DUP-Net: Denoiser and Upsampler Network for 3D Adversarial Point Clouds Defense

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 a Denoiser and UPsampler Network (DUP-Net) structure as defenses for 3D adversarial point cloud classification, where the two modules reconstruct surface smoothness by dropping or adding points. In this paper, statistical outlier removal (SOR) and a data-driven upsampling network are considered as denoiser and upsampler respectively. Compared with baseline defenses, DUP-Net has three advantages. First, with DUP-Net as a defense, the target model is more robust to white-box adversarial attacks. Second, the statistical outlier removal provides added robustness since it is a non-differentiable denoising operation. Third, the upsampler network can be trained on a small dataset and defends well against adversarial attacks generated from other point cloud datasets. We conduct various experiments to validate that DUP-Net is very effective as defense in practice. Our best defense eliminates 83.8% of C&W and l_2 loss based attack (point shifting), 50.0% of C&W and Hausdorff distance loss based attack (point adding) and 9.0% of saliency map based attack (point dropping) under 200 dropped points on PointNet.

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 [18], speech recognition system [5] and autonomous driving system [34].

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 FGSM [9], IGSM [10] and JSMA [23], and based on solving optimization problems, such as L-BFGS [29], Deepfool [21] and Carlini & Wagner (C&W) attack [3]. In the scenario where access to the model is not available, called *black-box attacks*.

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: input transformations [7, 19, 20], adversarial training [29] and gradient masking [22, 41]. In addition to defense, *detection* of adversarial examples before they are fed into the networks is another approach to resist attacks, such as MagNet [20] and steganalysis based detection [17].

The popularity of 3D sensors such as LiDAR and RGBD cameras draws many research concerns with 3D vision. An increasing number of accessible data motivates data-driven deep learning methods practical to be used in many areas, including autopilot [24, 43], robotics [12, 6] and graphics [35, 13, 31]. In particular, point cloud is one of the most natural data structures to represent the 3D geometry. After the difficult problem of irregular data format was addressed by DeepSets [40], PointNet [4] and its variants [26, 32], point cloud data can be directly processed by DNNs, and has become a promising data structure for 3D computer vision tasks. Hua *et al.* [11] propose a pointwise convolution operator that can output features at each point, which can offer competitive accuracy while being easy to implement. Yang *et al.* [36] 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.* [34] attack point clouds built upon C&W loss and point cloud-specific perturbation metric with high success rate. Zheng *et al.* [42] propose a malicious point-dropping method to generate adversarial

---
*Corresponding author.

point clouds based on learning a saliency map for a whole point cloud, which assigns each point a score reflecting its contribution to the model-recognition loss. Liu *et al*. [16] propose several iterative gradient based attack methods and input restoration based defenses. Yang *et al*. [37] propose point-detach strategy utilizing the critical point property to attack neural network based classification system, and introduce several perturbation defenses.

Adversarial examples do well in 3D point cloud classification, and probably cause inconvenient issues even security problems. Consequently, research on defense of 3D point cloud adversarial example is in urgent need. Based on the above reasoning, in this paper, we propose a defense method against adversarial point cloud by training a Denoiser and UPsampler Network (DUP-Net) to mitigate adversarial effects. As far as we know, this is the first work that demonstrates the effectiveness of point-dropping and point-adding operations 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 modules to mitigate adversarial point clouds, which have better performance compared with baseline methods.

- The nondifferentiability property of denoise layer, statistical outlier removal, is utilized to defend the adversarial white-box attacks.

- The upsampler network can be trained on a small dataset and defends well against adversarial attacks generated from other point cloud datasets.

We conduct comprehensive experiments to test the effectiveness of our defense method against point shifting/dropping/adding attacks from [34, 39]. The results in Section 4 demonstrate that the proposed DUP-Net can significantly mitigate adversarial effects.

2. Related Work

2.1. Point Clouds and PointNet

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. PointNet [25] and its variants [26] 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 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

Point Shifting. Xiang *et al*. [34] propose C&W framework [3] based unnoticeable adversarial point clouds by point perturbation. C&W is an optimization-based attack that combines a differentiable surrogate for the classification accuracy of the model. It generates adversarial examples by solving the following optimization problem:

$$\min_{\delta} D(\mathbf{X}, \mathbf{X} + \delta) + c \cdot f(\mathbf{X} + \delta)$$
$$s.t. \quad \mathbf{X} + \delta \in [0, 1]^n \tag{1}$$

This attack seeks for the solution of both acquiring the smallest perturbation 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 and the adversarial object, as defined by:

$$f(\mathbf{X}) = \max(Z(\mathbf{X})_{true} - \max_{i \neq true}\{Z(\mathbf{X})_i\}, -\kappa) \tag{2}$$

where κ denotes a margin for regulating model transferability and perturbation degree, and $Z(\mathbf{X})$ is the logit vector.

Xiang *et al*. shift existing points negligibly and adopt different perturbation metrics $D(\mathbf{X}, \mathbf{X}')$ (l_p norm, Hausdorff distance and Chamfer measurement), where $\mathbf{X}'$ stands for adversarial point cloud. Liu *et al*. [16] extend fast/iterative gradient method by constraining the perturbation magnitude onto the surface of an epsilon ball in different dimensions.

Point Adding. Xiang *et al*. [34] also propose points adding based attacks with C&W and Hausdorff/Chamfer measurements. Because directly adding points to the unconstrained 3D space is infeasible due to the large search space, they propose an initialize-and-shift method to find appropriate position for each added point. Besides, for adversarial clusters, they minimize the radius of the generated cluster to make it attached to the surface of original object. Yang *et al*. [37] develop a variant of one-pixel attack [28] using pointwise gradient method to only update the attached points without changing the original point cloud.

Point Dropping. For any point cloud $\mathbf{X}$, Qi *et al*. [25] prove that there exists a subset $\mathbf{C} \subseteq \mathbf{X}$, namely critical subset, which determines all the max pooled features $\max_{\mathbf{x}_i \in \mathbf{X}}\{\mathbf{h}(\mathbf{x}_i)\}$, and thus the output of PointNet, which is only applicable to network structures similar to $\gamma \circ \max_{\mathbf{x}_i \in \mathbf{X}}\{\mathbf{h}(\mathbf{x}_i)\}$. Visually, $\mathbf{C}$ usually distributes evenly along the skeleton of $\mathbf{X}$. In this sense, dropping points in $\mathbf{C}$ can also generate adversarial point clouds. Zheng *et al*. [42] propose point dropping based attack by first constructing the saliency map:

$$s_i = -r_i^{\alpha} r_i \frac{\partial \mathcal{L}}{\partial r_i}, \tag{3}$$

where $r_i^{\alpha} \frac{\partial \mathcal{L}}{\partial r_i} = (\mathbf{x}_i - \mathbf{x}_c) \cdot \mathbf{g}_i$ (inner product), $\mathbf{x}_c$ is the cloud center and $\mathbf{g}_i$ is the gradient under orthogonal coordinates
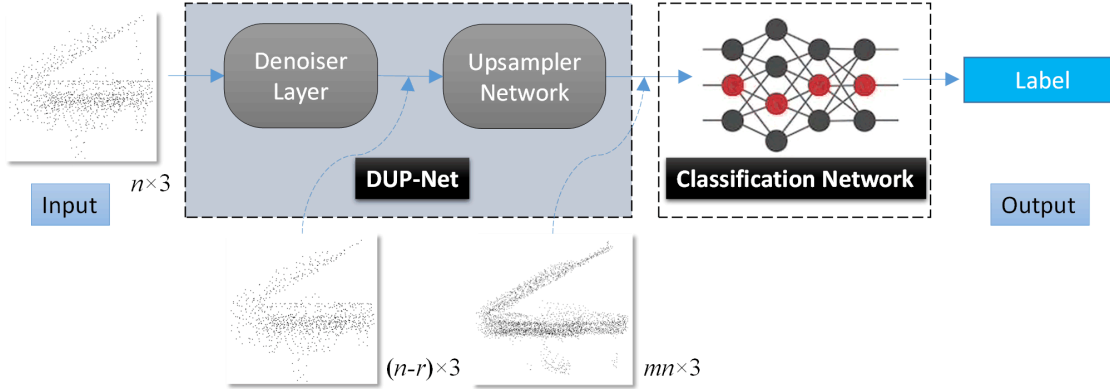
Figure 1: The pipeline of our DUP-Net defense method. The input point cloud is first denoised by a statistical outlier removal layer and then upsampled by a pretrained upsampling neural network. The preprocessed point cloud is then fed into the classification neural network.

$\mathbf{g}_i = \nabla_{\mathbf{x}_i} \mathcal{L}(\mathbf{X}, y; \boldsymbol{\theta})$, and α is a rescaling hyperparameter. Points with n/T lowest s_i are dropped, and the operation are iterative executed for T loops. Yang *et al.* [37] develop a point-detach strategy similar to [42], which utilizes the critical point property to iteratively detach the most important point to confuse the attacked network.

2.3. Existing Methods for defenses

Adversarial Training. Adversarial training [9, 15, 30] is one of the most investigated defenses against adversarial attacks, which augment the training set with adversarial examples to increase the model's robustness against a specific attack. Adversarial training improves the classification accuracy of the target model on adversarial examples.

Simple Random Sampling. In statistics, a simple random sampling, or shortly SRS, is a subset of individuals chosen from a larger set, where each sample is chosen randomly with the same probability. For $\mathbf{X}$ containing n points, we randomly sample r ($r < n$) points from $\mathbf{X}$ by

$$\mathcal{P}_i(\mathbf{X}) = \{\mathbb{1}_{\mathbf{x}} | \mathbf{x} \in \mathbf{X}, \mathbb{1}_{\mathbf{x}} \sim Ber(0.5)\}, \quad (4)$$

where $\mathbf{x}$ is sampled from $Bernoulli(0.5)$ distribution to indicate the existence of point $\mathbf{x}$ in the post-sampled set.

As described in [37], Gaussian noising and quantification are another two basic defenses for point clouds, which are similar to defenses for image adversarial examples.

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 $\tilde{\mathbf{X}}$, which may either be an

original input $\mathbf{X}$, or an adversarial input $\mathbf{X}'$, the goal of a defense method is to either augment data to train a robust f' such that $f'(\tilde{\mathbf{X}}) = f(\mathbf{X})$, or transform $\tilde{\mathbf{X}}$ by a transformation $\mathcal{T}$ such that $f(\mathcal{T}(\tilde{\mathbf{X}})) = f(\mathbf{X})$.

Towards this goal, we propose a method formed by a denoiser and an upsampler, as shown in Figure 1, which adds an outlier-removal layer and an upsampling network to the front of the classification networks, to realize network robustness. These layers are designed in the context of point cloud classification on ModelNet40 [33] dataset and are used in conjunction with a trained classifier f. The defense function is denoted as $D : \mathbf{X}' \to \hat{\mathbf{X}}$, where $\hat{\mathbf{X}}$ denotes the denoised and upsampled point cloud. Inspired from upsampling network for generating a denser and uniform set of points [39], we define the loss function as

$$\mathcal{L}(\mathbf{X}, \hat{\mathbf{X}}) = \mathcal{L}_{rec}(\mathbf{X}, \hat{\mathbf{X}}) + \beta \mathcal{L}_{rep}(\mathbf{X}, \hat{\mathbf{X}}) + \gamma \|\boldsymbol{\theta}\|_2^2, \quad (5)$$

where $\mathcal{L}_{rec}$ is the reconstruction loss and $\mathcal{L}_{rep}$ the repulsion loss. $\boldsymbol{\theta}$ indicates the network parameters, β balances the reconstruction loss and repulsion loss, and γ denotes the multiplier of weight decay.

3.1. Statistical Outlier Removal (SOR)

Since there exists outliers in raw point cloud data produced by 3D scanners, Rusu *et al.* [27] propose statistical outlier removal method (SOR for short) which corrects these irregularities by computing the mean μ and standard deviation σ of nearest neighbor distances, and trim the points which fall outside the $\mu \pm \alpha \cdot \sigma$, where α depends on the size of the analyzed neighborhood.

We use k-nearest neighbors (kNN) for outlier removal. Specifically, the kNN point set of each point $\mathbf{x}_i$ of point cloud $\mathbf{X}$ is defined as $knn(\mathbf{x}_i, k)$. Then the average distance

d_i that each point $\mathbf{x}_i$ has to its kNN is denoted by

$$d_i = \frac{1}{k} \sum_{\mathbf{x}_j \in knn(\mathbf{x}_i, k)} \|\mathbf{x}_i - \mathbf{x}_j\|_2, \quad i = 1, \ldots, n. \quad (6)$$

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

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

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

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

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

We explore the relationships between outliers and adversarial points to explain why SOR is effective for defending C&W based attacks. Despite that the attackers successfully fool the classification network, there is always a certain percentage of points that inevitably become abnormal points and are then captured and dropped by SOR. C&W attack makes some points off the manifold of point cloud surface, and such 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

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

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

$$\mathbf{X}_{adv} = \{\mathbf{x}'_i | \|\mathbf{x}_i - \mathbf{x}'_i\|_2 > T(\mathbf{X}, \mathbf{X}', \epsilon)\}, \quad (11)$$

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

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

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

By Equation (10), we acquire the percentage of adversarial points p of SOR and SRS methods, and denote them by $p_{\mathbf{SOR}}$ and $p_{\mathbf{SRS}}$ respectively. It is expected that $p_{\mathbf{SOR}} > p_{\mathbf{SRS}}$
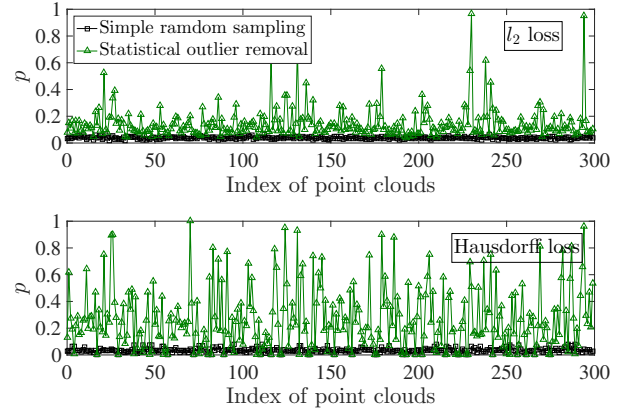

Figure 2: Comparison of $p_{\mathbf{SOR}}$ and $p_{\mathbf{SRS}}$ under l_2 and Hausdorff loss based targeted adversarial examples on PointNet network, respectively. The ratio ϵ is set with 0.04.
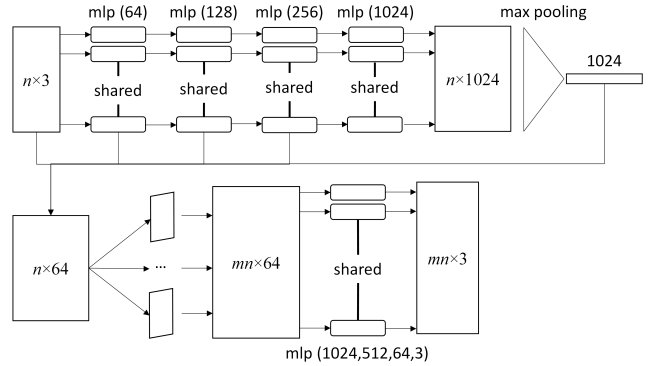

Figure 3: The network architecture of point cloud upsampling network (PU-Net).

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, as shown in Figure 2. Most of $p_{\mathbf{SOR}}$ of point clouds are larger than $p_{\mathbf{SRS}}$, implying that SOR drops more adversarial points than SRS. Thus SOR has a better ability of defense against adversarial point clouds than SRS.

3.2. Upsampling Network

Our goal is to defend a classification model f against the perturbed point clouds generated by an adversary. Our approach is motivated by the manifold assumption [44], which postulates that natural images or point clouds lie on low-dimensional manifolds. The perturbed point clouds are known to lie off the low-dimensional manifold of natural point clouds, which is approximated by deep networks. We

Algorithm 1 Denoise and upsample points as defense

Input: Point cloud $\mathbf{X}$, nearest neighbor number k, outlier truncation parameter α and network parameter $\boldsymbol{\theta}$

Output: Prediction label l

1: Initialize $\mathbf{X}' = \varnothing$
2: Compute the average distance d_i that each point $\mathbf{x}_i$ has to its nearest k neighbors by Equation (6)
3: Compute the mean $\bar{d}$ and standard deviation σ of all these distances by Equation (7) and (8)
4: **for** $i \leftarrow 0$ to n **do**
5:     **if** the average distance $d_i < \bar{d} + \alpha \cdot \sigma$ **then**
6:         $\mathbf{X}' = \mathbf{X}' \cup \mathbf{x}_i \ (\mathbf{x}_i \in \mathbf{X})$
7: The upsampled point clouds $\hat{\mathbf{X}}$ is generated by feeding $\mathbf{X}'$ into upsampling network
8: $\hat{\mathbf{X}}$ is fed into classification network $f(\hat{\mathbf{X}})$, and it outputs the prediction label l
9: **return** Prediction label l

---

propose to use point cloud upsampling to remap off-the-manifold adversarial samples on to the natural manifold to reconstruct surface. In this manner, robustness against adversarial perturbations is achieved by enhancing the visual quality of point clouds. This approach offers significant advantages over other defense mechanisms that truncate critical information to achieve robustness.

Since these perturbations are generally missing critical points from skeletons from point clouds, we use a point cloud upsampling network that output a denser point cloud that follows the underlying surface of the target object while being uniform in distribution. The network considered in this work is the Point Cloud Upsampling Network (PU-Net) [39], which learns geometry semantics of point-based patches from 3D models, and the architecture is illustrated in Figure 3. To address the uncertain presence of the varying part, the minimizer of Chamfer distance (CD) distributes some points outside the main body at the correct locations; while the minimizer of Earth Mover's distance (EMD) is considerably distorted [8]. We also try EMD [8] loss to observe the defense performance:

$$\mathcal{L}_{rec} = D(\mathbf{X}, \hat{\mathbf{X}}) = \frac{1}{\|\hat{\mathbf{X}}\|_0} \sum_{\mathbf{x}' \in \hat{\mathbf{X}}} \min_{\mathbf{x} \in \mathbf{X}} \|\mathbf{x} - \mathbf{x}'\|_2^2. \qquad (13)$$

The total loss function combines the reconstruction loss $\mathcal{L}_{rec}$ and repulsion loss $\mathcal{L}_{rep}$ proposed in [8]. In contrast, we use a simple upsampling method [2], which interpolates points at vertices of a Voronoi diagram for comparative trial.

In summary, we have given a formal definition of the proposed defense in Algorithm 1.

4. Experiments

4.1. Experimental Setup

Dataset. We utilize ModelNet40 [33] datasets to test our proposed DUP-Net, which contains 12,311 CAD models from 40 object categories, where 9,843 objects are used for training and the other 2,468 for testing. As done by Qi *et al.*, before generating adversarial point clouds using [39, 34] methods, first uniformly sample 1,024 points from the surface of each object and rescale them into a unit cube. We also use Visionair dataset collected by [39] for training DUP-Net, which contains 60 different models from the Visionair repository [1], ranging from smooth non-rigid and steep rigid objects.

Networks and Implementation Details. We use Point-Net [25] and PointNet++ [26] as targeted classification networks and train the models using default settings. To train the proposed DUP-Net, for ModelNet40, the upsampled point number is 2048 and the upsampling rate is 2; for Visionair dataset, the number is 4096 and upsampling rate is 4. Each input sample has $n = 1024$ points for both training procedures. The balancing weights β and γ are set as 0.01 and 10^{-5}, respectively. The implementation is based on TensorFlow. For the optimization, we train the network for 120 epochs using the Adam [14] algorithm with a mini-batch size of 28 and a learning rate of 0.001.

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.

4.2. Ablation Study

Our proposed defense first deploys the SOR layer, which aims to minimize the effect of outlier points perturbations, followed by an upsampling network to selectively introduce missing points into a point cloud and recover off-the-manifold attacked point clouds.

SOR as Defense. We compare the detection accuracy and attack success rate of targeted attacks of proposed SOR defense with baseline SRS for C&W and shifting [34] based attacks and dropping [42] based attacks on PointNet. Gaussian noising and quantification are not considered as defense because they will degrade accuracy of clean samples.

As shown in Figure 4a, for SRS baseline, as dropped points increases, the attack success rate drops dramatically, and the accuracy gradually increases with its maximum 65.1%; for clean examples, the accuracy is monotonically decreasing. The tendency of three curves can be explained: the attacks search the entire point cloud space for adversarial perturbations without considering the location of the
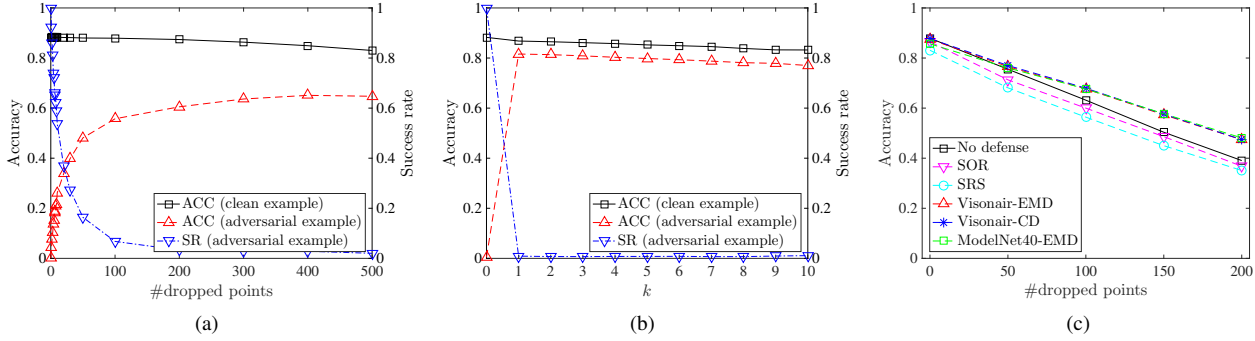
Figure 4: (a) SRS defense performance of clean and targeted adversarial point clouds on PointNet using C&W and shifting loss; (b) SOR defense performance of clean and targeted adversarial point clouds on PointNet using C&W and shifting loss under $\alpha = 1.1$; (c) Defense performance of adversarial point clouds on PointNet with or without defense.

Models	Target [4]	Defense (SRS)	Defense (SOR) (ours)	Defense (PU-Net) (ours)	Defense (DUP-Net) (ours)
Clean point cloud	**88.3%**	83.0%	86.5%	87.5%	86.3%
Adv (C&W + l_2 loss) [34]	0.7%	64.7%	81.4%	23.9%	**84.5%**
Adv (C&W + Hausdorff loss) [34]	12.7%	58.8%	59.8%	17.6%	**62.7%**
Adv (C&W + Chamfer loss) [34]	11.8%	59.5%	59.1%	18.0%	**61.4%**
Adv (C&W + 3 clusters) [34]	0.7%	**92.0%**	-	-	87.6%
Adv (C&W + 3 objects) [34]	2.7%	**92.4%**	-	-	68.4%
Adv (dropping 50 points) [42]	75.5%	68.1%	71.3%	**76.1%**	73.9%
Adv (dropping 100 points) [42]	63.2%	56.4%	60.0%	**67.7%**	64.3%
Adv (dropping 150 points) [42]	50.4%	45.0%	48.6%	**57.7%**	54.4%
Adv (dropping 200 points) [42]	39.1%	35.1%	36.8%	**48.1%**	43.7%

Table 1: Classification accuracy under the white-box attack on PointNet. For the SRS defense model, number of random dropped points is 500 and for SOR defense model, $k = 2$ and $\alpha = 1.1$ are set as hyperparameters.

Models	CW l_2	CW Hausd	Drop 200
Target [26]	0%	28.1%	56.4%
Defense (SRS)	66.7%	51.7%	47.3%
Defense (DUP-Net)	**75.7%**	**54.1%**	**61.9%**

Table 2: Comparison of classification accuracy using SRS and proposed DUP-Net under PointNet++ network.

point cloud content, which is contrary to the classification models that pay attention to object shapes [38]. Therefore, dropping a few points with SRS erases the artifact bothered by adversarial perturbation, which promotes defense of adversarial point clouds. The structure of point cloud is still preserved with a few points dropped; when more random sampled points are dropped, the shape of the point cloud deteriorates and thus the classification accuracy degrades.

As shown in Figure 4b, the SOR operation comprises

two influential factors, k the number of neighbor points and α the percentage of points that are regarded as outliers. When $k = 0$, the kNN point set only contains the point itself, thus the statistical removal defense is noneffective; when $k \geq 1$, the defense works. 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 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.

For the vanilla attack scenario, the attackers are not aware of the points-removal layer, and directly use the original networks as the target model to generate adversarial examples. For reading convenience, we coin a new acronym "adv" standing for "adversarial point clouds" in tables. As shown in Table 1, the points-removal layer can mitigate the adversarial effects on C&W methods significantly. As for

Networks	Task	Shifting (l_2) [34]	Adding (Hausdorff) [34]	Adding (Chamfer) [34]	Dropping 100 points [42]	Dropping 200 points [42]
PointNet [25]	Target	0.7%	12.7%	11.8%	63.2%	39.1%
PointNet++ [26]	Target	**89.5%**	52.9%	51.0%	**82.4%**	**75.6%**
	Defense (SRS)	82.9%	**59.6%**	**58.3%**	70.3%	54.5%
	Defense (DUP-Net)	84.3%	48.3%	48.5%	75.2%	67.3%
DGCNN [32]	Target	**91.2%**	**51.4%**	**50.8%**	**75.5%**	**67.9%**
	Defense (SRS)	68.2%	38.2%	37.1%	72.2%	54.9%
	Defense (DUP-Net)	40.7%	25.5%	26.7%	32.0%	26.7%

Table 3: Black-box attacks and defenses: accuracy of targeted C&W and shifting and adding based adversarial point clouds and saliency map and points dropping based adversarial point clouds generated from PointNet on other classification networks [26, 32, 11] with or without defense.

l_2 metric, SOR has the best performance with 81.4% accuracy. For Hausdorff and Chamfer loss metrics, SOR and SRS have similar results with lower accuracy.

To validate whether differentiable point removal layer is able to simulate non-differentiable layer, We train the modified PointNet with dropping probability $p = 0.5$ before the max pooling layer, and conduct white-box attacks. The classification accuracy of C&W l_2, C&W Hausdorff and drop 200 attacks are 0%, 0% and 54.5%, respectively, which implies that the differentiable random point removal cannot simulate non-differentiate layer well.

PU-Net as Defense. Here we investigate the individual impact of PU-Net module towards defending adversarial attacks. Since C&W based attack can be defended by the proposed SOR layer, in this section we only consider saliency point dropping based attack proposed by Zheng *et al*. [42]. We perform extensive experiments on three upsampler networks: pretrained PU-Net model from [39] (Visonair-EMD), PU-Net with Chamfer distance loss trained by Visonair (Visonair-CD) and PU-Net with Earth Mover's distance trained by ModelNet40 (ModelNet40-EMD), as shown in Figure 4c. The results demonstrate that, upsampler network helps adversarial examples filling missing points that are critical for classification, especially, when dropped number is 200, the defense has nearly 9% increase. Besides, the upsampler network can be trained on a small dataset and defends well against adversarial attacks generated from other point cloud datasets with fine generalization ability. We find that the performance of the model trained by CD loss is similar with that trained by EMD loss w.r.t. adversarial example defense, which means that the selection of loss function does not affect classification accuracy. SOR and SRS both deteriorate the defense performance, because the dropping attacks visually damage the local shapes of point clouds, and the SOR/SRS operation further amplifies the distortion.

We also compare PU-Net with a simple upsampling method [2], which interpolates points at vertices of a Voronoi diagram. Experiments show that PU-Net performs much better than [2] by 8% when attacked by "Drop 200".

4.3. Evaluation Results of DUP-Net

The last column in Table 1 shows the overall defense accuracy of proposed DUP-Net against different attacks (C&W based points-shifting/points-adding/cluster-adding/object-adding [34] and point-dropping [42]) on PointNet. For clean point clouds, DUP-Net slightly reduce 2% detection accuracy; for C&W attacks, DUP-Net performs better than baseline SRS, proposed SOR and proposed PU-Net. The DUP-Net performs better than PU-Net, implying that the outlier removing operation is effective in promoting defense performance. For point-dropping attacks, DUP-Net defense is slightly worse than PU-Net alone but still better than baseline, which implies that large modifications made by attackers cause some local shapes to disappear to a large degree, and SOR defense further breaks critical skeleton information. For C&W cluster and C&W object attacks, our DUP-Net defense improves the accuracies to 87.6% and 68.4%; with SRS defense, the accuracies are 92.0% and 92.4%. These results further demonstrate our strong defense ability. Because DUP-Net removes outliers of the manifold surface and SRS equiprobably removes points, for these two tasks, SRS performs better.

Overall, DUP-Net as a preprocessing network help ensure the robustness of neural network based classification and resist attacks from adversarial point clouds. Besides, DUP-Net is non-differentiable which makes attacker difficult to implement secondary adversarial attacks.

4.4. Generability of DUP-Net

The transferability of targeted C&W based points-shifting and points-adding attacks and saliency map based points-dropping attack of PointNet on black-box classification systems is shown in Table 3. Similar to [34], we test the success rate of adversarial examples on PointNet++ and
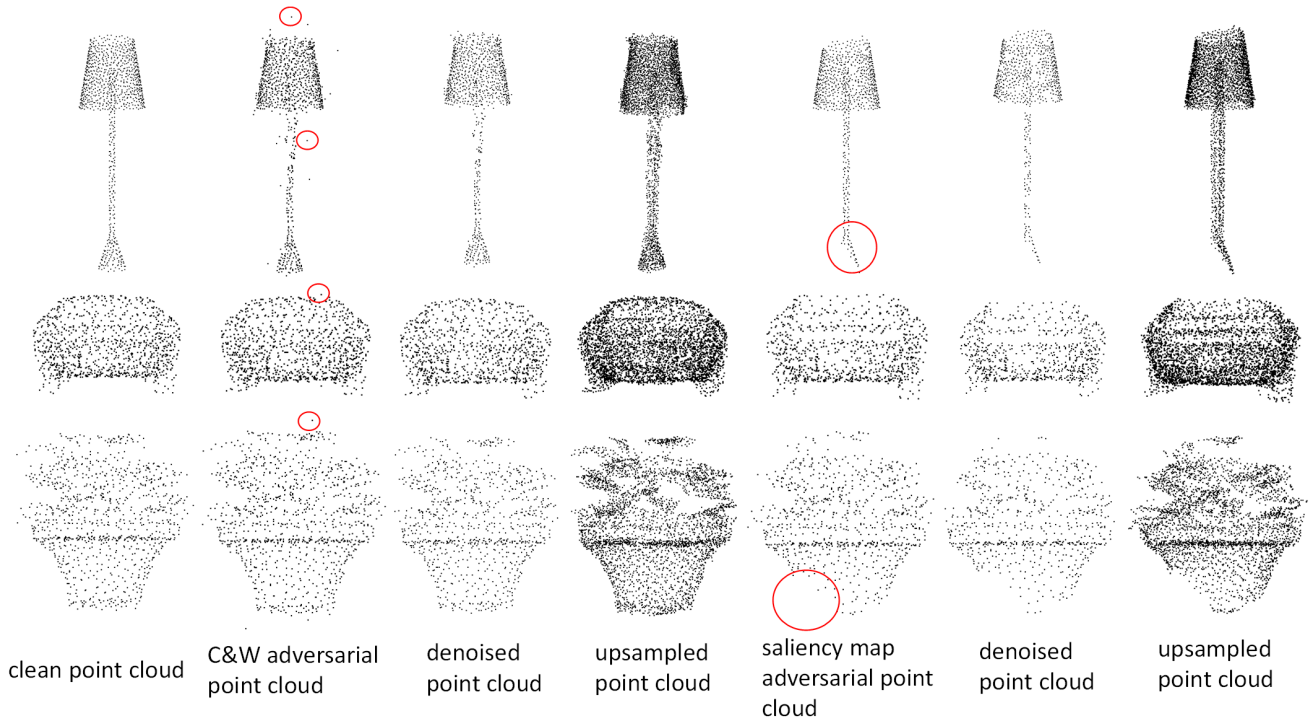
Figure 5: Visualization of point clouds. The fifth column is the 200 points dropping attack. Red circles denote outliers and missing object parts. Enlarge to see details.

DGCNN. The result illustrates that C&W based adversarial samples have limited transferability; while for points-dropping attacks, they are more transferable to other networks. We find that for black-box defense, adding a defense network (SRS layer or DUP-Net) before the classification network deteriorates the classification accuracy, thus our proposed defense scheme is only suitable for white-box attack, and in such cases, the defense sides do not need the preprocessing network. The results show that deploying DUP-Net does not improve defense ability against black-box attacks owning to the network structure dissimilarity.

In Table 2, we implement the experiments on Point-Net++, which further shows our proposed DUP-Net can be utilized for different target recognition models.

4.5. Visualization

Figure 5 shows the details of clean point clouds, adversarial point clouds and the denoised and upsampled point clouds. From top to bottom the classes of point clouds are "vase", "car" and "flower pot", respectively. It shows that C&W attacks produce outliers that lie far away from the manifold of points, while saliency map based attacks drop a cluster of points. We denote the outliers and dropped cluster by red circles. The SOR denoiser drops some outliers to neutralize the attack success rates of adversarial point

clouds, and then PU-Net strengths the smoothness of local region to assist the classification of models.

5. Conclusion

In this paper, we propose a denoiser and upsampler network (DUP-Net) formed by a statistical outlier removal (SOR) layer and a point cloud upsampling network (PU-Net) to defend against 3D point cloud adversarial examples. We propose to use point cloud restoration techniques to purify perturbed point clouds. As an initial step, we apply SOR to suppress any outlier based noise patterns and formulate a non-differentiable layer that is difficult to bypass. The central component of our approach is the upsampling operation, which enhances the point resolution while simultaneously removing adversarial patterns similar to image super-resolution operation. Our experiments show that point cloud upsampling network alone is sufficient to reinstate classifier beliefs towards correct categories; besides, the statistical outlier removal step provides added robustness since it is a non-differentiable denoising operation.

References

[1] Visionair. http://www.infra-visionair.eu. Accessed: 2019-03-20.

[2] Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David Levin, and Claudio T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on visualization and computer graphics*, 9(1):3–15, 2003.

[3] 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.

[4] R Qi Charles, Hao Su, Mo Kaichun, and Leonidas 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.

[5] Moustapha Cisse, Yossi Adi, Natalia Neverova, and Joseph Keshet. Houdini: Fooling deep structured prediction models. *arXiv preprint arXiv:1707.05373*, 2017.

[6] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppfnet: Global context aware local features for robust 3d point matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 195–205, 2018.

[7] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M Roy. A study of the effect of jpg compression on adversarial images. *arXiv preprint arXiv:1608.00853*, 2016.

[8] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017.

[9] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[10] Shixiang Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*, 2014.

[11] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 984–993, 2018.

[12] Felix Järemo Lawin, Martin Danelljan, Fahad Shahbaz Khan, Per-Erik Forssén, and Michael Felsberg. Density adaptive point set registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3829–3837, 2018.

[13] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3907–3916, 2018.

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[15] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[16] Daniel Liu, Ronald Yu, and Hao Su. Extending adversarial attacks and defenses to deep 3d point cloud classifiers. *arXiv preprint arXiv:1901.03006*, 2019.

[17] Jiayang Liu, Weiming Zhang, Yiwei Zhang, Dongdong Hou, Yujia Liu, Hongyue Zha, and Nenghai Yu. Detection based defense against adversarial examples from the steganalysis point of view. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4825–4834, 2019.

[18] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016.

[19] Jiajun Lu, Hussein Sibai, Evan Fabry, and David Forsyth. No need to worry about adversarial examples in object detection in autonomous vehicles. *arXiv preprint arXiv:1707.03501*, 2017.

[20] Dongyu Meng and Hao 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.

[21] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal 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.

[22] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram 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.

[23] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram 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.

[24] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 918–927, 2018.

[25] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas 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.

[26] 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.

[27] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Mihai Dolha, and Michael Beetz. Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927–941, 2008.

[28] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 2019.

[29] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[30] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble

adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

[31] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–67, 2018.

[32] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *arXiv preprint arXiv:1801.07829*, 2018.

[33] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

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

[35] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *Advances in Neural Information Processing Systems*, pages 1696–1704, 2016.

[36] Dawei Yang, Chaowei Xiao, Bo Li, Jia Deng, and Mingyan Liu. Realistic adversarial examples in 3d meshes. *arXiv preprint arXiv:1810.05206*, 2018.

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

[38] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

[39] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-net: Point cloud upsampling network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2790–2799, 2018.

[40] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in neural information processing systems*, pages 3391–3401, 2017.

[41] Stephan Zheng, Yang Song, Thomas Leung, and Ian 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.

[42] Tianhang Zheng, Changyou Chen, Kui Ren, et al. Learning saliency maps for adversarial point-cloud generation. *arXiv preprint arXiv:1812.01687*, 2018.

[43] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018.

[44] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.