IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 December 2022, accepted 31 December 2022, date of publication 5 January 2023, date of current version 10 January 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3234313

RESEARCH ARTICLE

Point Cloud Adversarial Perturbation Generation for Adversarial Attacks

FENGMEI HE[1,2], YIHUAI CHEN[2,3], RUIDONG CHEN[4], AND WEIZHI NIE[4], (Member, IEEE)

[1]Department of Automation Electrical Engineering, Tianjing University of Technology and Education, Tianjin 300222, China
[2]College of Information Technology, Wenzhou Vocational College of Science and Technology and Education, Wenzhou Academy of Agricultural Sciences, Wenzhou 325006, Zhejiang, China
[3]Teacher Teaching Development Center, City University of Wenzhou, Wenzhou 325000, Zhejiang, China
[4]The School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China

Corresponding authors: Fengmei He (hefengmei@wzvcst.edu.cn), Yihuai Chen (cyhuai@163.com), and Ruidong Chen (chenruidong@tju.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFB1711704, and in part by the National Natural Science Foundation of China under Grant 62272337.

ABSTRACT In recent years, 3D model analysis has made a revolutionary development. Point cloud contains rich 3D object geometry information, which is an important 3D object data format widely used in many applications. However, the irregularity and disorder of the point cloud also cause its vulnerability to environmental impact, which may bring security risks to safety-critical 3D applications such as self-driving tasks. Recently, there are only a few methods engaged to attack the point cloud models to improve the robustness of point cloud analysis models. Most of them only focus on the attack by adjusting the points but ignore learning the perturbation's distribution characteristics. In this work, we propose a novel framework to attack point cloud models. By introducing the GAN structure, we train a generator to produce slight point-to-point perturbations according to the sample's raw classification, which can effectively boost the attack performance. Meanwhile, we propose an outlier removal module to constrain the magnitude of the generated perturbation. The goal is to guarantee the visual quality of generation samples to improve the difficulty of training and further improve the robustness of 3D analysis models. Finally, we carry out extensive attack experiments, and the related results demonstrate the effectiveness of our proposed method.

INDEX TERMS 3D model, point cloud, adversarial attack, generative adversarial network.

I. INTRODUCTION

With the rapid development of sensor technology such as LiDAR, the point cloud can be easily captured from real space. Point cloud contains rich 3D object geometry information, and it has been widely studied in the field of 3D analysis. Recently, some excellent works have been proposed to make full use of point cloud data [1], [2], [3], [4]. By constructing deep neural networks, these works are devoted to making full use of the geometric and structural information of the point cloud, and their works have been widely used in many real tasks such as self-driving [5], [6], scene reconstruction [7], [8] and 3D recognition works [9].

The associate editor coordinating the review of this manuscript and approving it for publication was Jiachen Yang.

However, recent studies have shown that DNNs have vulnerability in facing adversarial samples, the 3D models also inherit this shortcoming, that is the point cloud models also tend to be susceptible to adversarial attacks. In real application scenarios, the point cloud collection may receive the environmental impact. If the 3D model is not robust enough to such natural attacks, it's likely to bring serious risks, especially for some safety-critical applications such as auto-driving tasks. So, it's serious and necessary to learn to improve the adversarial robustness of point cloud models, which also leads to the need of learning point cloud adversarial attacks and study the mechanism of attack formation.

3D-adv [10] is the first one to study 3D point cloud adversarial attack, which mainly proposes two approaches: shift existing points or add a small number of point clusters with meaningful shapes negligibly. Reference [11] transfer the

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

idea of attacking the image analysis networks into the point cloud attacking methods, they used iterative gradient-based method to make perturbation. Reference [12] attack point cloud by constructing a gradient-based salience map and dropping the points with lower salience scores. Reference [13] propose a generation-based method LG-GAN to generate adversarial samples, which can generate adversarial samples directly through the input point cloud. These prior works have achieved some good results in some aspects like attack success rate, but they still have some limitations, and the main points are as follows: 1) The optimization-based and gradient-based methods lack learning the distribution characteristics of perturbation, thus causing the poor flexibility of attack operations, specifically reflected in the time consumption and transferability of the attack. 2) In terms of the visual quality of adversarial samples, some of the existing methods will cause the adversarial points to be too discrete. From the perspective of attack performance, the excessive outliers in their adversarial samples may cause the attack mechanism easy to be defended by simple statistical point removal operations. Although the generation-based can obtain smoother surface samples, it will also cause shape distortion and destroy the geometric properties of the point cloud itself.

A. MOTIVATION

From the above analysis, we conclude that the existing point cloud attack methods still have limitations, and we summarize some improvements that need to be made. 1) Instead of just adjusting the position of adversarial points, we hope to explore the distribution characteristics of adversarial perturbation from the point cloud itself to improve the attack ability. 2) To simulate the natural attack under real applications, we need to focus on the visual quality of generated adversarial samples. We need to pay more attention to constrain the perturbation to preserve the visual and geometric properties of point clouds.

To solve the above problems, we propose a novel framework to attack the point cloud. First, feeding with label information, we design a generator to produce perturbation through training with a GAN structure to encourage the network to better learn the distribution of points in different classifications, which can effectively boost the attack ability of the attacker. Then, In order to further constrain the perturbation magnitude, we proposed a dynamic outlier removal step to remove excessive perturbations. Finally, adapted to our proposed framework, we design an untargeted attack loss that let the perturbation generator gain the ability to cheat point cloud models without a specified perturbation target. We have carried out a series of attack experiments on different point cloud models on the ModelNet dataset, and the final experimental results can verify the effectiveness of our method.

B. CONTRIBUTIONS

The main contributions of our method can be summarized as follows:

- We design a generator to produce point-to-point perturbations based on the original point cloud and its classification label. We apply the GAN structure to encourage distribution learning to boost the effectiveness of the attacker.
- In order to raise the generator's ability to cause attacks with only slight perturbation. We design a dynamic outlier removal step to constrain the number of perturbed points by canceling excessive perturbation.
- We have carried out a series of attack experiments and the final experimental results show the effectiveness of our proposed method.

The remainder of the paper is organized as follows. We introduce the related works in Section 2. Our proposed attack method is introduced in Section 3. In Section 4, we detail the dataset and related experiments and discuss the experimental results. Finally, the conclusions are presented in Section 5.

II. RELATED WORKS

A. DEEP 3D POINT CLOUD MODELS

Pointcloud is a collection of points sampled from the surface of a 3D object. It is an important data format for processing 3D models in computer vision tasks. Nowadays, point cloud has had a wide range of applications in various 3D model processing tasks such as classification, segmentation, retrieval, etc. PointNet [1] is the first deep learning algorithm framework that directly uses 3D point cloud, and it has good performance in several kinds of 3D model tasks. PointNet uses the symmetric function max-pooling to extract the features of point cloud data, but it will also make it lose local details. To solve this problem, inspired by CNN, PointNet++ [2] proposes a further improvement scheme, which makes the network extract local features and obtain better robustness. In order to make full use of the geometric or location information represented by point cloud data, DG-CNN [3] proposes a novel operation EdgeConv to better capture the local geometric features of point clouds while maintaining the invariance of point clouds. RS-CNN [4] achieves better use of the geometric properties of the point cloud. It proposes a novel learn-from-relationship revolution operator called relationship shape revolution, which is used to explicitly encode the geometric relationship of points so that the model can obtain better shape awareness and robustness.

B. POINT CLOUD ADVERSARIAL ATTACKS

Recently, some related work has brought 2D adversarial attack experience [14], [15], [16], [17], [18], [19] to the 3D field. To our knowledge, 3D-adv [10] is the first related work to study 3D point cloud adversarial attack, which proposes two attack methods: shift existing points, or add a small number of point clusters with meaningful shapes negligibly. [11] applies the fast iterative gradient method to 3D point cloud data to generate adversarial samples and takes into account the human perceptibility of perturbation on the 3D model. Reference [12] attacks the point cloud recognition model by

constructing a gradient-based salience map and dropping the points with the lowest salience scores. LG-GAN [13] proposed a generation-based adversarial attack method. Through the GAN structure training generator, targeted adversarial samples can be generated directly according to the input point cloud. AdvPC [20] leverages an encoder-decoder network to generate adversarial perturbation for each point, it could achieve a favorable transferability in attacking different victim networks. LP-GAN [21] first sample the key point from the original point cloud and then generate adversarial points with the sampled points. Meanwhile, they propose a perception loss to improve the quality of the adversarial samples from the original point cloud.

However, most of the existing methods mainly focused on targeted attacks, which means they need to specify the target label before generating the adversarial samples. Differing from them, in this paper, we hope to propose an untargeted attack method to generate point cloud adversarial samples in a more flexible way. We think this approach can better simulate natural attacks in real applications.

C. POINT CLOUD ADVERSARIAL DEFENSES

Inspired by the 2D defense, [22] proposed two methods: Gaussian noise and point quantization. They also proposed a simple random sampling (SRS) method for defense. DUP-NET [23] Proposed a Statistical Outlier Removal (SOR) method. By calculating the kNN distance of each point, the points that deviate from the surface will be removed, which will help reduce the dispersion of the surface. They also used a point cloud sampling network PU-NET [24], combined with the SOR method, and finally proposed DUP-NET as an end-to-end 3D adversarial defense framework.

III. OUR APPROACH

A. OVERALL FRAMEWORK

The overall framework of our proposed point cloud attack method is shown in Fig. 1.

Here, the original point cloud P and its classification label t are used as the input of G to generate the corresponding perturbation Δ. Applying the perturbation, and setting the perturbed point cloud as the input of module R, R is responsible for generating a mask vector for perturbation by calculating kNN distance to remove the excessive perturbation. The final adversarial example is $\hat{P} = P + R(\Delta)$. In the rest of this section, we will introduce the details of each part of our proposed method.

Formally, we define the problem as follows: We need to train a perturbation generator G to produce corresponding perturbation on the input point cloud P with minimal magnitude. In order to solve this problem, we design the structure of G, which also takes the real classification label of the original point cloud as an input. G consists of threshold parts: a label encoder E_l, a point cloud feature extractor E_p and a perturbation decoder D_p. E_l can convert label vector to a latent code z_l, and E_p encodes P into a latent feature F, concatenating z_l and F to fed D_p, then G output the perturbation Δ.

Adding the perturbation on the original point cloud and passing through the outlier removal module R, the final adversarial example is $\hat{P} = P + R(\Delta)$. Then, it will be sent to the discriminator D. We use an idea similar to ACGAN [25] to design the discriminator. Using the GAN structure can encourage the network to learn the geometric characteristics of the original point clouds. Through the confrontation between G and D, G can generate perturbation that conforms to the geometric characteristics of the original point cloud, which can boost the visual quality of final adversarial samples. Meanwhile, we utilized label information for training GAN, the purpose of D is not just to distinguish the disturbed point cloud from the original point cloud, it also trains an auxiliary classifier to judge whether the generated adversarial samples belong to the correct classification, it will greatly encourage the network to learn the distribution of point cloud with different classes, which can also help improve the attack ability of G to produce perturbation to fool the target classifier to other labels.

For discriminator D, the loss function contains a real-fake loss L_{adv} and a classification loss L_{cls}, which can be written as:

$$L_{adv} = \mathbb{E}\left[\log D(P)\right] + \mathbb{E}\left[\log(1 - D(\hat{P}))\right] \tag{1}$$

$$L_{cls} = \mathbb{E}\left[\log p(c = t \| P)\right] + \mathbb{E}\left[\log p(c = t \| \hat{P})\right] \tag{2}$$

Here, D is trained to maximize $L_{adv} + L_{cls}$, and the G is trained to maximize $L_{GAN} = L_{cls} - L_{adv}$.

B. DYNAMIC OUTLIER REMOVAL

The point cloud is the sampling of the real object surface, and thus has the property of a smooth surface. In the process of attacking the point cloud, if the points are disturbed too far, they will become outliers. On the one hand, the existence of these outliers will greatly affect the visual quality of the adversarial samples, on the other hand, they can be easily removed by the defense mechanism.

Adapted to our attack method, we propose a method to reduce the number of outliers. Inspired by [23], we calculate k-nearest neighbors (kNN) distance to remove the excessive perturbation. There are two advantages to this step: First, it can greatly improve the visual quality by constraining the number of perturbed points dynamically. Second, it can encourage the network to better learn the ability to cause attacks with slight perturbations.

For the adversarial sample $\hat{P} = \{p_1, p_2 \ldots, p_n\}$, for each point p_i, the average value of k points nearest to its euclidean distance is calculated as kNN distance:

$$d_i = \frac{1}{k} \sum_{\mathbf{p}_j \in knn(\mathbf{p}_i, k)} \left\| \mathbf{p}_i - \mathbf{p}_j \right\|_2, \quad i = 1, \ldots, n \tag{3}$$

Then the average and standard deviation of all these distances are calculated to set the threshold of perturbation

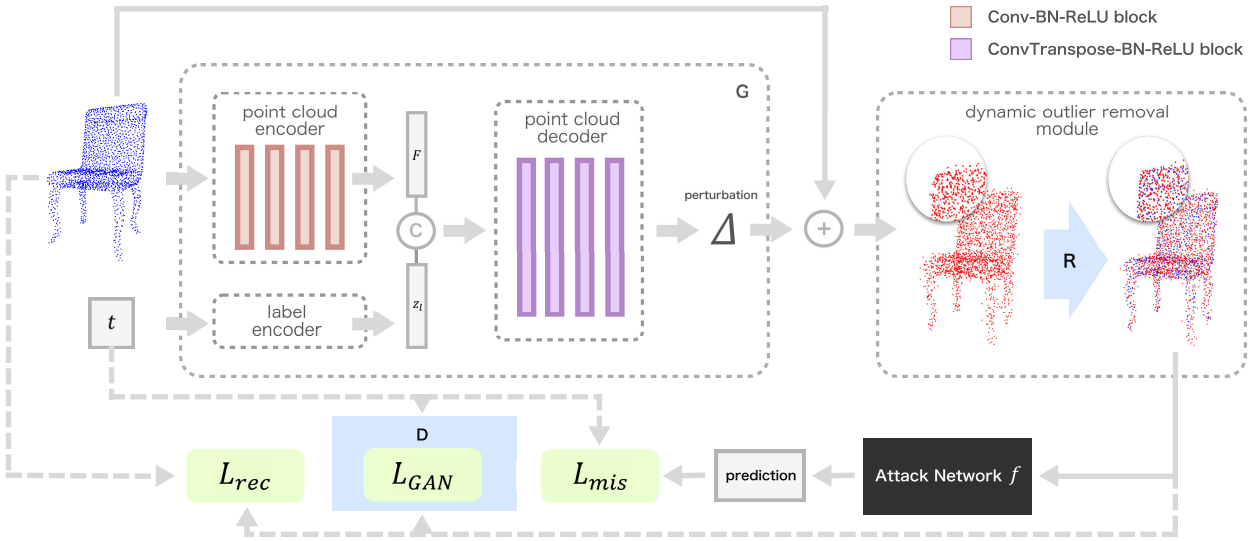

FIGURE 1. The illustration of our proposed attack framework. It mainly includes four parts: perturbation generator *G*, discriminator *D*, target attack network *f*, and dynamic outlier removal module *R*, and we use multiple loss function L_{rec}, L_{GAN}, L_{mis} to constrain the generated perturbation.

removal:

$$\bar{d} = \frac{1}{n}\sum_{i}^{n} d_i, \quad i = 1, \ldots, n \tag{4}$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i}^{n}\left(d_i - \bar{d}_i\right)^2} \tag{5}$$

On the basis of calculating KNN distance, we set a removal threshold $d_i < \bar{d} + \gamma \cdot \sigma$, the perturbation at the point where is greater than this threshold will be canceled. We cancel the perturbation by calculating a perturbation removal mask, the whole process of the outlier removal operation is shown in Algorithm.1.

C. OBJECT LOSS FUNCTION

Here, we use multiple loss to optimize the attack performance and visual quality of the generated adversarial samples, and we define the final object loss function as:

$$L = L_{GAN} + \alpha L_{rec} + \beta L_{atk} \tag{6}$$

The object loss function consists of three parts: the L_{GAN} is adversarial loss, which has been introduced in the above section. L_{rec} is a point cloud reconstruction loss based on L_2 norm which is used to limit the magnitude of the perturbation. L_{mis} is misclassification loss, which is used to enable the generator to be able to fool the target classifier. α and β are two hyperparameters to control the weight of different losses.

1) RECONSTRUCTION LOSS

To ensure that the generated adversarial samples are visually similar to the original point cloud, we need to limit the magnitude of the perturbation. In the reconstruction works of the point cloud, commonly used distance metric includes L_2, chamfer, EMD, and Hausdorff Distance. In our proposed framework, we adopt L_2 distance. Because compared with some point cloud generation or upsampling tasks, the task of attack method needs to make the adversarial samples to gain the ability to cheat the classifier, it's more difficult for the network to preserve the visual quality of generation. L_2 norm can represent the energy of the whole perturbation, so it can provide a hard constraint on the magnitude of the perturbation. Therefore, we finally choose the L_2 norm to serve as the reconstruction loss in our attack method.

Algorithm 1 Process of Outlier Removal Operation

Input: original points P, adversarial perturbation Δ
Output: adversarial points $\hat{P}$

```
Set C = P + Δ
Set as reference point cloud C = c_1, c_2, ..., c_n
for i = 1 to n do
    Identify k nearest point with c_i
    Calculate kNN distance d_i using Eq.3
end for
Calculate the mean d̄ using Eq.4
Calculate the standard deviation σ using Eq.5
Initial removal mask M = {1, 1 ..., 1}
for j = 1 to n do
    If d_i < d̄ + γ · σ then
        Set M[i] = 0
end for
The final adversarial points P̂ = P + Δ × M
return P̂
```

2) MISCLASSIFICATION LOSS

In our method, we train the generator with the label to encourage the network to learn the distribution of different classifications, which also can help the attacker to misclassify the inputs as any other class. Therefore, we choose to adopt the way of untargeted attacks to make full use of our designed framework. Inspired by [26] and [27], here, we use the opposite of the standard cross-entropy loss:

$$L_{mis} = -\frac{1}{\left[t \log f(\hat{P}) + (1-t)\log(1-f(\hat{P}))\right]} \tag{7}$$

where $\hat{P}$ is the generated adversarial samples, t is its original classification label. f represents the victim network, and $f(\hat{P})$ is its predicted probability of the adversarial point cloud.

IV. EXPERIMENT

In this section, we will introduce and discuss a series of experiments we have done. Firstly, We will introduce our experimental setup. Then we compare our method with some state-of-the-art methods in attack ability, performance against defense mechanisms, and transferability on different victim networks. The final experiment results can prove the superiority of our method. Finally, we conducted ablation studies to illustrate the effectiveness of each module in our method. The attack success rate is used as the evaluation metric of the experimental results.

A. EXPERIMENT SETUP

1) DATASET

To demonstrate the effectiveness of our proposed method, we carry out a series of attack experiments on the popular dataset ModelNet. The dataset contains 12,311 3D CAD with 40 object categories in the real world. ModelNet10 is a subset of ModelNet, it only has 10 popular classes but is cleaner than ModelNet40. We both used the above two datasets with the official split for adversarial training. We chose 2,048 $\times$ 3 point cloud as the input of all carried experiments.

2) VICTIM NETWORK

We chose some famous point cloud models as our target network, including [1], [2], [3], and [4]. We use the default settings in these works to train the classifier. When training our attack method, we freeze the parameters of the victim network.

B. COMPARISON ATTACK METHOD

In order to better demonstrate the effectiveness of our proposed method, we compare the performance with some existing state-of-the-art untargeted attack methods:

- **3d adv** [10]. They proposed two attack methods: shift existing points or add a small number of point clusters with meaningful shapes negligibly.
- **IFGM** [11]. They apply the fast/iterative gradient method to 3D point cloud data to generate adversarial samples.
- **AdvPC** [20]. They apply an encoder-decoder network to generate adversarial perturbation for each point. To obtain the favorable visual quality of generated adversarial samples, they manually set the threshold to limit the magnitude of the perturbations.

1) DEFENSE MECHANISM

We also analyze the performance of our attack and comparison method against several defenses mechanism. We experimented with the following defense mechanisms:

- **Simple Random Sampling (SRS)**. Using the statistical method, through the equal probability random sampling point cloud achieves the effect of defense attack
- **Statistical Outlier Removal (SOR)**. By calculating the KNN distance, remove the discrete points on the surface.
- **DUP-NET** [23]. It achieves defense by combining SOR and PU-NET to upsampling the point cloud after removing the surface discrete points.

2) IMPLEMENTATION DETAILS

We implement our proposed framework based on Pytorch. For the optimization, we train the network end-to-end using the Adam [28] optimizer with a batch size of 16, and the initial learning rate is set at 0.0001. The training process is carried out on a server with Intel i5 9600k CPU, an NVIDIA RTX 3090 graphic card, and 32GB RAM, it takes about 4h to train 100 epochs and complete the network training process. Finally, we select the trained models with the lowest validation loss for visualization and calculate the metrics for quantitative analysis.

C. ATTACKING RESULTS

1) COMPARING WITH METHODS

Results compared with prior methods are summarized in Table. 1, and several generated adversarial samples generated by our trained model are shown in Fig. 3. For the method we proposed in this paper, we set hyperparameters $\alpha = 1, \beta = 2, \gamma = 1$. As can be seen from the results table shown below, our method has better attack success rate performance when attacking the network without defense mechanisms compared with the baseline. When attacking defense models, benefit from the excellent visual quality of the generated adversarial samples, our method can achieve the best attack success rate which is much higher than the baseline.

2) ATTACKING DIFFERENT VICTIM NETWORK

We trained our attack framework with different point cloud models as targets to evaluate the attack performance of our method against different victim networks with defense mechanisms. The corresponding experimental result is shown in Table. 2.

3) TRANSFERABILITY

In recent years, transferability is a very important performance index in the field of adversarial attacks. Transferability

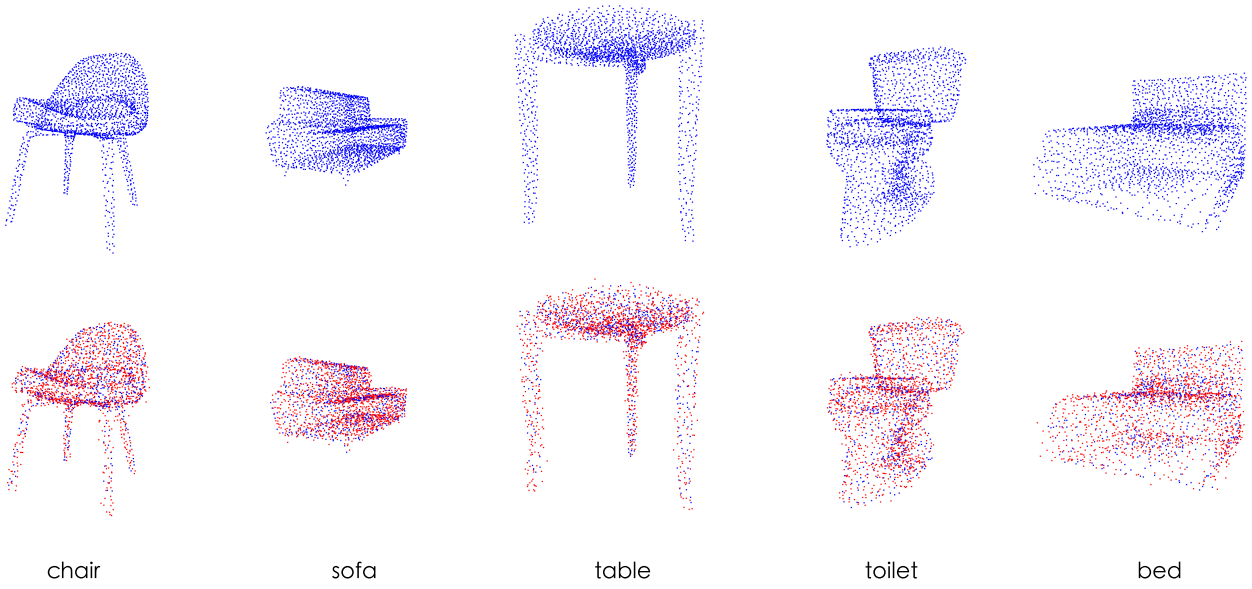

FIGURE 2. Visualization of adversarial samples attacking PointNet of different classes. The point clouds shown in the first row are unperturbed points, and the second row lists some generated adversarial samples. The red points represent adversarial points that are perturbed.

TABLE 1. Performance comparison with state-of-the-art untargeted point cloud attack methods.

Method	No def	SRS	SOR	DUP-NET [23]	L2 Distance(meter)	Time(second)
3D-adv + L_2 [10]	**100**%	0%	0%	0%	0.01	40.80
3D-adv + chamfer [10]	**100**%	0%	0%	0%	-	43.74
3D-adv + 3 clusters [10]	94.7%	2.7%	2.2%	2.0%	-	51.35
IFGM [19]	73.0%	14.5%	8.5%	3.3%	0.31	0.275
AdvPC [20]	94.8%	80.0%	36.8%	34.6%	-	-
OURS	91.7%	**81.4**%	**71.7**%	**61.5**%	**0.26**	**0.040**

TABLE 2. Attacking against defense mechanism.

	ModelNet10				ModelNet40			
	PointNet [1]	PointNet++ [2]	DG-CNN [3]	RS-CNN [4]	PointNet [1]	PointNet++ [2]	DG-CNN [3]	RS-CNN [4]
No Defense	95.65%	93.29%	90.81%	90.63%	91.71%	88.82%	85.16%	84.55%
SRS	85.71%	82.92%	80.41%	79.37%	81.74%	80.47%	77.24%	75.12%
SOR	75.44%	72.85%	70.15%	68.28%	71.69%	67.32%	65.78%	63.83%
DUP-NET [23]	65.89%	63.46%	61.85%	58.84%	61.53%	58.76%	56.89%	55.35%

evaluates the ability of the adversarial samples generated by training on a specific victim network to mislead other victim networks. For our proposed method, we make full use of the label information and train a GAN structure to let the model better learn the distribution of the original point cloud with different classes, thus improving the transferability of our attack method. Using the attacker trained on PointNet [1], PointNet++ [1], DGCNN [3], and RSCNN [4], we compare their transferability to each other. the experimental results are shown in Table. 3.

D. ABLATION STUDY

1) COMPARING DIFFERENT DISCRIMINATORS

In our attack framework, the GAN structure is the key for the perturbation generator to learn data distribution, which plays a very important role in improving the attack effect, especially in improving the quality of adversarial samples. As we all know, the choice of GAN structure is very important, which will greatly affect the training effect and difficulty. Here, we refer to several classic GAN structure designs, including GAN [29], PatchGAN [30], cGAN, and ACGAN [25]. According to the ideas of training GAN in these works, we implemented different discriminators to carry out experiments. The corresponding attack success rate results under different structure selections are shown in Table. 4.

From the experimental results, we come to the conclusion that the correct choice of discriminator design can effectively improve the success rate of attack and the quality of adversarial samples. comparing the four methods, we draw the following conclusions:

TABLE 3. Transferability of trained models.

	PointNet	PointNet++(SSG)	DGCNN	RSCNN
PointNet	\	17.18%	15.13%	13.21%
PointNet++(SSG)	17.98%	\	17.48%	18.39%
DGCNN	18.73%	18.89%	\	19.13%
RSCNN	16.46%	18.37%	18.47%	\

TABLE 4. Attack success rate using different GAN structure.

	GAN	PatchGAN	cGAN	ACGAN
Success Rate	84.45%	87.94%	88.12%	91.71%

TABLE 5. The success rate of the attacker with different hyperparameters.

Success Rate	α=0.1	α=0.5	α=1	α=2
β=1	91.34%	90.13%	89.19%	87.61%
β=2	93.16%	92.47%	91.71%	88.38%
β=3	94.57%	91.17%	90.12%	87.13%

- The effect of the original GAN is the worst, because, in the original GAN, the discriminator only focuses on distinguishing the disturbed point cloud from the original point cloud, and does not effectively use the local information and label information of the point cloud. This makes the generator unable to learn deeper data distribution and results in poor performance.
- We use the existing point cloud Graph PatchGAN [31] to carry on our experiment. Compared with the original GAN, PatchGAN uses convolutional operations to capture the local feature. This idea can bring some benefit to the quality of adversarial samples, but the attack performance is not very good due to the lack of label information.
- In cGAN, in addition to the point cloud, the discriminator also uses the label as an input. This can make GAN better learn the distribution of different classes of data, and the attack ability of the model is also improved.
- ACGAN is the final idea we use in the proposed method. It uses the auxiliary classifier to make the discriminator have the function of discriminating classification. Compared with cGAN, ACGAN is superior in attack ability and generation quality.

2) HYPER PARAMETER STUDY

We used two hyperparameters α and β to balance attack ability and the similarity between the adversarial and the original point cloud. We employ PointNet as the victim network to carry out experiments to figure out the appropriate value choice for these two parameters. The relevant experimental results are shown in Table. 5. From the experiment, we noticed that although the highest attack success rate could reach 94.57% under the experimental setting of $\alpha = 0.1$ and $\beta = 3$, the visual quality of the generated samples was relatively discrete, and the L2 distance could only achieve 0.35. To obtain the best balance between the visual quality and attack performance, we finally set $\alpha = 1$ and $\beta = 2$ to get the best attack performance with an attack success rate of 91.7% and L2 distance of 0.26.

V. CONCLUSION

With the rapid development of the 3D model analysis field, the point cloud has become an important 3D object data format widely used in many applications. The increasing demand for the safety and efficiency of 3D applications leads to higher demand for the robustness of point cloud models. In this work, we propose a novel framework to attack point cloud models. Fed with label information, we train a perturbation generator with the GAN structure to make the network fully learn the representation of point clouds under different classes. Meanwhile, we design a dynamic outlier removal method to remove the excessive perturbation. Our method can cause attacks with high performance and thus can be utilized to improve the robustness of 3D models. We carried out extensive experiments, and the results show the effectiveness of our method.

REFERENCES

[1] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.

[2] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst.*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds. Long Beach, CA, USA, 2017, pp. 5099–5108.

[3] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, p. 146, 2019.

[4] Y. Liu, B. Fan, S. Xiang, and C. Pan, "Relation-shape convolutional neural network for point cloud analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8895–8904.

[5] L. Xie, C. Xiang, Z. Yu, G. Xu, Z. Yang, D. Cai, and X. He, "PI-RCNN: An efficient multi-sensor 3D object detector with point-based attentive Cont-Conv fusion module," in *Proc. EAAI*, New York, NY, USA, Feb. 2020, pp. 12460–12467.

[6] B. Hu, Y. Feng, J. Sun, Y. Gao, and J. Tan, "Driving preference analysis and electricity pricing strategy comparison for electric vehicles in smart city," *Inf. Sci.*, vol. 504, pp. 202–220, Dec. 2019.

[7] Y. Li, Y. Liu, J. Zhu, S. Ma, Z. Niu, and R. Guo, "Spatiotemporal road scene reconstruction using superpixel-based Markov random field," *Inf. Sci.*, vol. 507, pp. 124–142, Jan. 2020.

[8] A. Pumarola, S. Popov, F. Moreno-Noguer, and V. Ferrari, "C-flow: Conditional generative flow models for images and 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7949–7958.

[9] C. He, H. Zeng, J. Huang, X.-S. Hua, and L. Zhang, "Structure aware single-stage 3D object detection from point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11870–11879.

[10] C. Xiang, C. R. Qi, and B. Li, "Generating 3D adversarial point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9136–9144.
[11] D. Liu, R. Yu, and H. Su, "Extending adversarial attacks and defenses to deep 3D point cloud classifiers," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 2279–2283.
[12] T. Zheng, C. Chen, J. Yuan, B. Li, and K. Ren, "PointCloud saliency maps," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1598–1606.
[13] H. Zhou, D. Chen, J. Liao, K. Chen, X. Dong, K. Liu, W. Zhang, G. Hua, and N. Yu, "LG-GAN: Label guided adversarial network for flexible targeted attack of point cloud based deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10353–10362.
[14] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2574–2582.
[15] M. Naseer, S. H. Khan, M. H. Khan, F. S. Khan, and F. Porikli, "Cross-domain transferability of adversarial perturbations," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst., (NeurIPS)*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds. Vancouver, BC, Canada, 2019, pp. 12885–12895.
[16] O. Poursaeed, I. Katsman, B. Gao, and S. Belongie, "Generative adversarial perturbations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4422–4431.
[17] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 39–57.
[18] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. 3rd Int. Conf. Learn. Represent., (ICLR)*, Y. Bengio and Y. LeCun, Eds. San Diego, CA, USA, 2015, pp. 1–11.
[19] S. Gu and L. Rigazio, "Towards deep neural network architectures robust to adversarial examples," in *Proc. 3rd Int. Conf. Learn. Represent., (ICLR)*, Y. Bengio and Y. LeCun, Eds. San Diego, CA, USA, May 2015, pp. 1–9.
[20] A. Hamdi, S. Rojas, A. Thabet, and B. Ghanem, "AdvPC: Transferable adversarial perturbations on 3D point clouds," in *Proc. Eur. Conf. Comput. Vis.*, in Lecture Notes in Computer Science, vol. 12357. Cham, Switzerland: Springer, 2020, pp. 241–257.
[21] Q. Liang, Q. Li, and S. Yang, "LP-GAN: Learning perturbations based on generative adversarial networks for point cloud adversarial attacks," *Image Vis. Comput.*, vol. 120, Apr. 2022, Art. no. 104370.
[22] J. Yang, Q. Zhang, R. Fang, B. Ni, J. Liu, and Q. Tian, "Adversarial attack and defense on point sets," 2019, *arXiv:1902.10899*.
[23] H. Zhou, K. Chen, W. Zhang, H. Fang, W. Zhou, and N. Yu, "DUP-Net: Denoiser and upsampler network for 3D adversarial point clouds defense," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1961–1970.
[24] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "PU-Net: Point cloud upsampling network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2790–2799.
[25] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. 34th Int. Conf. Mach. Learn. (PMLR)*, vol. 70, 2017, pp. 2642–2651.
[26] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.
[27] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song, "Generating adversarial examples with adversarial networks," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3905–3911.
[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[29] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst.*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Montreal, QC, Canada, 2014, pp. 2672–2680.
[30] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5967–5976.
[31] H. Wu, J. Zhang, and K. Huang, "Point cloud super resolution with adversarial residual graph networks," 2019, *arXiv:1908.02111*.

FENGMEI HE is currently working with the Department of Automation and Electrical Engineering, Tianjin University of Technology and Education, and the College of Information Technology, Wenzhou Vocational College of Science and Technology and Education. Her research interests include machine learning and computer vision.

YIHUAI CHEN is currently working with the Department of Automation and Electrical Engineering, Tianjin University of Technology and Education, and the City University of Wenzhou. His current research interests include computer vision and machine learning.

RUIDONG CHEN is currently pursuing the master's degree with the School of Electrical and Information Engineering, Tianjin University, Tianjin, China. His current research interests include 3D model recognition and 3D model generation tasks.

WEIZHI NIE (Member, IEEE) received the Ph.D. degree in electronics engineering from Tianjin University, China. He was a Visiting Scholar at the National University of Singapore. He is currently an Associate Professor with the School of Electrical and Information Engineering, Tianjin University. His research interests include computer vision and machine learning.