and centre as our interaction object $\boldsymbol{B}$. The interaction object $\boldsymbol{B}$ can be scaled and shifted in trigger implanting by $\boldsymbol{B}' \leftarrow \alpha \boldsymbol{B} + \boldsymbol{\beta}$, where $\alpha \in \mathbb{R}^+$ and $\boldsymbol{\beta} \in [0, 1]^3$. In experiments, the parameters are randomly sampled from uniform distributions $\alpha \sim \mathcal{U}(1 - \lambda_\alpha, 1 + \lambda_\alpha)$ and $\boldsymbol{\beta} \sim \mathcal{U}(-\lambda_{\boldsymbol{\beta}}, \lambda_{\boldsymbol{\beta}})$, where λ_α and $\lambda_{\boldsymbol{\beta}}$ are set to control the randomness. In terms of the enhanced trigger in clean-label attack, the angles of axis-angle-based rotation matrices for feature disentanglement is limited to $[0, \omega_{max}]$. We set a small ω_{max} to guarantee the unnoticeable perturbations of data.

Metric. To evaluate the effectiveness of our proposed backdoor attack, the attack success rate (ASR) defined in Eq. (3) is set as the measurement. Empirically, the ASR is calculated on the test set of the poisoned dataset.

DNN Training. All DNN models are trained using Adam optimizer with a learning rate of 0.001. We use batch size 32 and train all models for 200 epochs. All experiments are run on a GeForce RTX 2080Ti GPU. We mention that the above settings are configured for model training on both the poisoned dataset and clean dataset.

6.3. Effectiveness of 3D Backdoor Attack

We conduct extensive experiments to verify the effectiveness of proposed 3D backdoor attacks via the ASR on different models, datasets and parameter settings. The attack methods are the poison-label attack on interaction trigger (PL-I), poison-label attack on orientation trigger (PL-O) and clean-label attack (CL).

ASR Comparison. The ASR and the corresponding test accuracy (ACC) of the different backdoored models on the different test sets are presented in Tab. 1. We keep injection rates equal to or less than 0.05 for both PL and CL attack. For PL-I and CL attack, the sphere trigger is with radius at 0.05, centre at $(0.05, 0.05, 0.05)$, and without random scaling and shifting. For PL-O attack, the Euler angle of trigger rotation is set as $(0°, 0°, 10°)$. The CL attack enhances interaction trigger by rotation-based transformation with the rotation angle $\omega_j \in [0°, 25°]$ in axis-angle representation.

Tab. 1 demonstrates the main results under the above setting. At injection rates of 0.05, the PL-I and PL-O attacks can both achieve an ASR of greater than 93%, while CL attacks can achieve an ASR of greater than 45% across all 3D models. At the same time, the backdoored model only suffers a maximum loss of 2% of accuracy in all clean datasets. It is rational that the CL attack has a lower ASR than the PL attack because of the more challenging problem setting.

We experiment with all categories in ModelNet10 as

Table 1. ASR (%) of our proposed 3D poison-label attack on interaction trigger (PL-I) and orientation trigger (PL-O) as well as the clean-label (CL) attack, and the backdoored model's test accuracy (%) on the clean test set. The injection rate is 0.05 for poison-label attack and less than 0.05 of total dataset in the target classes (fixed ratio of target label data) for clean-label attack.

Models	ACC/ASR(%) of PL-I Attack			ACC/ASR(%) of PL-O Attack			ACC/ASR(%) of CL Attack		
	MN10	MN40	SNPart	MN10	MN40	SNPart	MN10	MN40	SNPart
PN	89.3/99.5	85.4/99.3	**98.4/100**	90.3/95.5	85.9/93.2	98.8/99.6	88.9/82.5	84.6/56.8	97.7/48.3
PN++	91.9/97.5	89.1/98.6	98.4/99.1	91.6/95.2	89.8/94.7	98.4/94.5	91.6/53.8	**88.7/66.0**	98.0/48.6
DGCNN	**92.2/100**	**90.1/100**	98.4/99.5	92.8/94.8	**89.1/97.5**	97.7/99.9	92.8/46.8	89.4/50.9	**97.7/66.6**
PCNN	91.5/97.8	88.5/97.0	97.1/97.5	**91.5/95.8**	88.7/93.1	**98.7/100**	**91.5/63.4**	88.4/61.2	97.7/51.4

Table 2. ASR(%) of PL-I, PL-O and CL attacks for each category in the ModelNet10 as the target category. The percentages under class names are the occupation rates of the class on the whole dataset.The attacked model is PointNet++ and injection rate of CL attack is the 0.5 of the target class proportion.

Attack/Class	Bathtub	Bed	Chair	Desk	Dresser	Monitor	Nightstand	Sofa	Table	Toilet
	(2.7%)	(12.9%)	(22.2%)	(5.0%)	(5.0%)	(11.6%)	(5.0%)	(17.0%)	(9.8%)	(8.6%)
PL-I	96.6	97.2	95.8	97.5	98.5	97.7	97.7	97.8	97.5	97.0
PL-O	98.4	96.6	95.8	98.7	97.4	97.4	96.9	97.5	98.8	96.6
CL	23.1	63.0	64.5	48.7	43.5	51.2	59.6	61.2	53.8	46.2

the target class against the PointNet++ model, the results of which are shown in Tab. 2. We observe that the high ASR above 95% of the PL attack is consistent across labels, whereas the ASR of the CL attack is highly varying for different labels. This variation can be attributed to the high sensitivity of the CL attack to the injection rate since it is shown that the ASR is highly correlated with the data proportion of the target labels.

Effect of Injection Rate. We then perform the injection rate experiments with PointNet++ on ModelNet10, following the same setup as previous experiments.As shown in Fig. 3, the ASR of the CL attacks is sensitive to the injection rate and tends to increase up to ASR 54%, at the rate of 0.07. In the contrast, except for some slightly lower values at the injection rate $\epsilon \leq 0.03$, the ASR of the PL attacks remains high at over 90% for both interactions and orientation triggers. Meanwhile, with an injection rate of less than 0.1, none of the three backdoor attacks significantly degrade the performance of the victim model on clean data.

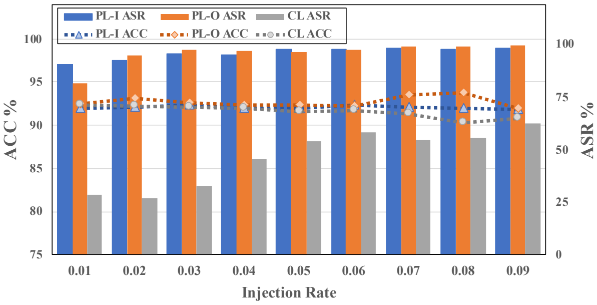
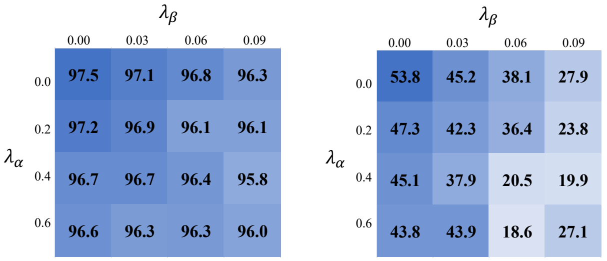


Figure 3. ASR (%) and ACC (%) against the varying injection rate on PL-I, PL-O and CL Attacks.

Effect of Trigger Parameters. We then investigate the effect of varying trigger parameters. For orientation trigger, we show in Tab. 3 that even a small angle of trigger ($\sim 5°$) can reach a high ASR, which also suggests the vulnerability of the 3D network to spatial transformations. For interaction trigger, we investigate the joint effect of randomness factors λ_α and λ_β. As shown in Fig. 4, the PL attack is shown not sensitive to dynamic triggers, suggesting the great flexibility of the trigger. For the CL attack on enhanced interaction trigger, the random shift obviously decreases the ASR down to 19.9% as λ_β increases, while the random scale does not have a significant effect.

Table 3. ASR(%) and ACC(%) of backdoored model from PL-O attack based on different rotation angles ω_z of orientation trigger.

Orientation (°)	1.25	2.5	5	10	20	40
ASR(%)	27.9	86.9	92.1	92.8	93.6	93.2
ACC(%)	92.6	92.1	93.1	92.8	92.3	92.7



λ_α \ λ_β	0.00	0.03	0.06	0.09
0.0	97.5	97.1	96.8	96.3
0.2	97.2	96.9	96.1	96.1
0.4	96.7	96.7	96.4	95.8
0.6	96.6	96.3	96.3	96.0

(a) ASR(%) of PL-I attack

λ_α \ λ_β	0.00	0.03	0.06	0.09
0.0	53.8	45.2	38.1	27.9
0.2	47.3	42.3	36.4	23.8
0.4	45.1	37.9	20.5	19.9
0.6	43.8	43.9	18.6	27.1

(b) ASR(%) of CL attack

Figure 4. ASR(%) of PL-I & CL attack against scale randomness factor λ_α and shift randomness factor λ_β of the interaction trigger.

Others. Other investigations into the triggers are also conducted and presented in the supplementary material. They are 1) The rotations along other axes for the orientation trigger. 2) Other shapes including cubes, cylinders and one point for the interaction triggers. They both present similar results to current experiments.

6.4. Study on Clean-label Attack

To better understand our CL attack, we conduct the ablation study towards the feature disentanglement method and demonstrate the effect of angle range in the method. We also highlight the transferablility of the proposed attack, which increases its generality and damage.

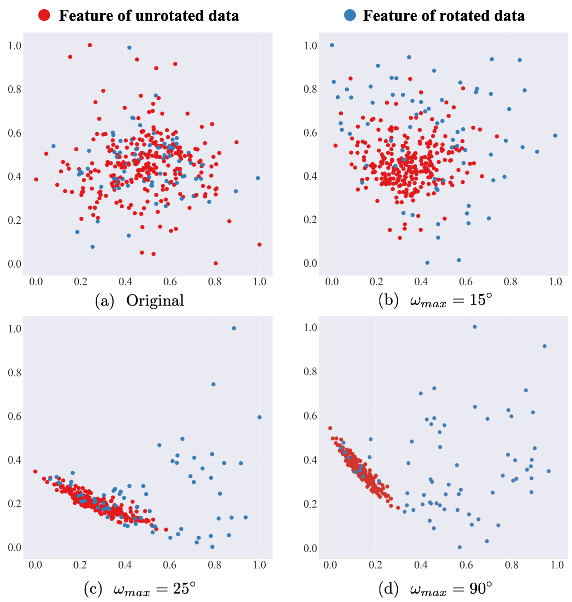
Ablation Study. Ablation studies are performed on three data processing methods for trigger enhancement. The experimental setup uses the same as above with PointNet++ architecture, the selected target label and ω_{max} at 25°. In addition to the ASR, we have also included the average loss defined in Eq. (10). There are two methods to compare with the BO method [9], one is to add triggers directly to the training data, and the other is to apply a random rotation to the data and add triggers. From Tab. 4 we find that the BO method can identify the transformation with higher losses and this rotation will lead to almost twice the success rate improvement as a random rotation in ModelNet10.

Effect of Angle Range. Feature disentanglement in clean-label attack is achieved by a rotational transformation found within the range of angle $[0, \omega_{max}]$ for angle-axis representation. We studied the effect of different ω_{max} on the feature disentanglement, which is shown in Fig. 5. It can be seen that the larger the angle range is, the further the disentangled features are from the features of the original data. However, considering that a large rotation angle according to Eq. (4) will cause a large perturbation to the original data, in our experiments we used a trade-off angle i.e. $\omega_{max} = 25°$.

Transferability. In additional experiments, we note that clean label attack has transferability across different models. The poisoned dataset obtained by PointNet++ used in CL experiments can reach the ASR 68.1% in PointNet, 38.3% in DGCNN, and 60.6 % in PointCNN. This is not surprising because of two reasons: 1) The setting of our proposed CL attack originally requires that the attack can work across models from different initializations. 2) The attack by spatial transformation is proposed to be strongly transferable [52], so the feature disentanglement based on it can have the same nature. More results of transferability experiments can be found in the supplementary.

Table 4. Ablation study of rotation-based feature disentanglement. ASR(%) and Avg Loss (in Eq. (10)) are respectively compared on ModelNet10 and ShapeNetPart under three different data processing methods. The results highlight the effectiveness of BO-optimized rotation over the others on trigger enhancement.

Data Processing Method	ModelNet10		ShapeNetPart	
	ASR	Avg Loss	ASR	Avg Loss
Without Rotation	23.8	0.66	19.0	0.80
Random Rotation	31.6	1.04	33.2	0.95
BO-optimized Rotation	**53.8**	**1.40**	**48.6**	**1.21**



(a) Original (b) $\omega_{max} = 15°$

(c) $\omega_{max} = 25°$ (d) $\omega_{max} = 90°$

Figure 5. PCA-based visualization of disentangled features with different rotation angle upper bounds ω_{max}. The features are derived from 'Table' label data from ModelNet10. The feature disentanglement is to move the features of the rotated data (blue) away from the other same labeled data (red) by rotation. Larger feature separation may better enhance the correlation between the implanted trigger and the label.

6.5. Resistance to Data Filtering

The authors of [40] propose a data filter method to successfully reject poison-label attack samples in 2D images. Inspired by their approach, we utilize a similar pipeline in 3D. We train a classifier via a small and clean dataset and then use it to filter poisoned data in the provided dataset from an untrustworthy source. We use the conventional point cloud descriptors [12] and MLP for the classifier. The result shows that the poisoned data in PL attack can be effectively removed with up to 82% filtering rate, while the CL attack can resist this defence with the rate of only 14% (See supplementary for more details).

7. Conclusion

In this paper, we have explored the backdoor attack to deep models applied on 3D point cloud. Firstly, we propose a unified framework of 3D backdoor trigger implanting function. Based on it, we design two 3D backdoor triggers and investigate the performance of widely used 3D deep models under poison-label attacks. To strengthen the concealment of the proposed trigger, we further introduce a clean-label attack by rotation-based feature disentanglement against point clouds. The experiments suggest the vulnerability of current 3D deep nets to our proposed attack and the limited effectiveness of data filtering towards attacks. It is expected that the proposed attacks can serve as a strong baseline for improving the robustness of deep models in 3D point cloud.

References

[1] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point convolutional neural networks by extension operators. *arXiv preprint arXiv:1803.10091*, 2018. 2

[2] Eugene Bagdasaryan and Vitaly Shmatikov. Blind backdoors in deep learning models. *arXiv preprint arXiv:2005.03823*, 2020. 2

[3] Alexandre Boulch. Convpoint: Continuous convolutions for point cloud processing. *Computers & Graphics*, 2020. 2

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

[5] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 6

[6] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017. 1, 2, 4

[7] 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

[8] Yueqi Duan, Yu Zheng, Jiwen Lu, Jie Zhou, and Qi Tian. Structural relational reasoning of point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 949–958, 2019. 2

[9] Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018. 6, 8

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 3

[11] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017. 1, 2, 3, 4

[12] Yulan Guo, Mohammed Bennamoun, Ferdous Sohel, Min Lu, Jianwei Wan, and Ngai Ming Kwok. A comprehensive performance evaluation of 3d local feature descriptors. *International Journal of Computer Vision*, 116(1):66–89, 2016. 8

[13] 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

[14] Xianfeng Han, Hamid Laga, and Mohammed Bennamoun. Image-based 3d object reconstruction: State-of-the-art and trends in the deep learning era. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 2

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 4

[16] Ram Shankar Siva Kumar, Magnus Nyström, John Lambert, Andrew Marshall, Mario Goertzel, Andi Comissoneru, Matt Swann, and Sharon Xia. Adversarial machine learning-industry perspectives. In *2020 IEEE Security and Privacy Workshops (SPW)*, pages 69–75. IEEE, 2020. 1

[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. 2, 6

[18] Yiming Li, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *arXiv preprint arXiv:2007.08745*, 2020. 2, 3

[19] 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. 3

[20] Yongcheng Liu, Bin Fan, Gaofeng Meng, Jiwen Lu, Shiming Xiang, and Chunhong Pan. Densepoint: Learning densely contextual representation for efficient point cloud processing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5239–5248, 2019. 2

[21] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8895–8904, 2019. 2

[22] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. *arXiv preprint arXiv:2007.02343*, 2020. 3

[23] Yuntao Liu, Ankit Mondal, Abhishek Chakraborty, Michael Zuzak, Nina Jacobsen, Daniel Xing, and Ankur Srivastava. A survey on neural trojans. *IACR Cryptol. ePrint Arch.*, 2020:201, 2020. 2

[24] Yuntao Liu, Yang Xie, and Ankur Srivastava. Neural trojans. In *2017 IEEE International Conference on Computer Design (ICCD)*, pages 45–48. IEEE, 2017. 1, 2

[25] 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. 3

[26] S Malihi, MJ Valadan Zoej, M Hahn, M Mokhtarzade, and H Arefi. 3d building reconstruction using dense photogrammetric point cloud. *Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3:71–74, 2016. 1

[27] 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. 2, 6

[28] 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. 2, 6

[29] Adnan Siraj Rakin, Zhezhi He, and Deliang Fan. Tbt: Targeted neural network attack with bit trojan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13198–13207, 2020. 2

[30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv:1506.01497*, 2015. 4

[31] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. *arXiv preprint arXiv:1910.00033*, 2019. 3

[32] Yiru Shen, Chen Feng, Yaoqing Yang, and Dong Tian. Mining point cloud local structures by kernel correlation and graph pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4548–4557, 2018. 2

[33] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019. 2

[34] Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3693–3702, 2017. 2

[35] Satya P Singh, Lipo Wang, Sukrit Gupta, Haveesh Goli, Parasuraman Padmanabhan, and Balázs Gulyás. 3d deep learning on medical images: a review. *Sensors*, 20(18):5097, 2020. 1

[36] Ruixiang Tang, Mengnan Du, Ninghao Liu, Fan Yang, and Xia Hu. An embarrassingly simple approach for trojan attack in deep neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 218–228, 2020. 2

[37] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6411–6420, 2019. 2

[38] 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

[39] James Tu, Mengye Ren, Sivabalan Manivasagam, Ming Liang, Bin Yang, Richard Du, Frank Cheng, and Raquel Urtasun. Physically realizable adversarial examples for lidar object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13716–13725, 2020. 3, 4

[40] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Clean-label backdoor attacks. 2018. 2, 8

[41] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*, 2019. 2

[42] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2569–2578, 2018. 2

[43] 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. 2, 6

[44] Yuxin Wen, Jiehong Lin, Ke Chen, and Kui Jia. Geometry-aware generation of adversarial and cooperative point clouds. *arXiv preprint arXiv:1912.11171*, 2019. 3

[45] 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

[46] 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. 3, 6

[47] 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

[48] Qiangeng Xu, Xudong Sun, Cho-Ying Wu, Panqu Wang, and Ulrich Neumann. Grid-gcn for fast and scalable point cloud learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5661–5670, 2020. 2

[49] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning object bounding boxes for 3d instance segmentation on point clouds. In *Advances in Neural Information Processing Systems*, pages 6740–6749, 2019. 2

[50] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. Modeling point clouds with self-attention and gumbel subset sampling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3323–3332, 2019. 2

[51] Shihao Zhao, Xingjun Ma, Xiang Zheng, James Bailey, Jingjing Chen, and Yu-Gang Jiang. Clean-label backdoor attacks on video recognition models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14443–14452, 2020. 2

[52] 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. 1, 3, 5, 8

[53] 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. 3

[54] 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. 3