ADVERSARIAL ATTACK BY LIMITED POINT CLOUD SURFACE MODIFICATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent research has revealed that the security of deep neural networks that directly process 3D point clouds to classify objects can be threatened by adversarial samples. Although existing adversarial attack methods achieve high success rates, they do not restrict the point modifications enough to preserve the point cloud appearance. To overcome this shortcoming, two constraints are proposed. These include applying hard boundary constraints on the number of modified points and on the point perturbation norms. Due to the restrictive nature of the problem, the search space contains many local maxima. The proposed method addresses this issue by using a high step-size at the beginning of the algorithm to search the main surface of the point cloud fast and effectively. Then, in order to converge to the desired output, the step-size is gradually decreased. To evaluate the performance of the proposed method, it is run on the ModelNet40 and ScanObjectNN datasets by employing the state-of-the-art point cloud classification models; including PointNet, PointNet++, and DGCNN. The obtained results show that it can perform successful attacks and achieve state-of-the-art results by only a limited number of point modifications while preserving the appearance of the point cloud. Moreover, due to the effective search algorithm, it can perform successful attacks in just a few steps. Additionally, the proposed step-size scheduling algorithm shows an improvement of up to 14.5% when adopted by other methods as well. The proposed method also performs effectively against popular defense methods.

1 INTRODUCTION

Deep Neural Networks (DNNs) have achieved considerable accuracy in various tasks (espicially in classification). However, several recent research has shown that DNNs are not reliable enough Carlini & Wagner (2017); Goodfellow et al. (2015); Madry et al. (2019); Naderi et al. (2021); Brown et al. (2018). Adding small perturbations in the input data can cause the DNN to misclassify them with high confidence Szegedy et al. (2014). This new input data is called an adversarial sample. It is a serious threat when it comes to safety-critical applications. Therefore, it is important to generate strong adversarial samples and study the DNN behavior against these adversarial samples. By considering the DNN behavior, the adversarial robustness of DNNs can be improved and more effective defenses can be constructed. Most attacks and defenses focus on 2D images Goodfellow et al. (2015); Carlini & Wagner (2017); Madry et al. (2019) They are still in the early stages on 3D data Liu et al. (2019b); Zhou et al. (2019); Yang et al. (2021). Paying attention on 3D data is interesting, because the world around us is a combination of 3D objects. In addition, 3D data has many applications in robotics, augmented reality, autopilot, and automatic driving. Thanks to presence of 3D sensors, such as various types of 3D scanners, LiDARs, and RGB-D cameras (such as Kinect, RealSense and Apple depth cameras), it is easier to capture 3D data. This paper proposes an untargeted 3D adversarial point cloud attack against point cloud classifiers; namely, PointNet, PointNet++ and DGCNN. The proposed attack, adds a few points while applying hard boundary constraints on the number of added points and on the point perturbation norms. By controlling the step-size, the generated adversarial sample yeilds to escape local optima and find the most appropriate attack. The rest of this paper is organized as follows. In Section 2, the related work is reviewed. The proposed 3D adversarial attack is introduced in Section 3. Experimental results are discussed in Section 4. Finally, Section 5 concludes the paper. In summary, the contributions of this work include:

- Proposing an adversarial attack method to perform effective attacks while preserving the point cloud appearance by applying two hard boundary constraints on the number of modified points and on the point perturbation norms.
- Proposing a learning rate scheduling algorithm to improve other existing methods in this setting.
- Managing to generate highly successful attacks with a small number of steps to perform fast and subtle attacks.

2 Related Work

2.1 Deep Learning on 3D Data

There are three strategies for 3D object classification including volume-based Wu et al. (2015); Maturana & Scherer (2015), multi-view-based Su et al. (2015); Yang & Wang (2019), and point cloud-based Qi et al. (2017a;b); Wang et al. (2019). This research focuses on point cloud-based models. As a pioneering work, the PointNet Qi et al. (2017a) can directly feed point clouds as its input. It achieves the features of each point independently and then aggregates them by max-pooling. It then extracts global features for 3D point cloud classification and segmentation tasks. An update of this work is the PointNet++. It improves the feature extraction through combined features from multiple scales in order to add locality to the PointNet. More recent work apply convolutions on neighborhood points to aggregate more local context Thomas et al. (2019); Hua et al. (2018); Wu et al. (2019); Li et al. (2018); Wang et al. (2019). For instace, DGCNN Wang et al. (2019) processes neighborhood points by applying EdgeConv to better capture local geometric structures of points and therefore achieves superior classification results. This paper uses the PointNet, PointNet++, and DGCNN architectures to evaluate the proposed attack in the case of 3D point cloud classification task.

2.2 Adversarial Point Clouds

Various studies have focused on adversarial attack on 3D point cloud classification. The adversarial attacks can be categorized into band-limited and unrestricted adversarial perturbations.

Band-limited approaches, apply the limitation on the perturbations in generated adversarial samples while preserving the point cloud appearance visually. Typical perturbation measurements include L2 norm, Chamfer distance, and Hausdorff distance. The band-limited attacks are divided into point adding, point shifting, and point dropping attacks. In terms of point adding, Xiang et al. Xiang et al. (2019) proposed three different targeted attacks by adding several point clusters, tiny objects, or extra points. These attacks optimize a Carlini & Wagner (C&W) function Carlini & Wagner (2017) and constraint point perturbation norm to push the added points towards the object surface. Yang et al. Yang et al. (2021) add a few points to the original point cloud based on the Fast Gradient Sign Method (FGSM) attack Goodfellow et al. (2015). They restrict both the number of added points and the point perturbation norm to generate imperceptible targeted adversarial samples. Liu et al. Liu et al. (2019a) add new points (sticks or line segments) into the original point cloud, where the sticks must arise from the object's surface. The position of each stick onto the object's surface and the number of points across the line segments are limited. In addition to generating point clouds by adding points into an original point cloud, both Zheng et al. Zheng et al. (2019) and Matthew et al. Wicker & Kwiatkowska (2019) iteratively drop points from the original point cloud to deceive the classifier. Also, Xiang et al. Xiang et al. (2019), Liu et al. Liu et al. (2019a;b), Yang et al. Yang et al. (2021), Tzungyu et al. Tsai et al. (2020), Hamdi et al. Hamdi et al. (2020), and Chengcheng et al. Ma et al. (2020) all propose adversarial attack based on point shifting methods. Most of those attacks extend 2D adversarial attacks Carlini & Wagner (2017); Goodfellow et al. (2015); Madry et al. (2019).

Another line of attacks focuses on unrestricted attacks, which are not limited to any distance criteria. These unlimited attacks do not necessarily look the same as the original point clouds. In other words, it is sufficient that the adversarial sample stays legitimate for the human eye but deceives the classifier. Applying isometry transformation on point cloud Zhao et al. (2020) and using trained Generative Adversarial Network (GAN) Zhou et al. (2020) to generate adversarial sample are some

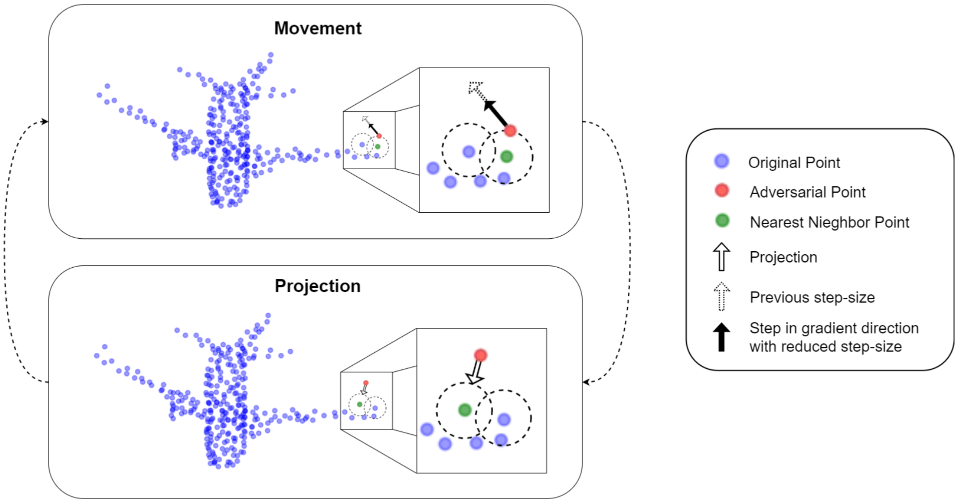

Figure 1: Demonstration of proposed Variable Step-Size Attack (VSA) method. At first, adversarial points take a step with reduced step-size in gradient direction. Then, points that fall out of the original points' boundaries are projected back. This process is repeated at each step.

research has been proposed to unrestricted attacks. Since unlimited attacks do not visually preserve the point cloud appearance, they are not discussed in this paper.

There are typical adversarial defense methods including Statistical Outlier Removal (SOR) Zhou et al. (2019) and saliency map removal Liu et al. (2019b), which discard outlier and saliency points, respectively. Also, Zhou et al. Zhou et al. (2019) propose a denoiser and upsampler network (DUP-Net) structure as defenses for the 3D classification task.

This paper proposes a untargeted attack by a point adding method that imposes hard boundary constraints on the number of added points and on the point perturbation norms. To the best of our knowledge, all previous attacks train with a fixed learning rate until objective function stagnates, but the proposed attack can escape local optima by controlling the learning rate. Furthermore, by imposing any constraint on point perturbation norms and number of added points it can find the most appearance preserving attacks.

3 Proposed Method

3.1 Problem Formulation

The objective is to generate an adversarial sample based on the original one, such that it deceives the model while retaining its appearance. Let $X = \{p_1, p_2, ..., p_k\}$ be the original point cloud, where $p_i \in \mathbb{R}^3$ represents the coordinates of the ith point. The adversarial sample $X^* = X \cup \delta_X$ is generated by adding point set δ_X to the original point cloud. The points in δ_X are denoted as adversarial points and the points in X are denoted as original points. To deceive the model, the proposed method needs to maximize the model's classification objective function $F(X)$. On the other hand, two hard boundary constraints are applied on the optimization problem to preserve the point cloud's appearance. More concretely, the proposed method is designed to solve the following optimization problem

$$\max_{\delta_X} F(X^*) \ s.t. \ dist(X^*, X) \leq \epsilon, \ |\delta_X| \leq n \tag{1}$$

where ϵ and n are hard boundaries on the perturbation norm and the number of added points, respectively. The function $dist$ measures the dissimilarity between the adversarial sample and the original one. This function is chosen to be the Hausdorff distance defined by

$$D_H(X,Y) = \max\{\max_{x_i \in X}\{\min_{y_j \in Y} ||x_i - y_j||_2\}, \ \max_{y_j \in Y}\{\min_{x_i \in X} ||y_j - x_i||_2\}\} \tag{2}$$

Algorithm 1 Variable Step-Size Attack

Input: Objective function F, point cloud $P = \{p_1, p_2, ..., p_k\}$
Parameter: Hausdorff distance boundary ϵ, initial step-size α_{init}, final step-size α_{final}, number of steps M, number of points n
Output: Adversarial sample P^*

```
$\epsilon^* = \epsilon$
Initialize $\delta_P$ using the $n$ points with the highest $||\nabla_{p_i} F(P)||_2$.
for $i = 1$ to M do
    Let $\alpha = \alpha_{init} + \frac{\alpha_{final} - \alpha_{init}}{\text{M}}$.
    Let $P_0$ and $P_1$ be subsets of $\delta_P$ with $||\nabla_{\delta_P} F(P \cup \delta_P)||_2$ equal to zero and non-zero respectively.
    $P_1 = P_1 + \alpha . \frac{\nabla_{P_1} F(P \cup \delta_P)}{||\nabla_{P_1} F(P \cup \delta_P)||_2}$
    $P_0 = P_0 + \alpha . v$ where $v$ is a random unit vector.
    $\delta_P = P_0 \cup P_1$
    for $p_i^* \in \delta_p$ do
        if $D_H(p_i^*, P) > \epsilon^*$ then
            $p_i^* = \text{NN}(p_i^*) + \epsilon^* . \frac{p_i^* - \text{NN}(p_i^*)}{||p_i^* - \text{NN}(p_i^*)||_2}$ where NN is the nearest neighbor in $P$
        end if
    end for
end for
return $P^* = P \cup \delta_P$
```

where x_i and y_j are points from point clouds X and Y, respectively. This function limits each point from δ_X to be in the ϵ neighborhood of its closest neighbor from X. In other words, all the added points must be close to the surface of the original point cloud.

Although the applied constraints preserve the adversarial sample's appearance, they make the search for the optimal δ_X much more difficult, especially in a first-order method. The Hausdorff distance constraint limits the movement of the added points which might lead the optimization process to get stuck in a bad local maxima. Moreover, due to the low number of added points which leads to low search space dimension, the optimization problem landscape is very non-concave Engstrom et al. (2017). The proposed methods in the following sections overcome these problems to produce better results using first-order methods.

3.2 Variable Step-Size Attack

The main proposed method uses the Projected Gradient Descent (PGD) Madry et al. (2019) algorithm with high step-size at the beginning of the algorithm, to solve Equation 1. High step-size lets the added points to explore the whole surface of the point cloud efficiently, while giving them the possibility to escape from local maxima. To converge to the desired result, the step-size is gradually reduced throughout the algorithm.

The algorithm is summarized in Algorithm 1. Suppose that P is the original point cloud. First, δ_P is initialized using the points of P with the highest gradient norms. These points have the most effect on the classification objective function and are thus a good initialization. At each step, the points in δ_P take a step in the gradient direction, and in a random direction if their respective gradient norm is zero. At the end of each step, each point in δ_P is projected into the ϵ^* neighborhood of its respective nearest neighbor from P. The step-size is reduced at each step to ensure the algorithm's convergence. For further insights, the algorithm is demonstrated in Figure 1.

3.3 Variable Boundary Attack

An alternative way to overcome the problem of local maxima is to vary the ϵ^* parameter. In this method, the ϵ^* parameter is set high at the beginning of the algorithm and is reduced to the desired final value ϵ, during the algorithm. By having a relaxed constraint for the first steps of the algorithm, it is easier to find solutions with a high objective function value. By proceeding throughout the algorithm, the solutions found with higher ϵ^* values serve as good initialization points for lower ϵ^* values. This finally leads to a better solution for the desired ϵ value at the end of the algorithm, compared to using the PGD algorithm with a constant ϵ^* and α parameter.

In this algorithm, compared to VSA, the hyperparameters α_{init} and α_{final} are replaced with initial boundary ϵ_{init} and step-size α. Moreover, instead of α, ϵ^* is updated using $\epsilon^* = \epsilon_{init} + \frac{\epsilon - \epsilon_{init}}{\mathrm{M}}$.

4 Experimental Results

In this section, the proposed method, which is denoted by VSA, is evaluated and compared with other methods to demonstrate its effectiveness. The experiments are carried out on three state-of-the-art point cloud processing architectures run on two benchmark datasets. The proposed method surpasses other state-of-the-art methods in attacking deep point cloud models when using a limited number of points. Moreover, it is shown that the proposed step-size scheduling algorithm can be adopted by existing methods to achieve higher results. The effectiveness of the proposed method against defense methods is also discussed in this section. Moreover, An ablation study is carried out to compare different variants of the proposed methods and to explore the effects of different hyperparameters. At the end, the generated samples of methods are visualized and compared in terms of perceptibility of adversarial points.

4.1 Experimental Setup

4.1.1 Baselines

The state-of-the-art methods in the scope of the discussed problem, which include the Point-Attach Method (PAM) in Yang et al. (2021) and the Adversarial Sticks Method (ASM) in Liu et al. (2019a), are employed. In ASM, the farthest point sampling is avoided and new points are sampled onto the adversarial sticks, for it to be used as a point addition method.

4.1.2 Datasets and Architectures

The main experiments are carried out against three popular models; namely PointNet Qi et al. (2017a), PointNet++ Qi et al. (2017b), and DGCNN Wang et al. (2019). The benchmark datasets used for these experiments are ModelNet40 and ScanObjectNN. The ModelNet40 dataset is used for 3D CAD model classification. The training split of the dataset with $9,843$ samples is used to train the models and the test split with $2,468$ samples is used to evaluate the attack methods. On the other hand, the ScanObjectNN which is a real-world dataset consisting of indoor 3D objects is divided into a training split with $2,309$ samples and a test split with 581 samples. The experiments are carried out on ModelNet40 against PointNet, if not mentioned otherwise. In all of the experiments the original point clouds have $k = 1,024$ points and are normalized according to Qi et al. (2017a).

4.1.3 Hyperparameters

All the hyperparameters are initialized according to this section unless mentioned otherwise. For the VSA method, α_{init} is set to 0.1 and α_{final} is set to $\min\{\frac{0.5}{n}, \frac{\epsilon}{2}\}$. By this, if n is low, the adversarial points spend more time exploring the surface of the point cloud which benefits them since they only cover a small portion of the surface and might need time to reach to the optimal solution. Note that according to the observations, the adversarial points tend to distance from each other when proceeding towards the optimal solution. This is because for the studied models in this paper, it is observed that when two points get too close to each other, one overshadows the other's contribution to the classification objective function. Since the points with the most impact on the classification objective function are chosen as the adversarial points' initialization, they tend to be distanced from each other too. Therefore, if n is high, less time is needed to search the point cloud surface since the points already cover the majority of the point cloud surface, which makes them more probable to be close to the optimal solution.

For the Variable Boundary Attack (VBA) method, ϵ_{init} is set to 2ϵ and α is set to $\min\{\frac{1}{n}, \epsilon\}$. The ϵ_{init} should be low enough to propose a solution similar enough to the optimal one at the final ϵ, and it should be high enough to solve the problem of local maxima to a certain extent. This makes

Table 1: Attack success rates (in percentage) on ModelNet40 against PointNet, PointNet++, and DGCNN. Success rates are reported for every pair of constraint boundaries $(\epsilon,\ n)$.

Model	Method	(0.05, 25)	(0.05, 100)	(0.1, 25)	(0.1, 100)
PointNet	PAM	2.5	11.3	8.5	36.9
	ASM	6.7	17.8	17.1	33.9
	VSA	**36.5**	**77.6**	**88.5**	**99.5**
PointNet++	PAM	3.2	10.7	7.8	22.6
	ASM	1.9	5.5	9.9	17.8
	VSA	**21.0**	**52.5**	**56.5**	**97.0**
DGCNN	PAM	3.4	8.0	5.1	15.2
	ASM	2.7	3.5	2.8	4.8
	VSA	**12.6**	**29.3**	**28.7**	**60.0**

Table 2: Attack success rates (in percentage) on ScanObjectNN against PointNet. Success rates are reported for every pair of constraint boundaries $(\epsilon,\ n)$.

Method	(0.05, 25)	(0.05, 100)	(0.1, 25)	(0.1, 100)
PAM	9.1	28.8	62.3	91.1
ASM	32.8	46.7	59.1	72.5
VSA	**59.0**	**84.7**	**95.1**	**99.4**

2ϵ an appropriate initialization for ϵ_{init}. For α, an initialization method similar to that of α_{final} of VSA is chosen due to the reasons discussed in the previous paragraph. The number of steps M is set to 500 for both methods.

4.2 Obtained Results

All the attack success rates are reported in percentage. The attack success rates against the models trained on ModelNet40 and ScanObjectNN are reported in Tables 1 and 2, respectively. The reported results are against PointNet, PointNet++, and DGCNN in Table 1 and against PointNet in Table 2. The experiments were repeated for different pairs of constraint boundaries (ϵ, n). Note that the nearest neighbor distance mean for the points in original point clouds (after normalization) is around 0.05. As such, $\epsilon \in \{0.05, 0.1\}$ makes the adversarial points to stay near the point cloud surface. As reported in these tables, the proposed method outperforms other state-of-the-art methods by a large margin. It can be seen that PointNet and PointNet++ are very vulnerable against the proposed method. They almost missclassify every given sample when attacked by adding less than 10% of the points, near the point cloud surface. In contrast, DGCNN performs much better against attack methods and is more challenging. Despite this, the proposed method manages to deceive this model 60% of times. Moreover, as shown in Table 2, the proposed method manages to generate subtle adeversarial samples with high accuracy on a real world dataset. This shows the effectiveness of the proposed method in real-world settings.

4.2.1 Adoption by Existing Methods

Due to the effectiveness of the proposed method, it can be adopted by other methods as well. To assess its effect, the step-size scheduling algorithm was used on PAM and ASM. For ASM, the learning rate was initialized with 2 instead of 0.1, and was reduced to 0.01 throughout the algorithm. That method is denoted as ASM+. For PAM, the step-size is initialized with $\frac{\epsilon}{2}$ and is divided by 2 at the end of each step. This version of PAM is denoted as PAM+.

The attack success rates of these methods and their enhanced versions are shown in Table 3. The results are reported for different pairs of constraint boundaries (ϵ, n). As shown in this table, the proposed step-size scheduling algorithm is able to improve the existing algorithms for every pair of constraint boundaries. The improvements are more significant for $\epsilon = 0.1$, especially for the ASM

Table 3: Attack success rates (in percentage) of existing methods and their step-size scheduling enhanced counterparts. Success rates are reported for different pairs of constraint boundaries (ϵ, n) on ModelNet40 against PointNet.

Method	(0.05, 25)	(0.05, 100)	(0.1, 25)	(0.1, 100)
PAM	2.5	11.3	8.5	36.9
PAM+	**3.7**	**12.8**	**10.7**	**42.7**
ASM	6.7	17.8	17.1	33.9
ASM+	**10.5**	**24.4**	**30.8**	**48.4**

Table 4: Attack success rates (in percentage) against outlier removal defense and salient point removal defense. Success rates are reported for every pair of constraint boundaries (ϵ, n) on ModelNet40 against PointNet. Best success rate for each number of points n is chosen for comparison.

Defense	Method	(0.025, 25)	(0.05, 25)	(0.1, 25)	25	(0.025, 100)	(0.05, 100)	(0.1, 100)	100
SOR	PAM	4.4	5.5	4.0	5.5	6.8	7.7	3.6	7.7
	ASM	5.2	6.8	6.5	6.8	2.9	11.9	18.1	18.1
	VSA	13.0	13.0	2.8	**13.0**	28.0	31.8	9.5	**31.8**
SPR	PAM	1.9	2.5	1.6	2.5	1.6	2.3	2.3	2.3
	ASM	2.3	3.7	7.1	**7.1**	5.4	6.5	13.7	**13.7**
	VSA	4.2	1.1	4.6	4.6	7.1	8.2	8.5	8.5

algorithm where an improvement of 14.5% is made with $(0.1, 100)$ as parameters. The difference in improvement between ASM and PAM is most likely due to the usage of projection in ASM's algorithm, which leads to a more effective search when paired with the step-size scheduling algorithm. Overall, this shows the impact of the proposed search strategy on other algorithms, in the scope of this problem.

4.2.2 Robustness Against Defense Methods

In this section, the proposed method is evaluated against statistical outlier removal (SOR) defense and salient point removal (SPR) defense. For the outlier removal defense, the 10 nearest neighbor average distance is calculated for each point. The points that have an average distance of greater than one standard deviation from the mean of this statistic are removed. For the salient point removal, the 200 points with the highest saliencies are removed.

The attack success rates against these two defense methods are reported in Table 4. For each number of points $n \in \{25, 100\}$, different values for $\epsilon \leq 0.1$ were tested and the best result was reported. The SOR defense is more challenging when the number of points is low. This comes from the fact that the mean of the statistic calculated for the original point cloud does not change drastically when the adversarial points are added. Despite of its challenges, the proposed method manages to evade the defense by adding the adversarial points very close to the point cloud surface with $\epsilon \in \{0.025, 0.05\}$, while having a high classification objective function. This is why it outperforms other methods against SOR. However, it is not as effective as ASM when it comes to SPR. This is because in ASM, when the salient points which are usually on the head of the sticks are removed, there are other close adversarial points that will replace the head of the sticks and make an successful attack.

4.3 Ablation

In this section, different variants of the proposed methods are compared to each other and different aspects of the proposed method are explored and evaluated. The simplest variant of the proposed method is the PGD algorithm, where the step-size α is constant compared to VSA. In this method, α is initialized according to VBA. A more complicated version of the proposed methods is a method comprising both of their ideas, denoted as VBA + VSA. In this method, the ϵ^* variable is scheduled

Table 5: Attack success rates (in percentage) of the proposed methods and their variants. Success rates are reported for every pair of constraint boundaries (ϵ, n) on ModelNet40 against PointNet.

Method	(0.05, 25)	(0.05, 100)	(0.05, 400)
PGD	32.6	67.0	86.4
VBA	34.2	69.8	87.5
VSA*	30.5	70.1	90.6
VSA	36.5	**77.6**	93.9
VBA + VSA	**37.4**	76.4	**94.5**

Table 6: Attack success rates (in percentage) for different number of steps values M and different number of points values n with $\epsilon = 0.05$. Success rates are reported on ModelNet40 against PointNet.

n	M					
	25	50	100	300	500	800
25	28.7	31.6	33.4	36.4	36.5	38.6
50	49.2	52.1	54.6	59.0	59.5	60.0
100	67.0	71.1	73.9	76.4	77.6	77.6
200	80.0	83.3	85.1	88.3	88.4	88.4
400	87.3	89.6	91.1	92.8	93.9	93.8

according to VBA and α is scheduled according to VSA. Moreover, a variant of VSA where a_{init} is not set high (it is set to 0.025, 0.02, and 0.01 for n equal to 25, 100, and 400, respectively) is also considered. α_{final} is set to 0.01 for $n = 25$ in that method. The attack success rates for different variants of the proposed methods are reported in Table 5. The experiments were carried out for different pairs of constraint boundaries. As shown in this table, VBA outperforms PGD by solving the problem of local maxima to a certain extent. Moreover, the methods improved by the proposed step-size scheduling algorithm outperform the other methods including VSA* to a large extent. It can be seen that α_{init} needs to be set high for VSA to work effectively, especially when n is set low. The improvement made by the step-size scheduling algorithm is due to a number of factors, like escaping improper local maxima, being able to explore the point cloud surface better, and converging to the desired result at the end of the algorithms. Between the algorithms that employ the proposed step-size scheduling, VBA + VSA performs slightly better. However, it is observed that VSA performs slightly better in Table 6 experiments when compared to VBA + VSA. Due to its slightly better performance and its simplicity, the VSA was chosen as the main proposed method.

Table 6 contains attack success rates for different values of M and n with $\epsilon = 0.05$. It can be seen that the proposed method performs very effectively, even when the number of steps is as low as 25. This makes the proposed method ideal for fast and subtle attacks. Note that higher number of steps lets the adversarial points to better explore the point cloud surface, though its impact slowly starts to decrease as the number of steps increases. Moreover, when the number of points is higher, the algorithm gets less affected by decreasing the number of steps, which is due to the effective initialization discussed in Section 4.1.3. To explore whether the local maxima problem is tackled or the improvements of VSA are merely due to the high number of steps and better exploration of the point cloud surface, the results in Table 6 are compared with Table 5. Consider $n \in \{25, 100\}$ where the local maxima problem is worse. By comparing the success rates of $(M = 100, n = 25)$ and $(M = 100, n = 100)$ in Table 6 to $(\epsilon = 0.05, n = 25)$ and $(\epsilon = 0.05, n = 100)$ of the VSA* method in Table 5 respectively, it can be seen that the respective success rates are higher. However, the respective traversed distance per points are lower. This shows that this improvement is due to solving the local maxima problem.

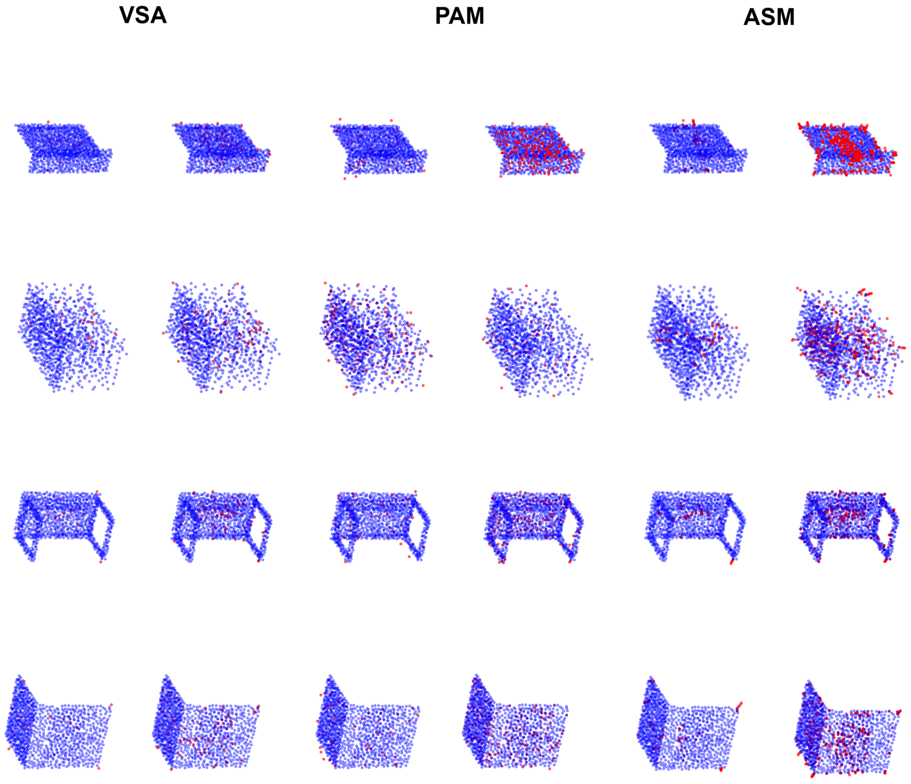

Figure 2: Adversarial Samples generated by each method with (ϵ, n) set to reach 25% attack success rate. For each method, first column represents low n and high ϵ and second column represents high n and low ϵ. Red points represent adversarial points and Blue points represent original points.

4.4 Visualization

In this section, the perceptibility of the generated adversarial sample is explored. For this, the attack success rate is fixed on 25% and $(\epsilon \in [0.025, 0.1], n \in [25, 400])$ are chosen for each of the methods to reach this success rate threshold. The generated samples are shown in Figure 2. Since there is a trad-off between n and ϵ to increase success rate, the first column for each method contains samples with low n and high ϵ and the second column contains samples with high n and low ϵ.

As shown in Figure 2, the adversarial points are far less perceptible in the proposed method compared to existing methods. Moreover, it is more outlier free due to its ability to perform successful attacks with low ϵ.

5 Conclusion

This paper proposed a new attack method to generate effective adversarial attacks by adding a limited number of points to the point cloud surface. The method introduced an effective step-size scheduling algorithm to overcome the local maxima problem and to explore the point cloud surface efficiently. The results showed that in addition to achieving state-of-the-art results, it can be adopted by other existing methods to improve their results. It also showed that the proposed method performs well against the SOR defense which is more challenging when the number of added points is low. Overall, this shows that 3D deep learning models are vulnerable against subtle yet effective attacks. By these observations, future work could investigate different step-size scheduling algorithms and their effects on the performance of first order attacks.

REFERENCES

Tom B Brown, Nicholas Carlini, Chiyuan Zhang, Catherine Olsson, Paul Christiano, and Ian Goodfellow. Unrestricted adversarial examples. *arXiv preprint arXiv:1809.08352*, 2018.

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

Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness, 2017.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015.

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

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*, pp. 984–993, 2018.

Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31:820–830, 2018.

Daniel Liu, Ronald Yu, and Hao Su. Adversarial point perturbations on 3d objects. *arXiv e-prints*, pp. arXiv–1908, 2019a.

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)*, pp. 2279–2283. IEEE, 2019b.

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*, pp. 1819–1827, 2020.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019.

Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 922–928. IEEE, 2015.

Hanieh Naderi, Leili Goli, and Shohreh Kasaei. Generating unrestricted adversarial examples via three parameters, 2021.

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*, pp. 652–660, 2017a.

Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space, 2017b.

Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pp. 945–953, 2015.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014.

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/CVF International Conference on Computer Vision*, pp. 6411–6420, 2019.

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, pp. 954–962, 2020.

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.

Matthew Wicker and Marta Kwiatkowska. Robustness of 3d deep learning in an adversarial setting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11767–11775, 2019.

Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9621–9630, 2019.

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*, pp. 1912–1920, 2015.

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

Jiancheng Yang, Qiang Zhang, Rongyao Fang, Bingbing Ni, Jinxian Liu, and Qi Tian. Adversarial attack and defense on point sets, 2021.

Ze Yang and Liwei Wang. Learning relationships for multi-view 3d object recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7505–7514, 2019.

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*, pp. 1201–1210, 2020.

Tianhang Zheng, Changyou Chen, Junsong Yuan, Bo Li, and Kui Ren. Pointcloud saliency maps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1598–1606, 2019.

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/CVF International Conference on Computer Vision*, pp. 1961–1970, 2019.

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