Curvature-Invariant Adversarial Attacks for 3D Point Clouds

Jianping Zhang[1], Wenwei Gu[1], Yizhan Huang[1], Zhihan Jiang[1], Weibin Wu[2*], Michael R. Lyu[1]

[1]Department of Computer Science and Engineering, The Chinese University of Hong Kong
[2]School of Software Engineering, Sun Yat-sen University
{jpzhang, wwgu21, yzhuang22, zhjiang22, lyu}@cse.cuhk.edu.hk, wuwb36@mail.sysu.edu.cn

Abstract

Imperceptibility is one of the crucial requirements for adversarial examples. Previous adversarial attacks on 3D point cloud recognition suffer from noticeable outliers, resulting in low imperceptibility. We think that the drawbacks can be alleviated via taking the local curvature of the point cloud into consideration. Existing approaches introduce the local geometry distance into the attack objective function. However, their definition of the local geometry distance neglects different perceptibility of distortions along different directions. In this paper, we aim to enhance the imperceptibility of adversarial attacks on 3D point cloud recognition by better preserving the local curvature of the original 3D point clouds. To this end, we propose the Curvature-Invariant Method (CIM), which directly regularizes the back-propagated gradient during the generation of adversarial point clouds based on two assumptions. Specifically, we first decompose the back-propagated gradients into the tangent plane and the normal direction. Then we directly reduce the gradient along the large curvature direction on the tangent plane and only keep the gradient along the negative normal direction. Comprehensive experimental comparisons confirm the superiority of our approach. Notably, our strategy can achieve 7.2% and 14.5% improvements in Hausdorff distance and Gaussian curvature measurements of the imperceptibility.

Introduction

Deep neural networks (DNNs) dominate state-of-the-art solutions for a variety of computer vision tasks, comprised of image classification (Russakovsky et al. 2015; Wu et al. 2019), object detection (Lin et al. 2014) and 3D point cloud recognition (Yi et al. 2016). 3D point cloud recognition models are widely deployed in lots of safety-critical real-world systems, such as autonomous driving and medical diagnosis systems (Dong, Wang, and Abbas 2021). However, recent research shows that DNNs are vulnerable to adversarial attacks (Zhang et al. 2023b,a, 2022), which inject imperceptible noise into the original input to mislead the DNN models. It raises security issues on the deployment of DNN applications (Zhang et al. 2023c,d; Wu et al. 2023). Similarly, 3D point cloud recognition models are also susceptible to adversarial attacks (Xiang, Qi, and Li 2019). Therefore, it is indispensable to design adversarial attack approaches to detect the deficiencies inside the 3D point cloud recognition models before their deployment in real-world applications.

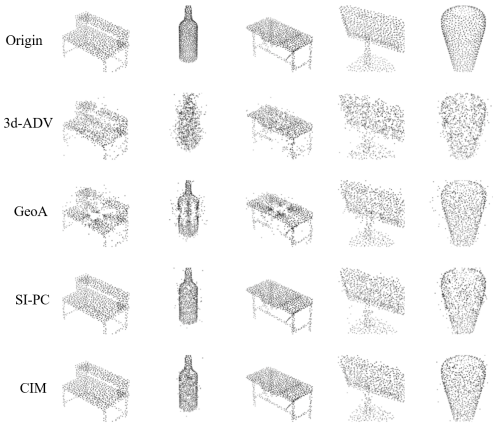

Figure 1: Visualization of generated adversarial point cloud by various attacking algorithms on randomly selected five classes. The adversarial examples are generated on the PointNet model.

3D point cloud adversarial attacks work by adding, deleting, or shifting points in the point clouds (Xiang, Qi, and Li 2019). Among the schemes of 3D point cloud adversarial attacks, shifting points, i.e., changing the coordinates of the points, attracts more attention from researchers (Hamdi et al. 2020). Similar to 2D adversarial images (Wu et al. 2021, 2020b), 3D adversarial point clouds are also usually crafted with the guidance of the back-propagated gradients. For example, attackers can adapt the Fast Gradient Sign Method (FGSM) to generate 3D adversarial point clouds (Goodfellow, Shlens, and Szegedy 2014).

Similar to adversarial attacks on 2D images, one of the core challenges of 3D point cloud adversarial attacks is imperceptibility, which requires that the modification of the point cloud is unnoticeable for humans. Inspired from the L_p constraint of 2D image adversarial attacks that the p norm of the perturbation is constrained by a budget of ϵ (Deng et al. 2023; Wu et al. 2020a), 3D point cloud adversarial attacks also adopt the similar idea. That is, the modification of the point's coordinates should satisfy the L_p constraints.

*Corresponding author.

However, satisfying this hard constraint is not enough, since the resultant adversarial point clouds can still contain noisy outliers, which destroy the local geometry properties of the original point clouds and deteriorate the visual quality of the adversarial point clouds in Figure 1.

To improve the imperceptibility of adversarial point clouds, some researchers combine the misclassification objective function with the Mean Square Error (MSE) to make small L_2 perturbations on the target point clouds. Some (Liu and Hu 2022; Wen et al. 2020) improve the MSE loss with other advanced distance measurements, like Chamfer distance (Fan, Su, and Guibas 2017) and Hausdorff distance (Taha and Hanbury 2015). Others adopt local shape descriptors, like normal direction, to reduce noisy outliers in 3D adversarial point cloud (Wen et al. 2020).

Nevertheless, the imperceptibility of existing 3D point cloud adversarial attacks is still unsatisfactory. The reasons are as follows: (1) Though previous approaches take 3D distance measurements and local shape descriptors into consideration, they neglect the influence of the local curvature, which is also a vital surface property along with the normal direction. (2) Some methods (Wen et al. 2020) try to introduce the local curvature distance into the attack objective function. However, their definition of the local curvature distance focuses on the average angle between the normal vector and the vectors starting from a point to each of its neighboring points, which neglects different perceptibility of distortions along different directions. Besides, such a combination of the misclassification objective function and the distance measurements requires extra hyper-parameters to balance the power of different terms during the generation of adversarial point clouds, which are time-consuming to tune.

In this paper, we propose the Curvature-Invariant Method (CIM) to overcome the above flaws of previous approaches. To improve imperceptibility, CIM attempts to utilize the information of local curvature to preserve the local surface geometry of 3D point clouds. Instead of incorporating complicated objectives into the attack objective function, we directly rectify the directions of the back-propagated gradients during the search for adversarial point clouds.

Specifically, we first decompose the back-propagated gradient of each point in the point cloud into three orthogonal directions: the normal direction, the maximum principal direction, and the minimum principal direction. The maximum and minimum principal directions reside on the tangent plane of the point. Then as shown in Figure 3, to maintain the local geometry of point clouds, we directly modify the directions of the update gradients as follows: (1) On the tangent plane, we reduce the gradient along the large curvature direction. To trade off imperceptibility and attack effectiveness, we achieve this goal by only keeping the gradient along a linear combination of two principal directions with more weights on the minimum principal direction. (2) Along the normal direction, we only keep the gradient along the negative normal direction, having a negative dot product with the normal.

We conduct extensive experiments to validate the effectiveness of our proposed CIM. Remarkably, on average, our CIM can not only enhance the Hausdorff distance by over 7.2 % , but also boost the adversarial imperceptibility measured by the Gaussian curvature difference by above 14.5 %. Our contributions are:

- To improve the imperceptibility of the generated 3D adversarial point clouds, we propose the Curvature-Invariant Method (CIM). CIM attempts to utilize the information of local curvature to preserve the local surface geometry of 3D point clouds. To this end, we directly regularize the update gradient by reducing the gradient along the large curvature direction and only keeping the gradient along the negative normal direction.
- We derive the mathematical proofs of two assumptions and the upper bound of the gradient variation for each point generated by our CIM.
- We conduct comprehensive experiments to validate the advantages of CIM, which promotes both the attack success rate and the imperceptibility of 3D adversarial point clouds.

Related Work

3D Point Cloud Recognition

A 3D point cloud is a discrete set of data points to represent the 3D shape of an object. With the development of deep learning, various deep learning-based approaches have achieved surprising performance on 3D point cloud recognition. PointNet (Qi et al. 2017a) is a representative approach applying a multi-layer perception to point features and deploying a max-pool module for aggregating point features efficiently. PointNet++ (Qi et al. 2017b) extends PointNet with single-scale and multi-scale designs for better extracting local features. DGCNN (Wang et al. 2019) utilizes point neighbors to better extract local geometric features. PointConv (Wu, Qi, and Fuxin 2019) reformulates the convolution operation to efficiently compute the weight functions for scaling up the network.

3D Adversarial Attacks and Defenses

Current 3D adversarial attacks can be roughly divided into three categories based on the perturbation schemes: adding points, deleting points, and shifting points. Some researchers utilize the saliency map for deleting important points (Zheng et al. 2019) or add synthetic points to the original point cloud (Xiang, Qi, and Li 2019). More research attention is focused on perturbing the coordinates of the point in the point cloud (Tu et al. 2020; Zhou et al. 2020). Usually, the adversarial point clouds are crafted by employing the gradient of a C&W attack objective function that combines the misclassification loss with different quality measurements, including 3D distance metrics (Liu and Hu 2022) and local shape descriptors (Wen et al. 2020).

Our Curvature-Invariant Method (CIM) proposes to consider the local surface property (i.e., the curvature) of the point in the point cloud. Previous approaches try to introduce the local curvature distance into the attack objective function. However, their definition of the local curvature distance neglects different perceptibility of distortions along different

directions. Besides, such a combination of the misclassification objective function and the distance measurements requires extra hyper-parameters to balance their power, which are time-consuming to tune. Unlike previous methods, we directly regularize the update gradient on each point based on the curvature property to maintain the local geometry during the search for adversarial point clouds.

Multiple defense approaches have been proposed to defend against 3D adversarial attacks. Mainstream schemes are based on pre-processing (Zhou et al. 2019), adversarial training (Liu, Yu, and Su 2019; Sun et al. 2021), and gather-vectors (Dong et al. 2020).

Methodology

Preliminary

A point cloud is composed of an unordered set of points $\boldsymbol{P} = \{\boldsymbol{p_i}\}_{i=1}^{n} \in \mathbb{R}^{n\times 3}$ sampled from the surface of an object with the ground truth label c. Each point $\boldsymbol{p_i} \in \mathbb{R}^3$ is a vector representing the coordinates (x, y, z) of the point i. n is the number of points in the point cloud. A classifier $f(\cdot)$ takes the point cloud $\boldsymbol{P}$ as the input and outputs the label prediction $c' = f(\boldsymbol{P})$. In the setting of 3D adversarial point cloud attacks, we aim to craft an adversarial point cloud $\boldsymbol{P}^{adv}$ by shifting the original point cloud with $\boldsymbol{\Delta} \in \mathbb{R}^{n\times 3}$ (i.e., $\boldsymbol{P}^{adv} = \boldsymbol{P} + \boldsymbol{\Delta}$) to mislead the classifier (i.e., $f(\boldsymbol{P}^{adv}) \neq c$). With the aim of imperceptibility, the perturbation $\boldsymbol{\Delta}$ should satisfy the L_p constraint such that $\|\boldsymbol{\Delta}\|_p < \epsilon$, where $\|\cdot\|_p$ is the L_p norm. In this paper, we focus on the L_∞ norm by following the baseline (Huang et al. 2022).

Curvature-Invariant Method

The motivation behind our Curvature-Invariant Method (CIM) is to improve the adversarial imperceptibility by taking the local curvature into consideration. Based on our two assumptions, we directly regularize the update gradient for each point in the point cloud during the generation of adversarial point clouds. To this end, we first transform the original axis to a proper axis for each point in the point cloud for efficiently regularizing the update gradient. Then we directly regularize the update gradient under the transformed axis based on our two assumptions to generate adversarial point clouds.

Coordinate Transformation For a given point cloud $\boldsymbol{P} = \{\boldsymbol{p_i}\}_{i=1}^{n} \in \mathbb{R}^{n\times 3}$, the local surface property of a point $\boldsymbol{p_i}$ can be approximated by its k nearest neighbors $\mathcal{N}_{\boldsymbol{p_i}}$ on the point cloud (Hoppe et al. 1992). Specifically, we first compute the covariance matrix $\mathcal{C}_{\boldsymbol{p_i}}$ of the differences between $\boldsymbol{p_i}$ and each of its k nearest neighbors $\mathcal{N}_{\boldsymbol{p_i}}$ as shown in Equation 1:

$$\mathcal{C}_{\boldsymbol{p_i}} = \sum_{\boldsymbol{q}\in\mathcal{N}_{\boldsymbol{p_i}}} (\boldsymbol{q} - \boldsymbol{p_i})(\boldsymbol{q} - \boldsymbol{p_i})^T. \tag{1}$$

The covariance matrix $\mathcal{C}_{\boldsymbol{p_i}}$ is positive semi-definite. We then obtain its three eigenvalues $(\lambda_1, \lambda_2, \lambda_3)$ in descending order and the corresponding eigenvectors $(\boldsymbol{e_1}, \boldsymbol{e_2}, \boldsymbol{e_3})$.

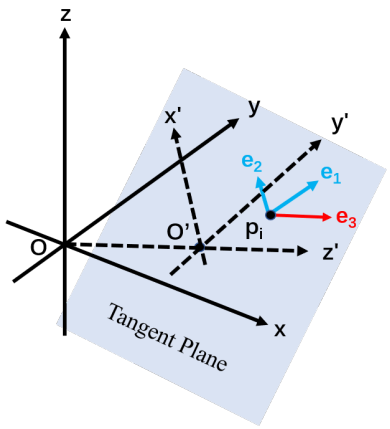

Figure 2: The illustration of coordinate transformation. We project the origin of the original coordinate system to the tangent plane and set the projection point as the origin of the transformed coordinate system. The x' and y' are two directions on the tangent plane determined by two parameters a and b, while z' is the normal direction.

Following the basic concept of differential geometry (Do Carmo 2016), the first two eigenvalues (λ_1 and λ_2) are the principal curvatures of the local surface of $\boldsymbol{p_i}$ determined by its neighbors. Specifically, λ_1 is the maximum principal curvature, and the corresponding maximum principal direction is $\boldsymbol{e_1}$. Besides, λ_2 is the minimum principal curvature, and the corresponding minimum principal direction is $\boldsymbol{e_2}$. The two principal directions define the tangent plane of the point $\boldsymbol{p_i}$. Furthermore, the last eigenvector $\boldsymbol{e_3}$ is the normal vector of the tangent plane, and we denote it to be the normal direction.

To conveniently regularize the update gradient based on the local geometry, we introduce a new coordinate system, which sets the normal direction $\boldsymbol{e_3}$ to be its $\boldsymbol{z'}$ direction. Since the normal direction is perpendicular to the tangent plane, any pair of orthogonal vectors residing on the tangent plane can form the $\boldsymbol{x'}$ and $\boldsymbol{y'}$ directions of the new coordinate system, respectively. We note that the two principal directions ($\boldsymbol{e_1}$ and $\boldsymbol{e_2}$) form one basis of the tangent plane. Therefore, we can represent new $\boldsymbol{x'}$ and $\boldsymbol{y'}$ axes with the linear combination of the two principal directions.

Theorem 1. *$\boldsymbol{x'} = a \cdot \boldsymbol{e_1} + b \cdot \boldsymbol{e_2}$ and $\boldsymbol{y'} = b \cdot \boldsymbol{e_1} - a \cdot \boldsymbol{e_2}$ such that $a, b \in \mathbb{R}$ and $a^2 + b^2 = 1$ form one basis of the tangent plane.*

After the determination of the three axis directions of the new coordinate system, we compute the new origin $\boldsymbol{O'}$ of the transformed coordinate system. We take the projection of the origin $\boldsymbol{O}$ from the original coordinate onto the tangent plane as shown in Figure 2 and assign the projected origin as the new origin.

We can now formulate the transformation from the original coordinate system $\boldsymbol{O - xyz}$ to the new coordinate system $\boldsymbol{O' - x'y'z'}$. As we can observe from Figure 2, the coordinate transformation consists of the translation from $\boldsymbol{O}$ to $\boldsymbol{O'}$ and the rotation of the axes. Therefore, we utilize a

rotation matrix and a translation matrix to define the coordinate transformation.

Theorem 2. *Denote $\boldsymbol{S_i} : \mathbb{R}^3 \mapsto \mathbb{R}^3$ is the transformation to convert the original coordinate system $\boldsymbol{O - xyz}$ to the new coordinate system $\boldsymbol{O' - x'y'z'}$. The transformation $\boldsymbol{S_i}$ consists of a rotation matrix $\boldsymbol{R_i}$ and a translation matrix $\boldsymbol{T_i}$. The coordinate $\boldsymbol{p'_i}$ under the new coordinate system and the coordinate $\boldsymbol{p_i}$ under the original coordinate system can be transformed conversely by the following equation:*

$$\begin{aligned} \boldsymbol{p'_i} &= \boldsymbol{R_i}(\boldsymbol{p_i} + \boldsymbol{T_i}), \\ \boldsymbol{p_i} &= \boldsymbol{R_i}^T \boldsymbol{p'_i} - \boldsymbol{T_i}. \end{aligned} \tag{2}$$

The rotation matrix $\boldsymbol{R_i}$ and the translation matrix $\boldsymbol{T_i}$ are given as follows:

$$\begin{aligned} \boldsymbol{R_i} &= \begin{pmatrix} a & b & 0 \\ b & -a & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{e_1} \\ \boldsymbol{e_2} \\ \boldsymbol{e_3} \end{pmatrix}, \\ \boldsymbol{T_i} &= -(\boldsymbol{p_i}^T \boldsymbol{e_3})\boldsymbol{e_3}. \end{aligned} \tag{3}$$

Attacking Algorithm For a given classifier f and the input point cloud $\boldsymbol{P}$, we first apply the coordinate transformation to all the points in the point cloud:

$$\begin{aligned} \boldsymbol{P'} &= \{\boldsymbol{R_i}(\boldsymbol{p_i} + \boldsymbol{T_i})\}_{i=1}^N, \\ \boldsymbol{P} &= \{\boldsymbol{R_i}^T \boldsymbol{p'_i} - \boldsymbol{T_i}\}_{i=1}^N. \end{aligned} \tag{4}$$

We then compute the max-margin logit loss function by following C&W (Carlini and Wagner 2017) as the attack objective function:

$$L(\boldsymbol{P}, c) = \max([f(\boldsymbol{P})]_c - \max_{i \neq c}[f(\boldsymbol{P})]_i, 0), \tag{5}$$

where c is the ground truth label of the input point cloud, and $[f(\boldsymbol{P})]_i$ is the model's confidence score of classifying the input point cloud $\boldsymbol{P}$ into the class i.

With the objective to obtain the gradient on the transformed coordinate for efficiently regularizing the gradient, we treat the input point cloud $\boldsymbol{P}$ as a function of the point cloud $\boldsymbol{P'}$ in the tangent-normal space with the rotation and translation matrices. The coordinate transformation is differentiable, so we can directly take the gradient of the loss function with respect to the transformed point cloud. We denote the gradient of the transformed point cloud as $\boldsymbol{G}$, where $\boldsymbol{g_i} = (g_{i1}, g_{i2}, g_{i3})$ is the gradient of the attack objective function with respect to the transformed point $\boldsymbol{p'_i}$:

$$\boldsymbol{G} = \{\boldsymbol{g_i}\}_{i=1}^N = \nabla_{\boldsymbol{P'}} L(\boldsymbol{P}, c) = \frac{\partial L(\{\boldsymbol{R_i}^T \boldsymbol{p'_i} - \boldsymbol{T_i}\}_{i=1}^N, c)}{\partial \{\boldsymbol{p'_i}\}_{i=1}^N}. \tag{6}$$

After the coordinate transformation, the original coordinate system is transformed into the normal vector and tangent plane coordinate system for each point in the point cloud, which is convenient for regularizing the update gradient. In order to keep the local geometry, we consider two assumptions to regularize the update gradient. The first assumption is to constrain the update gradient along the large curvature direction, while the second assumption is to remove the update gradient along the positive normal direction.

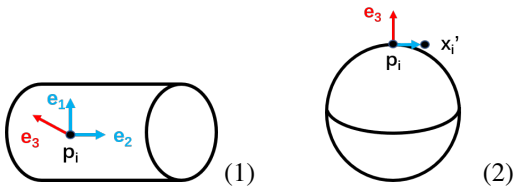

Figure 3: Observations for assumptions: (1) Perturbations along the small curvature direction ($\boldsymbol{e_2}$) on the tangent plane keep the local shape. (2) Perturbations along the negative normal direction ($-\boldsymbol{e_3}$) keep the local shape.

From Figure 3 (1), we can derive our first assumption. Specifically, if we alter the point on the boundary of the cylinder along the direction $\boldsymbol{e_2}$, the shape of the cylinder will never change. If we perturb the point along the direction $\boldsymbol{e_1}$, the shape will greatly change. Therefore, with the aim of keeping the local geometry, we propose to regularize the update gradient on the tangent plane by reducing the update gradient along the large curvature direction.

Assumption 1. *The perturbation along smaller curvature directions changes less on the local shape.*

From Figure 3 (2), we can derive our second assumption. Specifically, if the updating direction on the tangent plane is $\boldsymbol{x'}$, shifting the point along the negative normal direction ($-\boldsymbol{e_3}$) is consistent with the local shape. If we perturb the point along the positive normal direction, the local shape will greatly change. As a result, to keep the local shape, we propose to regularize the update gradient along the normal direction by only allowing the update gradient along the negative normal direction.

Assumption 2. *The perturbation along the negative normal direction changes less on the local shape.*

Based on these two assumptions, we regularize the update gradient to keep the local shape of the original point cloud. We detail our regularization scheme on the tangent plane and along the normal direction as follows.

Gradient Regularization on the Tangent Plane. We propose an adaptive gradient regularization scheme on the tangent plane based on the first assumption. To preserve the local shape, we should reduce the perturbation along the large curvature direction on the tangent plane. According to the property of differential geometry, the curvature of any direction on the tangent plane is bounded by the two principal curvatures. Therefore, we should reduce the update gradient along the maximum principal direction. Besides, we note that if the difference between the two principal curvatures is large, removing the update gradient along the maximum principal direction can largely preserve the local shape. However, if the maximum principal curvature is similar to the minimum principal curvature, perturbations along the maximum principal direction achieve similar changes to the local shape with those along the minimum principal direction. Therefore, to trade off imperceptibility and attack ef-

fectiveness, we keep the update gradient along a linear combination of two principal directions with more weights on the minimum principal direction.

We first detail how to conveniently reduce the update gradient along the large curvature direction. From Section , we can see that the parameters a and b determine the transformed directions. Therefore, we can rectify the update gradient on the tangent plane by tuning the parameters a and b and only keeping the update gradient along the $\boldsymbol{x'}$ direction. For example, we can set $a = 0$ and $b = 1$ to align the new $\boldsymbol{x'}$-axis with the minimum principal direction (i.e., $\boldsymbol{x'} = \boldsymbol{e_2}$) and the new $\boldsymbol{y'}$-axis with the maximum principal direction (i.e., $\boldsymbol{y'} = \boldsymbol{e_3}$). Afterwards, we can remove the update gradient along the $\boldsymbol{y'}$-axis to only allow the perturbation along the $\boldsymbol{x'}$-axis, which is the minimum principal direction.

We then detail how to trade off imperceptibility and attack effectiveness. We define the curvature ratio by the following equation:

$$cr = \frac{\lambda_2}{\sqrt{\lambda_1^2 + \lambda_2^2}}. \tag{7}$$

The curvature ratio satisfies the inequality $0 < cr \leq \frac{1}{\sqrt{2}}$, since $0 < \lambda_2 \leq \lambda_1$.

To identify the small difference between the two principal curvatures, we utilize a hyper-parameter γ. If the curvature ratio is smaller than γ, the curvature difference is defined to be large. In this case, we completely remove the gradient along the maximum principal direction to keep the local shape. In contrast, if the curvature ratio is larger than γ, the curvature difference is defined to be small. In this case, we do not need to completely remove the gradient along the maximum principal direction. Instead, we deploy the adaptive gradient direction by taking the curvature ratio into consideration. Specifically, we set the parameters $a = \frac{\lambda_2}{\sqrt{\lambda_1^2+\lambda_2^2}}$ and $b = \frac{\lambda_1}{\sqrt{\lambda_1^2+\lambda_2^2}}$. As such, we combine the two principal directions with more weights on the minimum principal direction. We denote the update direction as the balanced principal direction.

In summary, the gradient update direction $\boldsymbol{x'}$ is:

$$\boldsymbol{x'} = \begin{cases} \boldsymbol{e_2} & cr < \gamma \\ \frac{\lambda_2}{\sqrt{\lambda_1^2+\lambda_2^2}} \cdot \boldsymbol{e_1} + \frac{\lambda_1}{\sqrt{\lambda_1^2+\lambda_2^2}} \cdot \boldsymbol{e_2} & cr \geq \gamma \end{cases} \tag{8}$$

With the aim of rectifying the update gradient along the balanced principal direction, we can directly modify the gradient of the point i by the following equations:

$$\begin{aligned} g_{i1} &= g_{i1}, \\ g_{i2} &= 0. \end{aligned} \tag{9}$$

Gradient Regularization along the Normal Direction. We describe the gradient rectification along the normal direction based on the second assumption. With the objective to only allow the gradient along the negative normal direction, we can directly rectify the update gradient by the formula:

$$g_{i3} = \min(g_{i3}, 0) \tag{10}$$

Algorithm 1: Curvature-Invariant Method

Input: input point cloud $\boldsymbol{P}$ and its ground-truth label c
Input: the classifier f, attack budget ϵ, and iteration T
Input: hyper-parameter γ and loss function L
Output: adversarial point cloud $\boldsymbol{P_T}$
$\alpha = \frac{\epsilon}{T}$, $\boldsymbol{P_0} = \boldsymbol{P}$
for $t = 0 \leftarrow T - 1$ **do**
  Compute (λ_1, λ_2) and $(\boldsymbol{e_1}, \boldsymbol{e_2}, \boldsymbol{e_3})$ ▷ Eq. (1)
  $cr = \frac{\lambda_2}{\sqrt{\lambda_1^2+\lambda_2^2}}$
  if $cr < \gamma$ **then**
    $a = 0, b = 1$
  else
    $a = \frac{\lambda_2}{\sqrt{\lambda_1^2+\lambda_2^2}}, b = \frac{\lambda_1}{\sqrt{\lambda_1^2+\lambda_2^2}}$
  end if
  Transform $\boldsymbol{P_t}$ to $\boldsymbol{P'_t}$ ▷ Eq. (2)
  $\boldsymbol{G} = \frac{\partial L(\{\boldsymbol{R_t}^T \boldsymbol{P'_t} - \boldsymbol{T_t}\})}{\partial \{\boldsymbol{P'_t}\}}$
  $\boldsymbol{G_2} = \boldsymbol{0}$ ▷ Eq. (9)
  $\boldsymbol{G_3} = \min(\boldsymbol{G_3}, \boldsymbol{0})$ ▷ Eq. (10)
  $\boldsymbol{P'_{t+1}} = \boldsymbol{P'_t} - \alpha \cdot \frac{\boldsymbol{G}}{\|\boldsymbol{G}\|_1}$
  Transform $\boldsymbol{P'_{t+1}}$ to $\boldsymbol{P_{t+1}}$ ▷ Eq. (2)
  $\boldsymbol{P_{t+1}} = Clip^{\epsilon}\{\boldsymbol{P_{t+1}}\}$
end for

In a nutshell, we rectify the update gradient of each point in the point cloud by the constraints on the tangent plane and the normal direction. Our overall attacking algorithm is shown in Algorithm 1.

In our Curvature-Invariant Method, we can compute the upper bound of the variation of the loss function for each point in one iteration.

Theorem 3. *Given the loss function L and the variable point i in the point cloud (x'_i, y'_i, z'_i) initialized as $(p'_{i1}, p'_{i2}, p'_{i3})$. The variation of L is upper bounded by $\sqrt{g_{i1}^2 + g_{i2}^2}$.*

Experiments

In this section, we conduct extensive experiments to validate the effectiveness of our proposed Curvature-Invariant Method. We first clarify the setup of the experiments. After that, we demonstrate the white-box attacking performance and the imperceptibility measures of our method against competitive baseline methods. We also compare the attack effectiveness on defense models. The experiment results demonstrate the effectiveness of our methods that both improve the attack success rate and the imperceptibility of adversarial examples compared with baseline methods. Furthermore, we present the ablation study on the attack budget to further demonstrate the superiority of our approach in terms of attacking performance and imperceptibility.

Experiment Setup

We follow the protocol of the baseline method (Huang et al. 2022) to set up the experiments for a fair comparison to attack 3D point cloud classification models trained on Model-

Methods	PointNet				PointNet++				PointConv				DGCNN			
	ASR	MSE	D_H	D_G	ASR	MSE	D_H	D_G	ASR	MSE	D_H	D_G	ASR	MSE	D_H	D_G
3d-ADV	90.6	3.19	4.05	6.88	92.8	4.44	4.13	12.22	88.3	4.36	4.01	12.80	95.8	5.39	4.11	15.00
GeoA	92.5	2.16	3.47	5.35	94.0	2.61	3.20	7.38	92.5	3.07	3.59	9.90	95.7	3.34	3.12	8.51
SI-PC	94.2	1.83	3.45	4.54	91.9	2.83	3.49	8.57	93.2	3.15	3.68	9.47	96.3	3.44	3.12	7.97
CIM	**96.3**	**1.82**	**3.29**	**4.18**	**94.9**	**2.55**	**3.13**	**6.68**	**94.8**	**2.93**	**3.36**	**8.00**	**96.5**	**3.25**	**2.96**	**6.96**

Table 1: The attacking performance on ModelNet40. ASR is the attatck success rate (%) and MSE is the mean square errors. D_H measures the Hausdorff distance (10^{-2}) and D_G shows the Gaussian curvature distance (10^{-4}).The best result is in bold.

Net40 (Wu et al. 2015). ModelNet40 is also the most widely utilized benchmark task for 3D point cloud adversarial attacks (Wen et al. 2020; Xiang, Qi, and Li 2019; Huang et al. 2022). Here are the details of the experiment setup.

Dataset. We follow the dataset selection of the baseline method (Huang et al. 2022) by utilizing the dataset ModelNet40. ModelNet40 consists of 12,311 CAD models from 40 object categories, in which 9,843 models are intended for training and the other 2,468 for testing. Following the preprocessing of the PointNet (Qi et al. 2017a), we uniformly sample 1,024 points from the surface of each object and rescale them into a unit cube.

Models. We choose four representative 3D point cloud recognition models containing PointNet (Qi et al. 2017a), PointNet++ with MSG (Qi et al. 2017b), PointConv (Wu, Qi, and Fuxin 2019) and DGCNN (Wang et al. 2019) as the target model to craft adversarial point clouds and directly test the models under the white-box setting. Furthermore, we also consider the defended models as the target ones. We select three defense methods covering input preprocessing-based defense SRS, point cloud statistical outlier removal SOR and DUP-Net (Yang et al. 2019).

Baseline Methods. We compare our approach with three state-of-the-art attacking algorithms: 3d-ADV (Xiang, Qi, and Li 2019), GeoA (Wen et al. 2020), and SI-PC (Huang et al. 2022). 3d-ADV and GeoA are optimization approaches, which incorporate different quality measures like MSE (Xiang, Qi, and Li 2019), Hausdorff Distance (Taha and Hanbury 2015), and local curvature (Wen et al. 2020)) into the loss function to guarantee the quality of adversarial point cloud. While, SI-PC is a gradient regularization approach, which drops out the gradient along the normal direction to keep the shape of the adversarial point cloud. We compare our approach with them under various settings to validate the effectiveness of our method.

Evaluation. We first evaluate the imperceptibility of the crafted adversarial point cloud from two perspectives. We compare the l_2 distance (MSE) and Hausdorff Distance D_H (Taha and Hanbury 2015) between the original point cloud and the adversarial point cloud to measure the perturbation generated by the attacking methods. We also evaluate the imperceptibility from the point of view of the local surface that we compute the difference of the Gaussian curvature D_G (Do Carmo 2016) between the original point cloud and the adversarial one by following (Miao et al. 2022), which is the difference of the two principle curvatures multiplication. In addition to the measurement of imperceptibility, we also evaluate the attacking performance by deploying the attack success rate (ASR). The attack success rate is the ratio of the adversarial examples that successfully mislead the target model among all the generated adversarial examples. All the experiments are conducted on a server equipped with one TITAN X GPU.

Attack	SOR	SRS	DUP-Net	Average
3d-ADV	56.5	56.5	58.3	57.1
GeoA	60.6	63.3	60.8	61.6
SI-PC	77.7	70.0	79.3	75.7
CIM	**78.4**	**73.7**	**80.4**	**77.5**

Table 2: The attack success rates (%) of the adversarial point clouds on three defense mechanisms. The examples are generated on the PointNet model and the best result is in bold.

Parameter. For a fair comparison, we set the maximum L_∞ budget of all the attacking methods to be $\epsilon = 0.16$. In addition, the number of iterations is set to be $T = 5$, and the step length is 0.07. In the experiment, we adopt the untargeted attack under the same setting to evaluate the imperceptibility and attacking performance. For our approach, we set the hyper-parameter γ to regularize the gradient on the tangent plane to be 0.3.

Performance Comparison

In this section, we analyze the performance of our approach against the state-of-the-art baselines from the perspective of imperceptibility and attack success rate, respectively.

As shown in Table 1, our approach achieves the highest 95.6% white-box attacking success rate on average compared with all the baselines. In addition, our method outperforms all the other baselines on all three measures of imperceptibility, demonstrating the high quality of adversarial samples generated by our approach. Especially, we outperform the other baselines on the measure of Hausdorff distance and the Gaussian curvature with a large margin of 7.2% and 14.5% improvement respectively. Though GeoA considers taking the curvature into the loss function, the complex compound loss terms hinder the attacking algorithm from achieving high quality, and GeoA disregards the different perceptibility of distortion along different directions. Furthermore, SI-PC regularizes the gradient by allowing the perturbation along the tangent plane to keep the local shape. However, our approach takes the curvature into consideration, and we propose to constrain the gradient on the tangent plane with large curvature to preserve the local shape. Furthermore, we allow the negative gradient along the normal direction to further enhance the performance.

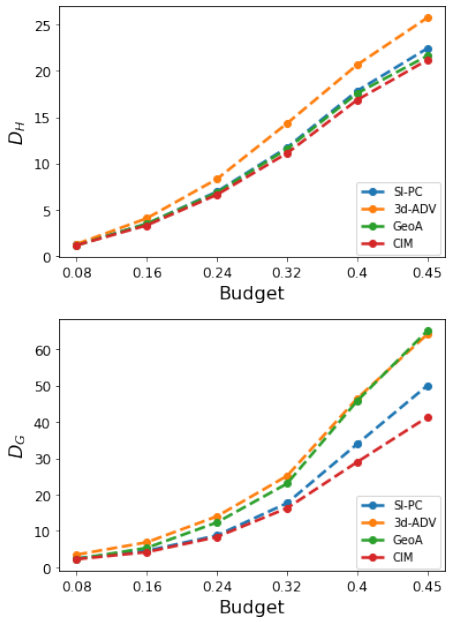

Figure 4: Ablation study on budget.

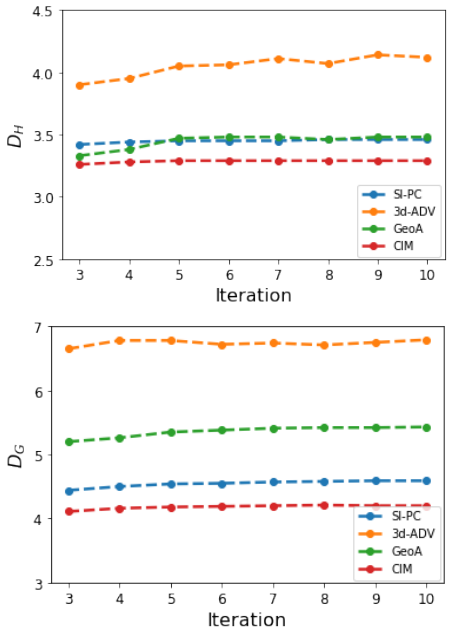

Figure 5: Ablation study on iteration.

In addition, we assess the performance against the model with defense mechanisms. We take PointNet as the source model and generate adversarial examples for all the baseline methods. Then we test the prediction accuracy of adversarial examples on the PointNet with defense methods as shown in Table 2. Our proposed method achieves a 77.5 % attack success rate on average and surpasses all of the baselines with a margin of 1.8%.

From the above experiments, our proposed method has more imperceptibility compared with baselines. We conclude the reasons why CIM has good imperceptibility are two-folded. Firstly, CIM attempts to utilize the information of local curvature to preserve the local geometry of 3D point clouds. We further regularize the update gradient by reducing the gradient along the large curvature direction and only keeping the gradient along the negative normal direction.

Qualitative Results

We further visualize the adversarial point clouds to show the qualitative results. We observe from Figure 1, our attacking algorithm preserves the local curvature well. We can hardly find any outliers on the generated adversarial point clouds of our approach. Notably, our approach can preserve the local shape compared with 3d-ADV and SI-PC. Furthermore, our method has fewer outliers compared with GeoA. The qualitative further validates the good imperceptibility of our proposed approach against all the baselines.

Ablation Study

We do ablation studies on the influence of two branch factors 1) Inner factor: the regularization on the tangent plane and normal direction. We want to see the contribution of each regularization to the imperceptibility and attacking performance. 2) Outer factor: the query budget and iteration time.

Attack	MSE	D_H	D_G
None	1.83	3.65	5.29
Normal Regularization	1.83	3.63	5.33
Tangent Regularization	1.83	3.36	4.24
Tangenmt+Normal (our)	**1.82**	**3.29**	**4.18**

Table 3: The results of the ablation study.

Regularization. We do an ablation study on the gradient regularization of CIM and observe the imperceptibility to show the effectiveness of the gradient regularization on both the tangent plane and the normal direction. We choose PointNet as the source model with different regularization strategies. As shown in Table 3, regularizing the gradient on the tangent can largely enhance the imperceptibility, while constraining the gradient on the normal direction lonely does not boost the imperceptibility. However, combining the regularization together benefits the improvement of imperceptibility, which is consistent with our two assumptions.

Query Budget & Iteration Time. We measure the performance of adversarial examples generated from the PointNet model by altering the factors. We observe from Figure 4 and Figure 5 that our attacking algorithm outperforms all the baselines under all the outer factor settings.

Conclusion

In this paper, we find that current attacking methods fail to keep the local shape of the adversarial point cloud. Therefore, we propose the curvature-invariant method by constraining the gradient on the tangent plane along a small curvature direction and eliminating the negative gradient along the normal direction. Our approach boosts both the attack imperceptibility and the attack success rate.

Acknowledgments

The work described in this paper was supported by the National Natural Science Foundation of China (Grant No. 62206318) and the Research Grants Council of the Hong Kong Special Administrative Region, China (CUHK 14206921 of the General Research Fund).

References

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

Deng, Y.; Wu, W.; Zhang, J.; and Zheng, Z. 2023. Blurred-Dilated Method for Adversarial Attacks. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Do Carmo, M. P. 2016. *Differential geometry of curves and surfaces: revised and updated second edition*. Courier Dover Publications.

Dong, S.; Wang, P.; and Abbas, K. 2021. A survey on deep learning and its applications. *Computer Science Review*, 40: 100379.

Dong, X.; Chen, D.; Zhou, H.; Hua, G.; Zhang, W.; and Yu, N. 2020. Self-robust 3d point recognition via gather-vector guidance. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11513–11521. IEEE.

Fan, H.; Su, H.; and Guibas, L. J. 2017. 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*, 605–613.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Hamdi, A.; Rojas, S.; Thabet, A.; and Ghanem, B. 2020. Advpc: Transferable adversarial perturbations on 3d point clouds. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*, 241–257. Springer.

Hoppe, H.; DeRose, T.; Duchamp, T.; McDonald, J.; and Stuetzle, W. 1992. Surface reconstruction from unorganized points. In *Proceedings of the 19th annual conference on computer graphics and interactive techniques*, 71–78.

Huang, Q.; Dong, X.; Chen, D.; Zhou, H.; Zhang, W.; and Yu, N. 2022. Shape-invariant 3D Adversarial Point Clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15335–15344.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755. Springer.

Liu, D.; and Hu, W. 2022. Imperceptible transfer attack and defense on 3d point cloud classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Liu, D.; Yu, R.; and Su, H. 2019. Extending adversarial attacks and defenses to deep 3d point cloud classifiers. In *2019 IEEE International Conference on Image Processing (ICIP)*, 2279–2283. IEEE.

Miao, Y.; Dong, Y.; Zhu, J.; and Gao, X.-S. 2022. Isometric 3D Adversarial Examples in the Physical World. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252.

Sun, J.; Cao, Y.; Choy, C. B.; Yu, Z.; Anandkumar, A.; Mao, Z. M.; and Xiao, C. 2021. Adversarially robust 3d point cloud recognition using self-supervisions. *Advances in Neural Information Processing Systems*, 34: 15498–15512.

Taha, A. A.; and Hanbury, A. 2015. Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. *BMC medical imaging*, 15(1): 1–28.

Tu, J.; Ren, M.; Manivasagam, S.; Liang, M.; Yang, B.; Du, R.; Cheng, F.; and Urtasun, R. 2020. Physically realizable adversarial examples for lidar object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13716–13725.

Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5): 1–12.

Wen, Y.; Lin, J.; Chen, K.; Chen, C. P.; and Jia, K. 2020. Geometry-aware generation of adversarial point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6): 2984–2999.

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

Wu, W.; Su, Y.; Chen, X.; Zhao, S.; King, I.; Lyu, M. R.; and Tai, Y.-W. 2020a. Boosting the transferability of adversarial samples via attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1161–1170.

Wu, W.; Su, Y.; Chen, X.; Zhao, S.; King, I.; Lyu, M. R.; and Tai, Y.-W. 2020b. Towards global explanations of convolutional neural networks with concept attribution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8652–8661.

Wu, W.; Su, Y.; Lyu, M. R.; and King, I. 2021. Improving the transferability of adversarial samples with adversarial transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9024–9033.

Wu, W.; Xu, H.; Zhong, S.; Lyu, M. R.; and King, I. 2019. Deep Validation: Toward detecting real-world corner cases for deep neural networks. In *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 125–137. IEEE.

Wu, W.; Zhang, J.; Wei, V. J.; Chen, X.; Zheng, Z.; King, I.; and Lyu, M. R. 2023. Practical and Efficient Model Extraction of Sentiment Analysis APIs. In *IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, 524–536. IEEE.

Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1912–1920.

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

Yang, J.; Zhang, Q.; Fang, R.; Ni, B.; Liu, J.; and Tian, Q. 2019. Adversarial attack and defense on point sets. *arXiv preprint arXiv:1902.10899*.

Yi, L.; Kim, V. G.; Ceylan, D.; Shen, I.-C.; Yan, M.; Su, H.; Lu, C.; Huang, Q.; Sheffer, A.; and Guibas, L. 2016. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35(6): 1–12.

Zhang, J.; Huang, J.-t.; Wang, W.; Li, Y.; Wu, W.; Wang, X.; Su, Y.; and Lyu, M. R. 2023a. Improving the Transferability of Adversarial Samples by Path-Augmented Method. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8173–8182.

Zhang, J.; Huang, Y.; Wu, W.; and Lyu, M. R. 2023b. Transferable Adversarial Attacks on Vision Transformers with Token Gradient Regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16415–16424.

Zhang, J.; Huang, Y.-C.; Wu, W.; and Lyu, M. R. 2023c. Towards semantics-and domain-aware adversarial attacks. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 536–544.

Zhang, J.; Wu, W.; Huang, J.-t.; Huang, Y.; Wang, W.; Su, Y.; and Lyu, M. R. 2022. Improving Adversarial Transferability via Neuron Attribution-Based Attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14993–15002.

Zhang, J.; Xu, Z.; Cui, S.; Meng, C.; Wu, W.; and Lyu, M. R. 2023d. On the Robustness of Latent Diffusion Models. *arXiv preprint arXiv:2306.08257*.

Zheng, T.; Chen, C.; Yuan, J.; Li, B.; and Ren, K. 2019. Pointcloud saliency maps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1598–1606.

Zhou, H.; Chen, D.; Liao, J.; Chen, K.; Dong, X.; Liu, K.; Zhang, W.; Hua, G.; and Yu, N. 2020. 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*, 10356–10365.

Zhou, H.; Chen, K.; Zhang, W.; Fang, H.; Zhou, W.; and Yu, N. 2019. Dup-net: Denoiser and upsampler network for 3d adversarial point clouds defense. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1961–1970.