Deflecting 3D Adversarial Point Clouds Through Outlier-Guided Removal

Hang Zhou  Kejiang Chen  Weiming Zhang*  Han Fang  Wenbo Zhou  Nenghai Yu
University of Science and Technology of China
{zh2991, chenkj, fanghan, welbeckz}@mail.ustc.edu.cn  {zhangwm, ynh}@ustc.edu.cn

Abstract

Neural networks are vulnerable to adversarial examples, which poses a threat to their application in security sensitive systems. We propose simple random sampling (SRS) and statistical outlier removal (SOR) as defenses for 3D point cloud classification, where both methods remove points by estimating probability of points serving as adversarial points. Compared with ensemble adversarial training which is the state-of-the-art defending method, SOR has several advantages: better defense performance, randomization makes the network more robust to adversarial point clouds, no additional training or fine-tuning required, and few computations are needed by adding the points-removal layer. In particular, our experiments on ModelNet40 show that SOR is very effective as defense in practice. The strength of those defenses lies in their non-differentiable nature and inherent randomness, which makes it difficult for an adversary to circumvent the defenses. Our best defense eliminates 81.4% of strong white-box attacks by C&W and $l_2$ loss based attack methods.

1. Introduction

Deep Learning has shown superior performance on several categories of machine learning problems, especially classification task. These Deep Neural Networks (DNN) learn models from large training data to efficiently classify unseen samples with high accuracy. However, recent works have demonstrated that DNNs are vulnerable to adversarial examples, which are maliciously created by adding imperceptible perturbations to the original input by attackers. Adversarially perturbed examples have been deployed to attack image classification service [13], speech recognition system [4] and autonomous driving system [27].

Heretofore, numerous algorithms have been proposed to generate adversarial examples for 2D images. When model parameters are known, a paradigm called white-box attacks includes methods based on calculating the gradient of the network, such as Fast Gradient Sign Method (FGSM) [7], Iterative Gradient Sign Method (IGSM) [8] and Jacobian Saliency Map Attack Method (JSMA) [18], and based on solving optimization problems, such as L-BFGS [23], Deepfool [16] and Carlini & Wagner (C&W) attack [2]. In the scenario where access to the model is not available, called black-box attacks, a secondary model can be trained using the model to be attacked as a guide [17, 13].

Since the robustness of DNNs to adversarial examples is a critical feature, defenses that target to increase robustness against adversarial example are urgently considered and can be classified into three main categories. Given the local instability of adversarial examples, the first category contrives to remove adversarial perturbations from the input by input transformations such as JPEG compression [6] or image rescaling [14]. Meng & Chen [15] introduce MagNet by training an auto-encoder reformer network to move adversarial examples closer to the manifold of natural examples. The second category is adversarial training, where Goodfellow et al. [23] augment training data with adversarial examples to increase the robustness of the model against a specific attack. The third category is gradient masking, including modifying network architecture and optimization techniques to suppress the generation of adversarial examples. Obfuscated gradients, as a special case of gradient masking [17], make it harder for attackers to compute the feasible gradient to generate adversarial examples. Zheng et al. [32] append a stability term to the objective function to enforce the model to share similar outputs for both the normal and adversarial examples.

In addition to defense, detection of adversarial example before they are fed into the networks is another approach to resist attacks. The detector networks from MagNet [15] learn to distinguish between normal and adversarial examples by approximating the manifold of normal examples. Liu et al. [12] detect adversarial examples by estimating modification probability of each pixel caused by adversarial attacks from the steganalysis point of view.

As for 3D geometric data such as point clouds or meshes, a couple of works on point cloud classification have been researched. After the awkward problem of irregular data for-
mat was addressed by PointNet [3] and its variants [21, 25], point cloud data can be directly processed by DNNs, and has become a promising data structure for 3D computer vision tasks. Hua et al. [9] propose a pointwise convolution operator that can output features at each point in a point cloud, which can offer competitive accuracy while being simple to implement. Yang et al. [29] construct losses based on mesh shape and texture to generate adversarial examples, which aim to project the optimized “adversarial meshes” to 2D with a photorealistic renderer, and still able to mislead different DNNs. Xiang et al. [27] attack point clouds built upon C&W loss and point cloud-specific perturbation metric with high success rate. To our best of our knowledge, this is the only work on 3D adversarial point clouds. Since techniques like Lidar have been widely deployed in safety-critical scenarios such as autonomous driving [31], the robustness of 3D point cloud recognition against adversarial examples is of great significance.

Based on the above reasoning, in this paper, we propose a defense method against adversarial point cloud by randomization at inference time using simple random sampling or statistical outlier removal controlled by random seed, to mitigate adversarial effects. As far as we know, this is the first work that demonstrates the effectiveness of point-removal operation at inference time on mitigating adversarial effects on the 3D dataset, e.g., ModelNet40. We summarize the key contributions of our work as follows:

- We present two new defense operations to mitigate adversarial point clouds, which have better defense performance compared with adversarial training.

- Randomization at inference time makes the network more robust to adversarial point clouds but hardly deteriorates the performance on clean point clouds.

- There is no additional training or fine-tuning required, and very few computations are required by adding the points-removal layer. Thus there is nearly no runtime increase.

We conduct comprehensive experiments to test the effectiveness of our defense method against Xiang et al.’s attacks [27] with multiple loss metrics, and under different attack scenarios. The results in Section 4 demonstrate that the proposed points-removal layer can significantly mitigate adversarial effects.

2. Related Work

2.1. Point Clouds

A point cloud is a set of points which are sampled from object surfaces. Consider a 3D point cloud with \( n \) points, denoted by \( \mathbf{X} = \{ \mathbf{x}_i \}_{i=1,2,...,n} \), where each point \( \mathbf{x}_i \) is a vector of its \( xyz \) coordinates. Note that unlike images, point cloud data are unordered and dimensionality-flexible, which are dealt with differently.

**PointNet.** This method [20] and its variants [21] proposed by Qi et al. exploit a single symmetric function, max pooling, to reduce the unordered and dimensionality-flexible input data to a fixed-length global feature vector and enable end-to-end neural network learning. They demonstrate the robustness of the proposed PointNet and introduce the concept of critical points and upper bounds. The points sets laying between critical points and upper bounds yield the same global features, and thus PointNet is robust to missing points and random perturbation.

2.2. Existing Methods for Adversarial Attacks

**Carlini & Wagner.** This method [2] is an optimization-based attack that combines a differentiable surrogate for the classification accuracy of the model with three forms of distortion term \((l_0, l_2, l_\infty)\). It generates adversarial examples by solving the following optimization problem:

\[
\begin{align*}
\min_{\delta} & \quad D(\mathbf{X}, \mathbf{X} + \delta) + c \cdot f(\mathbf{X} + \delta) \\
\text{s.t.} & \quad \mathbf{X} + \delta \in [0, 1]^n
\end{align*}
\]  

where \( c \) is a hyperparameter to balance the two parts. This attack seeks for the solution of both acquiring the smallest perturbation measured by pre-defined perturbation loss \( D \) and impelling the network to classify the adversarial example incorrectly. For an untargeted attack, \( f(\mathbf{X}) \) is the loss function to measure the distance between the input object and the adversarial object, as defined by:

\[
f(\mathbf{X}) = \max(Z(\mathbf{X})_{\text{true}} - \max_{i \neq \text{true}} Z(\mathbf{X})_i), -\kappa
\]

where \( \kappa \) denotes a margin parameter for regulating model transferability and perturbation degree, and where \( Z(\mathbf{X}) \) is the operation that computes the logit vector. So far C&W attack is too strong to defend.

**Xiang’s method.** Xiang et al. [27] propose the first adversarial examples in the point cloud space, including unnoticceable and manufacturable adversarial point clouds. Based on the framework of C&W attack, for unnoticeable adversarial examples, they either shift existing points or add new points negligibly and adopt different perturbation metrics \( D(\mathbf{X}, \mathbf{X'}) \) based on them, where \( \mathbf{X'} \) stands for adversarial point cloud. To measure the attack performance, they propose to use \( l_2 \) norm, Pompeiu-Hausdorff distance

\[
D(\mathbf{X}, \mathbf{X'}) = \max \min_{\mathbf{x'} \in \mathbf{X'}} \| \mathbf{x} - \mathbf{x}' \|_2^2,
\]

Chamfer measurement

\[
D(\mathbf{X}, \mathbf{X'}) = \frac{1}{\| \mathbf{X} \|_0} \sum_{\mathbf{x'} \in \mathbf{X'}} \min_{\mathbf{x} \in \mathbf{X}} \| \mathbf{x} - \mathbf{x}' \|_2^2,
\]
and the number of added points
\[
\text{Count}(X, X') = \sum_{x' \in X'} \mathbb{I}[\min_{x \in X} \|x - x'\|_2 > T_{thre}]
\]  

as perturbation metrics, where \(\mathbb{I}[]\) is the indicator function and \(T_{thre}\) the threshold of outliers. Experiments demonstrate that all the adversarial point clouds reach high success rate given an acceptable perturbation budget.

2.3. Existing Methods for Defenses

As far as we know, there are no work on 3D adversarial point cloud defenses. However, adversarial training is a universal approach to defend against adversarial examples.

Adversarial Training. Adversarial training [7, 10, 24] is one of the most extensively investigated defenses against adversarial attacks. It aims to train a robust model from scratch on a training set augmented with adversarially perturbed data. Adversarial training improves the classification accuracy of the target model on adversarial examples. However, adversarial training is more time consuming than training on clean objects only, because online adversarial example generation needs extra computation, and it takes more epochs to fit adversarial examples. These limitations hinder the usage of harder attacks in adversarial training.

3. Defenses against Adversarial Point Cloud

The goal of defense on 3D point clouds is to build a network that is robust to adversarial examples, i.e., it can classify adversarial point clouds correctly with little performance loss on clean point clouds. Formally, given a classification model \(f\) and an input \(X\), which may either be an original input \(X\) or an adversarial input \(X'\), the goal of a defense method is to either augment data to train a robust \(f'\) such that \(f'(X) = f(X)\), or transform \(X\) by a transformation \(T\) such that \(f(T(X)) = f(X)\).

Towards this goal, we propose a random points-removal method, as shown in Figure 1, which adds a points-removal layer to the beginning of the classification networks, to realize network robustness against adversarial examples. These points-removal layers are designed in the context of point cloud classification on ModelNet40 [26] dataset and are used in conjunction with a trained classifier \(f\) (By default pre-trained PointNet [20] in this study). There is no retraining or fine-tuning needed which makes the proposed method very easy to implement.

We propose two defense schemes: simple random sampling and statistical outlier removal to defend against 3D point cloud adversarial examples generated from C&W loss and multiple distance metrics.

3.1. Simple Random Sampling (SRS)

In statistics, a simple random sample, or shortly SRS, is a subset of individuals chosen from a larger set. Each sample is chosen randomly and entirely by chance, such that each has the same probability of being chosen at any stage during the sampling process, which is an unbiased surveying technique.

We randomly leave out \(r\) points from \(n\) points to preprocess the input point clouds. The input order of points is
Algorithm 2 Statistical outlier removal transform

\textbf{Input}: Point cloud $X$, nearest neighbor number $k$ and outlier truncation parameter $\alpha$

\textbf{Output}: Point cloud $X'$

1: Initialize $X' = \emptyset$
2: Compute the average distance $d_i$ that each point $x_i$ has to its nearest $k$ neighbors by Equation (6)
3: Compute the mean $\bar{d}$ and standard deviation $\sigma$ of all these distances by Equation (7) and (8)
4: for $i \leftarrow 0$ to $n$ do
5: \hspace{1em} if the average distance $d_i < \bar{d} + \alpha \cdot \sigma$ then
6: \hspace{2em} $X' = X' \cup x_i$
7: return $X'$

The mean and standard deviation of all these distances are computed to determine a distance threshold:

$$\bar{d} = \frac{1}{n} \sum_{i=1}^{n} d_i, \quad i = 1, \ldots, n, \quad (7)$$

and

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (d_i - \bar{d})^2}. \quad (8)$$

We trim the points which fall outside the $\mu \pm \alpha \cdot \sigma$, and the manicured point set $X'$ is acquired by

$$X' = \{x_i | d_i < \bar{d} + \alpha \cdot \sigma\}. \quad (9)$$

In summary, we have given a formal definition in Algorithm 2. The outliers generated from 3D scanners share certain similarities with points generated by C&W based 3D adversarial point clouds. The addition of Hausdorff or Chamfer distance loss still cannot hinder the detection of adversarial example, which is primarily on account of the fact that despite the attackers successfully fool the classification network, there is always a certain percentage of points which are added or shifted inevitably becoming abnormal points. Also, the outlier evaluation variances between single distortion measurements of loss function and statistic outlier are unequal.

Below we explore the relationship between the two measures. The limitation of C&W optimization function inevitably creates some points that are on the manifold of point cloud object which are taken as normal points and some are outliers. The outliers mostly mislead the classification performance. Therefore, the more outliers removed by preprocessing layer, the better the defense ability against adversarial examples. Here, we denote the percentage of adversarial points in the removed point set by

$$p = \frac{|X_{adv} \cap (X - X')|}{|X - X'|}, \quad (10)$$

where $X_{adv}$ is the set of adversarial points which is defined differently w.r.t. diverse adversarial distortion constraints. For a $l_2$ loss, $X_{adv}$ is defined by

$$X_{adv} = \{x'_i | \|x_i - x'_i\|_2 > T(X, X', \beta)\}, \quad (11)$$

where $x'_i \in X'$ and $T(X, X', \beta)$ is the threshold of $l_2$ norm of each paired points controlled by $\beta$ the ratio of points that are considered as adversarial points. For Hausdorff or Chamfer based loss, $X_{adv}$ is defined by

$$X_{adv} = \{x'_i | \min_{m \in X} \|m - x'_i\|_2 > T(X, X', \beta)\}, \quad (12)$$

where $x'_i \in X'$ and $T(X, X', \beta)$ is the threshold of Hausdorff/Chamfer distance between each point from $X'$ and point set $X$ controlled by $\beta$. 

trivial and makes no difference to the classification performance. The effectiveness of points deletion manipulation is attributed to the structure of PointNet that on the final convolutional layer, a global max pool is applied to aggregate available features to represent shape characteristics. Random removal of some points from the original point cloud does not alter the classification of a clean point cloud, while the accuracy of adversarial point clouds has soared from 0% to 57%. A formal definition is given in Algorithm 1.

3.2. Statistical Outlier Removal (SOR)

Because point clouds are generally produced by 3D scanners which measure a large number of points on the external surfaces of objects around them, measurement errors by scanners inevitably lead to sparse outliers which corrupt the shapes of point clouds. The phenomenon complicates the estimation of local point cloud region characteristics such as surface normals or curvature changes, leading to erroneous values. Rusu et al. [22] propose statistical outlier removal method (SOR for short) which corrects these irregularities by computing the mean $\mu$ and standard deviation $\sigma$ of nearest neighbor distances, and trim the points which fall outside the $\mu \pm \alpha \cdot \sigma$, where $\alpha$ depends on the size of the analyzed neighborhood.

Specifically, the $k$-nearest neighbors ($k$NN) point set of each point $x_i$ of point cloud $X$ is defined as $knn(X, x_i, k)$. Then the average distance $d_i$ that each point $x_i$ has to its nearest $k$ neighbors is denoted by

$$d_i = \frac{1}{k} \sum_{x_j \in knn(X, x_i, k)} \|x_i - x_j\|_2, \quad i = 1, \ldots, n. \quad (6)$$

The mean and standard deviation of all these distances are computed to determine a distance threshold:
By Equation (10), we acquire the percentage of adversarial points \( p \) of SOR and SRS method, and denote them by \( p_{\text{SOR}} \) and \( p_{\text{SRS}} \) respectively. It is expected that \( p_{\text{SOR}} > p_{\text{SRS}} \), since SOR scheme recognizes outliers as adversarial points in a statistical pattern rather than random guess as SRS does. We choose 300 point clouds as test examples to verify the above inference, which is shown in Figure 2. Most of \( p_{\text{SOR}} \) of point clouds are larger than \( p_{\text{SRS}} \), implying that SOR removes more adversarial points than SRS. Similar results can be obtained on Hausdorff and Chamfer loss based adversarial point clouds. Thus SOR has a better ability of defense against adversarial point clouds.

4. Experiments

4.1. Experimental Setup

**Dataset.** We use the aligned benchmark ModelNet40 [26] dataset for our experiments. The ModelNet40 dataset contains 12,311 CAD models from 40 most common object categories in the world. 9,843 objects are used for training and the other 2,468 for testing. As done by Qi et al., we uniformly sample 1,024 points from the surface of each object and rescale them into a unit cube.

**Networks.** We use the same PointNet structure as proposed in [20] and train the model with all ModelNet40 training data to obtain the PointNet model. The adv-PointNet is adversarially trained whose training set is combined with original clean point clouds and adversarial examples from C&W and \( l_2/\text{Hausdorff/Chamfer} \) loss metrics.

**Defense Models.** The defense model consists of the original networks (PointNet) and the preprocessing layer. By applying the points-removal layer, we can create innumerable different patterns for a single point cloud, which is hard for estimation.

**Untargeted Models.** We utilize all the test examples to generate untargeted adversarial point clouds with one attack built upon C&W loss and \( l_2 \) metric [27].

**Targeted Models.** We utilize all the test examples to generate untargeted adversarial point clouds with three attacks built upon C&W loss and \( l_2 \) metric, Hausdorff metric and Chamfer metric, respectively [27]. The target class of each adversarial example is picked randomly from the remaining 39 categories. We do not consider defense of manufacturable adversarial point cloud proposed in [27], as visually it is not a normal point cloud and can be identified before it is fed into point cloud recognition network.

**Attacks Evaluations.** The attackers first generate adversarial examples using the untargeted/targeted models and then evaluate the classification accuracy of these generated adversarial examples on the target and defense models. Low accuracy of the untargeted/targeted model indicates that the attack is successful, and high accuracy of the defense model indicates that the defense is effective.

- **Vanilla Attack:** The attackers do not know the existence of the random points-removal layer and the target model is just the original network.

- **Single-Pattern Attack:** The attackers know the existence of the random points-removal layer. In order to mimic the structures of defense models, the target model is chosen as the original network + points-removal layer with only one predefined pattern.

- **Ensemble-Pattern Attack:** The attackers know the existence of the random points-removal layer. To mimic the structures of defense models in a more representative way, the target model is chosen as the original network + points-removal layer with an ensemble of predefined patterns.

4.2. Parameter Selection

We take SRS and SOR as our two defense schemes and adversarial examples generated from C&W and \( l_2/\text{Hausdorff/Chamfer} \) loss for performance verification.

**SRS as Defense.** As shown in Figure 3, we compare the detection accuracy and attack success rate of targeted attacks with a varying number of removed points \( r \) from 0 to 1000. Note that the two evaluations are not directly associated because a low detection accuracy does not mean a high success rate for a targeted attack. As \( r \) increases, the success rate of adversarial examples drops dramatically, the average accuracy of adversarial example first increases and then decreases with its maximum 65.1%, and the accuracy of clean point clouds are monotonically decreasing.
The tendency of three curves can be explained below: the attacks search the entire point cloud space for adversarial perturbations without regarding for the location of the point cloud content. This is contrary to the classification models that show high activation in regions where object shapes are present [30]. Therefore, simple removal-based filtering with a slight amount of deletion erases the artifact bothered by adversarial perturbation, which promotes detection of adversarial point clouds. When a few points are deleted, the structure of point cloud is still preserved; when more random sampled points are deleted, the shape of the point cloud deteriorates and degrades the classification performance.

**SOR as Defense.** The SOR operation comprises two influential factors, \( k \) the number of neighbor points and \( \alpha \) the percentage of points that are regarded as outliers. As shown in Figure 5, with fixed \( k = 2 \), distribution of number of points with various \( \alpha \) are presented. Compared to the clean points clouds with 1024 points, a smaller \( \alpha \) filters out more points than in a statistic sense. Similarly, for the target model, we evaluate the targeted attack based adversarial examples, as shown in Figure 6. When \( k = 0 \), the \( k\text{NN} \) point set only contains the point itself, thus the statistical removal is inoperative. Once \( k \geq 1 \), the defense behavior comes into force. When \( k = 2 \) and \( \alpha = 1.1 \), the accuracy of clean point clouds and adversarial examples are 86.5% and 81.4% respectively. Compared to SRS defense with its best accuracy of adversarial examples with 65.1%, SOR has a substantial increase of 16.3% on performance. Similar results can be obtained on defenses of untargeted attacks and Hausdorff loss based attacks, which are shown in Figure 7 and 8. In Figure 4 we show the visual variation of point positions of each stage. The figure on the far right consists of the removed outliers \((X_{adv} \cap (X - X'))\) in red and the removed points that are not outliers \((X_{adv} - X_{adv} \cap (X - X'))\) in black. The larger ratio the red points has, the better the defense performance has.
to the data augmentation. As for $l_2$ metric, the success rate of adversarial examples nearly reaches 100% while after filtering the detection rate of SRS is 64.7% and that of SOR is higher than 80%. In contrast, Hausdorff and Chamfer loss metrics have similar results with lower success rates but have lower detection rates after points-removal preprocessing. The success rate of adversarial examples on target adv-PointNet model reaches 100%, but is it more fragile than defense against PointNet model with more than 80% accuracy on both SRS and SOR models. Furthermore, due to the unknowability of the defense type, there is only 1-round attack for attackers, and we only have a 1-round defense.

### 4.4. Single-Pattern Attack Scenario

For the single-pattern attack, the attackers are aware of the existence of points-removal layer and also the parameters of the outlier removal operation (i.e., from $k = 2 \sim 5$ and $\alpha = 0.6 \sim 1.1$) or random sampling (i.e., from $r = 200 \sim 600$), but they do not know the specific patterns utilized by the defense models (even the defense models themselves do not know these specific patterns since they are randomly instantiated at test time). Distinct from the adversarial examples of 2D images [28] that considers the preprocessing layer as a part of the convolutional network to compute gradients, it is unlikely for attackers to acquire gradients of a layer with unknown neuron numbers (number of points after removal manipulation). To generate robust adversarial examples against preprocessing operation, the attackers try to attack the model more than once. First, generate the 1-round adversarial point cloud and use one specific pattern of points-removal to acquire point clouds as the clean point cloud, which are used to acquire the 2-round adversarial point cloud. In this experiment, the specific pattern of SOR preprocessing that we use is to transform the original input $X$ to points-removed point cloud $T(X)$ with $k = 2$ and $\alpha = 1.1$, while for SRS scheme $r = 500$.

Table 2 shows the accuracy of both target and defense models, and the adversarial point clouds are all 2-round attacks. $l_2$ loss based 2-round attacks 100% successfully fool the original target PointNet but has an obvious accuracy increase on both the SRS and SOR defense model. Consistent with the analysis in Subsection 3.2, SOR removes more adversarial points than SRS. For Hausdorff/Chamfer based

---

1Different from the adversarial training of 2D images, to align input dimension of 3D point cloud training data while keeping the classification accuracy of training data unchanged, we pad the vacant points by replication of existing points.
adversarial examples, the 2-round attacks perform worse than $l_2$ based attacks with near 50% classification accuracy, which is due to the difficulty of adversarial points generation caused by the compact distribution of added points.

### 4.5. Ensemble-Pattern Attack Scenario

For the ensemble-pattern attack, similar to single-pattern attack, the attackers are aware of the points-removal layer and the parameters of the outlier removal and random sampling (i.e., from $k = 2 \sim 5$ and $\alpha = 0.6 \sim 1.1$), but they do not know the specific patterns utilized by the defense models at test time. The target models thus are constructed in a more representative way: let points-removal layer choose an ensemble of predefined patterns, and the goal of the attackers is to let all chosen patterns fail on classification. In this experiment, the specific ensemble patterns that we choose are: we select $k$ and $\alpha$ from both intervals randomly for each point cloud, resulting in an infinite patterns. For the results presented in Table 2, we can see that the for $l_2$ loss based 2-round adversarial examples generated under ensemble-pattern attack scenario are inferior to single-pattern based attacks. For Hausdorff/Chamfer based adversarial examples, the 2-round attacks perform similarly to single-pattern based attacks.

### 4.6. Black-Box Attack Scenario

The transferability of C&W loss based 3D adversarial point cloud on black-box classification systems is experimented, which is shown in Table 3. Similar to [27], we test the success rate of adversarial examples generated from PointNet on PointNet++, DGCNN and PointwiseCNN. The result illustrates that C&W based 3D adversarial point clouds have limited transferability, thus for black-box defense, we do not have to add the preprocessing layer. For PointwiseCNN, the input size of points should be fixed (such as 1024 points) and is limited for classification. Thus some classification results are absent.

## 5. Conclusion

In this paper, we propose a points-removal based network layer as a defense mechanism to mitigate 3D point cloud adversarial effects and strength the robustness of DNNs. We conduct comprehensive experiments to validate the effectiveness of our defense method against different C&W based attacks under different attack scenarios. The results show that it is effective to defend against white-box attacks; while for black-box attacks, it is not necessary to add the proposed preprocessing layer. By adding the proposed points-removal layer to a trained classification model, it achieves the best score of 0.814 of accuracy on the C&W and $l_2$ loss based adversarial point clouds.
References

[1] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. arXiv preprint arXiv:1802.00420, 2018.

[2] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. arXiv preprint arXiv:1608.04644, 2016.

[3] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, pages 77–85. IEEE, 2017.

[4] M. Cisse, Y. Adi, N. Neverova, and J. Keshet. Houdini: Fooling deep structured prediction models. arXiv preprint arXiv:1707.05373, 2017.

[5] N. Das, M. Shanbhogue, S.-T. Chen, F. Hohman, L. Chen, M. E. Kounavis, and D. H. Chau. Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. arXiv preprint arXiv:1705.02900, 2017.

[6] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy. A study of the effect of jpeg compression on adversarial images. arXiv preprint arXiv:1608.00853, 2016.

[7] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples (2014). arXiv preprint arXiv:1412.6572.

[8] S. Gu and L. Rigazio. Towards deep neural network architectures robust to adversarial examples. arXiv preprint arXiv:1412.5086, 2014.

[9] B.-S. Hua, M.-K. Tran, and S.-K. Yeung. Pointwise convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 984–993, 2018.

[10] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial machine learning at scale. arXiv preprint arXiv:1611.01236, 2016.

[11] F. Liao, M. Liang, Y. Dong, T. Pang, J. Zhu, and X. Hu. Defense against adversarial attacks using high-level representation guided denoiser. arXiv preprint arXiv:1712.02976, 2017.

[12] J. Liu, W. Zhang, Y. Zhang, D. Hou, Y. Liu, and N. Yu. Detecting adversarial examples based on steganalysis. arXiv preprint arXiv:1806.09186, 2018.

[13] Y. Liu, X. Chen, C. Liu, and D. Song. Delving into transferable adversarial examples and black-box attacks. arXiv preprint arXiv:1611.02770, 2016.

[14] J. Lu, H. Sibai, E. Fabry, and D. Forsyth. No need to worry about adversarial examples in object detection in autonomous vehicles. arXiv preprint arXiv:1707.03501, 2017.

[15] D. Meng and H. Chen. Magnet: a two-pronged defense against adversarial examples. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pages 135–147. ACM, 2017.

[16] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2574–2582, 2016.

[17] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, pages 506–519. ACM, 2017.

[18] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. The limitations of deep learning in adversarial settings. In Security and Privacy (EuroS&P), 2016 IEEE European Symposium on, pages 372–387. IEEE, 2016.

[19] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. arXiv preprint arXiv:1511.04508, 2015.

[20] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. Proc. Computer Vision and Pattern Recognition (CVPR), IEEE, 1(2):4, 2017.

[21] C. R. Qi, L. Yi, H. Su, and L. 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.

[22] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz. Towards 3d point cloud based object maps for household environments. Robotics and Autonomous Systems, 56(11):927–941, 2008.

[23] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199, 2013.

[24] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. Ensemble adversarial training: Attacks and defenses. arXiv preprint arXiv:1705.07204, 2017.

[25] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. Bronstein, and J. M. Solomon. Dynamic graph cnn for learning on point clouds. arXiv preprint arXiv:1801.07829, 2018.

[26] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. 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.

[27] C. Xiang, C. R. Qi, and B. Li. Generating 3d adversarial point clouds. arXiv preprint arXiv:1809.07016, 2018.

[28] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille. Mitigating adversarial effects through randomization. arXiv preprint arXiv:1711.01991, 2017.

[29] D. Yang, C. Xiao, B. Li, J. Deng, and M. Liu. Realistic adversarial examples in 3d meshes. arXiv preprint arXiv:1810.05206, 2018.

[30] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. arXiv preprint arXiv:1506.06579, 2015.

[31] D. Zermas, I. Izzat, and N. Papanikolopoulos. Fast segmentation of 3d point clouds: A paradigm on lidar data for autonomous vehicle applications. In Robotics and Automation (ICRA), 2017 IEEE International Conference on, pages 5067–5073. IEEE, 2017.

[32] S. Zheng, Y. Song, T. Leung, and I. Goodfellow. Improving the robustness of deep neural networks via stability training.
In Proceedings of the ieee conference on computer vision and pattern recognition, pages 4480–4488, 2016.