# FLAT: Flux-aware Imperceptible Adversarial Attacks on 3D Point Clouds

Keke Tang[1*], Lujie Huang[1*], Weilong Peng[1(✉)], Daizong Liu[2], Xiaofei Wang[3], Yang Ma[1], Ligang Liu[3], and Zhihong Tian[1]

[1] Guangzhou University, Guangzhou, China
{tangbohutbh,huanglujiekz,gzhumayang}@gmail.com,
{wlpeng,tianzhihong}@gzhu.edu.cn
[2] Peking University, Beijing, China
dzliu@stu.pku.edu.cn
[3] University of Science and Technology of China, Hefei, China
wxf9545@mail.ustc.edu.cn, lgliu@ustc.edu.cn

**Abstract.** Adversarial attacks on point clouds play a vital role in assessing and enhancing the adversarial robustness of 3D deep learning models. While employing a variety of geometric constraints, existing adversarial attack solutions often display unsatisfactory imperceptibility due to inadequate consideration of uniformity changes. In this paper, we propose FLAT, a novel framework designed to generate imperceptible adversarial point clouds by addressing the issue from a flux perspective. Specifically, during adversarial attacks, we assess the extent of uniformity alterations by calculating the flux of the local perturbation vector field. Upon identifying a high flux, which signals potential disruption in uniformity, the directions of the perturbation vectors are adjusted to minimize these alterations, thereby improving imperceptibility. Extensive experiments validate the effectiveness of FLAT in generating imperceptible adversarial point clouds, and its superiority to the state-of-the-art methods.

**Keywords:** Adversarial attacks · Point clouds · Imperceptibility · Flux

## 1 Introduction

With advancements in deep learning techniques [20, 39] and the growing availability of affordable depth-sensing devices, deep neural network (DNN)-driven 3D point cloud perception has become a leading approach. Nevertheless, recent studies have highlighted the vulnerability of DNN classifiers to adversarial attacks [26,50]. Notably, subtle perturbations to the input point clouds can result in incorrect model predictions, posing significant challenges for real-world deployment. Consequently, exploring adversarial attacks on point clouds is essential for evaluating and improving the adversarial robustness of 3D deep learning models.

In achieving imperceptibility for adversarial attacks on 3D point clouds, methods predominantly fall into two categories. The first category encompasses

*Joint first authors. (✉)Corresponding author.

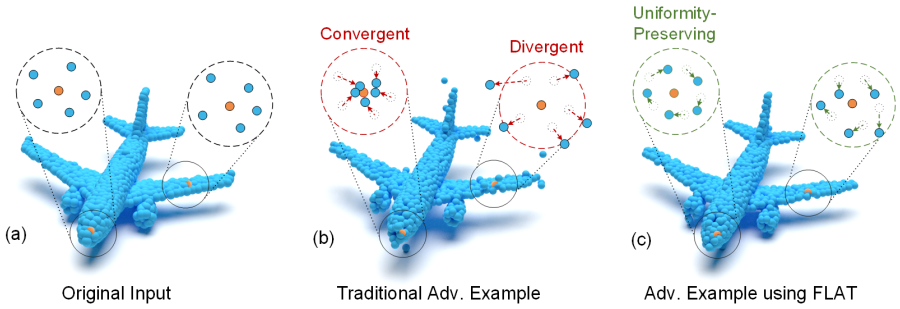


**Fig. 1:** Visualization of adversarial attacks on 3D point clouds: (a) the original point cloud; (b) the adversarial point cloud generated by SI-Adv; (c) the adversarial point cloud generated by FLAT. Notably, FLAT achieves greater imperceptibility by preserving the uniformity of the point cloud, as compared to SI-Adv.

the incorporation of recognizable geometries, such as spheres or airplanes [50], or the application of physical-intuitive deformations [42], both designed to be hidden in the human psyche. The second, more widely-adopted category, emphasizes constraints to minimize perturbations. Traditional methods within this paradigm employ metrics such as the $l_2$-norm, Chamfer distance, and Hausdorff distance. Recent advancements, however, have sought to harness the intrinsic geometric properties of 3D point clouds, focusing on minimizing distortions by constraining geometric regularity [46] or guiding perturbations along established normal [25] or tangential directions [16]. While the distortion has been notably reduced, traces of adversarial attacks remain perceptible.

This raises a question: why are adversarial point clouds still perceptible under these geometric constraints? We observe that adversarial manipulation, even under these constraints, significantly alters the uniformity of the perturbed samples relative to their original configuration, see the zoomed areas of divergence and convergence in Fig. 1(b). Since the human eye can easily notice changes in uniformity, mitigating these variations during adversarial attacks could enhance their imperceptibility. However, preserving uniformity presents a challenge as it is a regional attribute, characterized by the anisotropic perturbation of points [4], rather than a property of individual points. This distinction is often overlooked by traditional methods.

In this paper, we introduce a novel **FL**ux-aware imperceptible adversarial **AT**tacks (FLAT) on 3D point clouds. By attributing the issue of significant alterations in uniformity to irregular flow of points, we propose to handle it through a novel flux perspective. Specifically, we adapt the flux concept to perturbation vectors, creating a simplified flux measure to monitor the uniformity changes, e.g., divergence within local regions, during attacks. When encountering significant flux, we suppress it by fine-tuning the directions of the perturbation vectors for preserving uniformity. This flux-aware refinement effectively results in enhanced imperceptibility, as illustrated in Fig. 1(c). We validate the effec-

tiveness of our FLAT in attacking common DNN classifiers for 3D point clouds under various metrics. Extensive experimental results show that the adversarial point clouds generated by FLAT are significantly more imperceptible than those generated by state-of-the-art methods. Besides, we demonstrate that our flux-based approach can be readily integrated with other attack techniques to improve their imperceptibility.

Overall, our contribution is summarized as follows:

- We are the first to attribute the inadequate imperceptibility of adversarial attacks on 3D point clouds to the large deviation of uniformity.
- We develop a novel adversarial attack framework that preserves point cloud uniformity by suppressing the simplified flux of perturbation vectors.
- We show by experiments that our FLAT with preserving uniformity achieves superior performance in terms of imperceptibility under various metrics.

## 2 Related Work

**Adversarial Attacks on Point Clouds.** Adversarial attacks, designed to craft samples that mislead target models, were initially developed for 2D image classification [38] and subsequently adapted for 3D point clouds. These 3D point cloud attacks fall into three categories: addition-based attacks [50]; deletion-based attacks [47, 52, 54, 57]; and perturbation-based attacks [18, 50, 56]. Our study concentrates on the perturbation-based category.

Early perturbation-based attacks [26, 50] adapted C&W [5] and FGSM [11] from 2D to 3D. Zhao *et al*. [56] introduced isometric transformations for manipulating point clouds, and Kim *et al*. [18] focused on minimal point perturbations. Generative methods by Lee *et al*. [21] and Zhou *et al*. [58] explored latent space noise and GANs. Tang *et al*. [42] altered the 2-manifold surface. Despite high success rates, improving attack imperceptibility remains challenging.

**Imperceptibility of Adversarial Attacks.** To ensure the imperceptibility of attacks, common constraints include managing the $l_2$-norm, Chamfer and Hausdorff distance [26, 50, 58]. GeoA$^3$ [46] focuses on preserving geometric regularity. Directional perturbations guided by normal vectors [25, 41] and tangential plane perturbations [16] have also been explored. Tang *et al*. [40] adapted constraints to follow these directions. Our framework emphasizes the often-overlooked role of uniformity deviations, introducing flux metrics to measure and regulate changes in uniformity. Unlike GeoA$^3$ [46], which aims for uniform distribution, we ensure consistency in uniformity before and after perturbations.

**Uniformity of 3D Point Clouds.** Uniform distribution in 3D point clouds is essential for accurate geometry capture, high-quality mesh generation in surface reconstruction [2, 15], and effective noise removal in point cloud denoising while preserving structural integrity [13, 27]. It also improves segmentation and classification accuracy [19], and is crucial for applications like architectural modeling [44] and autonomous navigation [24]. This paper enhances the imperceptibility of adversarial point clouds by preserving uniformity during attacks.

**Deep Point Cloud Classification.** Deep learning techniques [10] for point cloud classification have significantly advanced [3, 7, 12, 36]. Initial approaches adapted 2D methodologies using 3D voxel grids [22,28]. The emergence of PointNet enabled direct point cloud processing [30], followed by innovations like hierarchical structures [31], point-specific convolutions [23, 43, 48, 51], and graph-based CNNs [6, 33, 34, 45, 55]. For more comprehensive reviews, refer to survey papers [12, 17]. Our research aims to attack these classifiers imperceptibly.

## 3 Problem Formulation

**Typical Adversarial Attacks.** Given a point cloud $\mathcal{P} \in \mathbb{R}^{n\times 3}$ and its label $y \in \{1, ..., K\}$, where $K$ is the category number, perturbation-based adversarial attack aims to mislead a 3D deep classification model $\mathcal{F}$ by feeding an adversarial point cloud $\mathcal{P}^{adv}$ instead of $\mathcal{P}$ via applying an intentionally designed perturbation, such that the model $\mathcal{F}$ makes an error prediction,

$$P_i^{adv} = P_i + \sigma_{P_i} \cdot \overrightarrow{d_{P_i}}, \tag{1}$$

where $\sigma_{P_i}$ is the perturbation size for the $i$-th point in $\mathcal{P}$, i.e., $P_i$, and $\overrightarrow{d_{P_i}}$ is the unit perturbation direction. Formally, the perturbation $\sigma_{P_i} \cdot \overrightarrow{d_{P_i}}$ can be obtained by solving the below general-form equation, e.g., via gradient descent,

$$\min L_{mis}(\mathcal{F}, \mathcal{P}^{adv}, y) + \lambda_1 D(\mathcal{P}, \mathcal{P}^{adv}), \tag{2}$$

where $L_{mis}(\cdot,\cdot,\cdot)$ is the loss to promote misclassification, e.g., the negation of cross-entropy loss, $\mathcal{P}^{adv}$ is the adversarial point clouds consists of $\{P_i^{adv}\}_{i=1:n}$, $D(\cdot,\cdot)$ is the constraints on distortion to facilitate imperceptibility, and $\lambda_1$ is a weighting parameter. Here, all referenced attacks are untargeted unless specified otherwise.

To achieve imperceptibility, adversarial attack solutions typically apply geometric constraints such as $l_2$-norm, Chamfer distance, Hausdorff distance, and curvature [46] to limit perturbations. However, these methods neglect an essential aspect: uniformity, a characteristic whose alterations are readily detected by the human eye.

**Uniformity-Preserving Adversarial Attacks.** To mitigate changes in uniformity, we formulate a novel type of *Uniformity-Preserving Adversarial Attacks*, building upon Eq. (2) with an additional constraint,

$$\text{Uniformity}(\Omega(P_i^{adv})) = \text{Uniformity}(\Omega(P_i)), \tag{3}$$

where $\Omega(P_i)$ denotes the local region centered at $P_i$, represented discretely by a set of points, and Uniformity$(\cdot)$ is the operator employed to compute uniformity.

By preserving the uniformity of local regions, this type of adversarial attack achieves better imperceptibility compared to conventional methods.

## 4 Method

In this section, we start by introducing the concept and mathematical definition of flux, as well as its association with point cloud uniformity. Following that, we

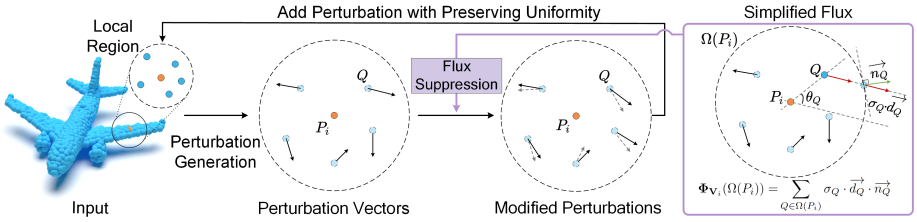


**Fig. 2:** Workflow of FLAT. Given an input point cloud, the model iteratively generates perturbation vectors for each local region. These vectors are then refined by measuring the simplified flux and applying suppression, thus ensuring the preservation of local shape uniformity.

delve into how flux can be utilized to maintain point cloud uniformity. Finally, we demonstrate the application of these principles in implementing the flux-aware imperceptible attacks.

### 4.1 Preliminary

**Definition of Flux.** Flux is commonly used to describe the magnitude of a vector field's flow through a surface [1]. Given a 3D vector field $\mathbf{F}$ and a simple, directed surface $\Omega$, the flux of $\mathbf{F}$ through $\Omega$ is defined as the integral of the component of the field vector $\overrightarrow{f}$ at each point on the surface in the direction of the surface's normal vector,

$$\mathbf{\Phi}_{\mathbf{F}}(\Omega) = \iint_{\Omega} \overrightarrow{f} \cdot \overrightarrow{n} \, \mathrm{d}S, \tag{4}$$

where $\mathrm{d}S$ represents the integration over infinitesimal segments of the surface $\Omega$, and $\overrightarrow{n}$ signifies the outward-pointing unit normal vector at each point on $\Omega$.
**Flux and Point Cloud Uniformity.** For a closed surface, by analogizing the movement of points within the region to the flow of a field, the flux values can reflect a trend towards divergence and convergence within that region. Hence, employing the flux concept holds promise for managing uniformity alterations in point clouds during adversarial attacks.

### 4.2 Flux-based Uniformity Preserving

For perturbation vectors $\mathbf{V} = \{\sigma_{P_i} \cdot \overrightarrow{d_{P_i}}\}_{i=1:n}$ calculated for an attack, our objective is to modify $\mathbf{V}$ in a way that, upon application, preserves the uniformity of the point clouds $\mathcal{P}$.

Given a subset of perturbation vectors within $\Omega(P_i)$, i.e., $\mathbf{V}_i = \{\sigma_Q \cdot \overrightarrow{d_Q} | Q \in \Omega(P_i)\}$, we employ the concept of flux to develop a metric that indicates uniformity changes in this subset. While a direct method would involve interpolating these vectors into a continuous field and integrating within $\Omega(P_i)$, as described

in Eq. (4), the irregularity and sparsity of the vectors make managing this interpolation and integration challenging. Therefore, we opt for a simplified approach that concentrates simply on the discrete vectors.

**Simplified Flux for Perturbation Vector Field.** In the region $\Omega(P_i)$, we assume the field intensity to be constant along the perturbation vector $\overrightarrow{d_Q}$, i.e., $\sigma_Q$, originating from point $Q$, see the right part of Fig. 2. In this configuration, each perturbation vector intersects a point on the boundary surface. Therefore, the simplified flux can be viewed as the integration of vectors at these specific intersection points,

$$\begin{aligned}\mathbf{\Phi}_{\mathbf{V}_i}(\Omega(P_i)) &= \sum_{Q\in\Omega(P_i)} \sigma_Q \cdot \overrightarrow{d_Q} \cdot \overrightarrow{n_Q}, \\ &= \sum_{Q\in\Omega(P_i)} A_Q\sqrt{1+B_Q\cos^2(\theta_Q)},\end{aligned} \tag{5}$$

where

$$A_Q = \frac{\sigma_Q \cdot |\overrightarrow{d_Q}|}{r}\sqrt{r^2-|\overrightarrow{P_iQ}|^2},$$

$$B_Q = \frac{|\overrightarrow{P_iQ}|^2}{r^2-|\overrightarrow{P_iQ}|^2},$$

$\overrightarrow{n_Q}$ is the unit outward normal vector at the intersection point on the boundary of $\Omega(P_i)$, where $\overrightarrow{d_Q}$ meets, starting from point $Q$. The angle $\theta_Q$ is defined between $\overrightarrow{d_Q}$ and $\overrightarrow{P_iQ}$, with $r$ representing the radius of the local shape.

**Flux Suppression by Rotating Perturbation Vectors.** To preserve the uniformity of point clouds during attacks, we suppress the simplified flux of the perturbation vector field. Specifically, we utilize gradient descent to adjust $\mathbf{\Theta}$,

$$\begin{aligned}\mathbf{\Theta}' &= \mathbf{\Theta} - \alpha\frac{\partial\mathbf{\Phi}_{\mathbf{V_i}}}{\partial\mathbf{\Theta}}, \\ \text{with}\quad \mathbf{\Theta} &= \{\theta_Q | Q\in\Omega(P_i)\}, \\ \frac{\partial\mathbf{\Phi}_{\mathbf{V_i}}}{\partial\theta_Q} &= -\frac{A_Q B_Q\sin(\theta_Q)\cos(\theta_Q)}{\sqrt{1+B_Q\cos^2(\theta_Q)}},\end{aligned} \tag{6}$$

where $\alpha$ denotes the step size for angle adjustment.

By suppressing flux in regions with significant values, point divergence is avoided. Essentially, by preventing divergence locally, convergence effects in other regions are also precluded. As a result, the uniformity of the point cloud is preserved after perturbation.

### 4.3 Flux-aware Imperceptible Adversarial Attacks

Given a point cloud $\mathcal{P}$, we utilize farthest point sampling (FPS) to select $m$ center points. Subsequently, local regions are constructed around these points,

each with a radius of $r$. Following this, we generate initial perturbation for each point in the point cloud, refine their directions to maintain uniformity from a flux perspective, and then determine their magnitudes, see Fig. 2.

**Generating Initial Perturbations.** We employ IFGM [8] to generate initial perturbations with a consistent magnitude. It is noteworthy that alternative methods could also be employed to similar effect.

**Refining Perturbation Directions with Flux.** Within each local region, we calculate the simplified flux. If the region exhibits a substantial flux value, exceeding a threshold $t$, we adjust the directions of the perturbation vectors within it as described in Sec. 4.2 to preserve uniformity.

**Determining Perturbation Magnitudes.** With the refined perturbation directions established, we proceed to determine perturbation magnitudes following [25]. By iteratively executing the above three steps, FLAT creates highly imperceptible adversarial point clouds. Note that we initially considered optimizing both perturbation direction and magnitude simultaneously for flux suppression. However, this approach tended to favor reducing perturbation magnitude. Thus, we opt for separate optimization.

## 5 Experimental Results

### 5.1 Experimental Setup

**Implementation.** The FLAT framework is implemented in PyTorch [29]. We start by sampling $m = 20$ key points using the farthest point sampling (FPS) method. Around these points, we define local regions with a radius of $r = 0.1$ for simplified flux computation. We focus on regions demonstrating significant flux, particularly those whose flux values surpass the threshold $t$, determined as the median flux among all regions. To suppress deviations in uniformity, we adjust the perturbation directions using a step size of $\alpha = 0.02$, over a total of 8 cycles. All experiments are executed on a workstation equipped with dual 2.40 GHz CPUs, 128 GB of RAM, and four NVIDIA RTX 3090 GPUs.

**Datasets.** We adopt two public datasets for evaluation: ModelNet40 [49] and ShapeNet Part [53]. For ModelNet40, we designate 9,843 point clouds for training and 2,468 for testing. For ShapeNet Part, we allocate 14,007 point clouds for training and 2,874 for testing. Particularly, we uniformly sample 1,024 points from the surface of each object and rescale them into a unit cube following [30].

**Victim Classifiers.** We use four well-established deep point cloud classifiers as victim models, including PointNet [30], PointNet++ [31], DGCNN [45], and PointConv [48]. We train these models according to their original papers.

**Baseline Attack Methods.** We assess the effectiveness of our approach by comparing it with six state-of-the-art techniques: the gradient-based IFGM [8] and PGD [8], the direction-based SI-Adv [16] and ITA [25], and the optimization-based methods GeoA$^3$ [46] and 3d-Adv [50]. This diverse set of attacks provides a robust baseline to validate the effectiveness of our approach.

**Evaluation Setting and Metrics.** To ensure fair comparisons, we configure each attack method to attain its maximum attack success rate (ASR), which is

**Table 1:** Comparison of perturbation sizes needed by various methods to achieve their highest ASR in untargeted attacks.

| Model | Attack | ModelNet40 | | | | | | | ShapeNet Part | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ASR (%) | CD ($10^{-4}$) | HD ($10^{-2}$) | $l_2$ | GR | Curv ($10^{-2}$) | EMD ($10^{-2}$) | ASR (%) | CD ($10^{-4}$) | HD ($10^{-2}$) | $l_2$ | GR | Curv ($10^{-2}$) | EMD ($10^{-2}$) |
| PointNet | PGD | 100 | 7.155 | 5.025 | 0.981 | 0.302 | 1.624 | 2.315 | 100 | 13.172 | 17.068 | 1.569 | 0.521 | 3.679 | 3.358 |
| | IFGM | 100 | 4.039 | 5.565 | 0.789 | 0.314 | 0.775 | 0.864 | 100 | 3.328 | 10.269 | 0.785 | 0.408 | 0.619 | 0.556 |
| | GeoA$^3$ | 100 | 4.646 | 0.497 | 1.307 | 0.121 | 0.396 | 2.319 | 100 | 7.531 | 1.444 | 2.655 | 0.146 | 0.465 | 4.104 |
| | 3d-Adv | 100 | 6.115 | 4.372 | 0.863 | 0.250 | 1.215 | 1.410 | 100 | 15.659 | 5.495 | 1.787 | 0.279 | 4.006 | 3.693 |
| | SI-Adv | 100 | 2.768 | 2.595 | 0.731 | 0.220 | 0.271 | 0.725 | 100 | 3.435 | 3.692 | 0.881 | 0.233 | 0.441 | 0.825 |
| | ITA | 100 | 2.747 | 0.414 | 0.534 | 0.122 | 0.555 | 1.214 | 100 | 5.872 | 1.917 | 1.002 | 0.181 | 1.016 | 2.035 |
| | Ours | 100 | **1.539** | **0.371** | **0.426** | **0.114** | **0.249** | **0.460** | 100 | **1.905** | **1.327** | **0.545** | **0.120** | **0.301** | **0.284** |
| PointNet++ | PGD | 100 | 5.182 | 0.636 | 0.753 | 0.125 | 1.508 | 2.146 | 100 | 10.090 | 3.257 | 1.342 | 0.215 | 3.969 | 3.328 |
| | IFGM | 100 | 3.558 | 1.162 | 0.640 | 0.146 | 1.149 | 1.454 | 100 | 4.532 | 3.608 | 0.548 | 0.220 | 1.824 | 1.584 |
| | GeoA$^3$ | 100 | 6.579 | 0.461 | 1.615 | 0.114 | 0.762 | 2.919 | 100 | 7.701 | 0.847 | 2.875 | 0.105 | 1.375 | 4.176 |
| | 3d-Adv | 100 | 8.915 | 3.564 | 1.535 | 0.141 | 1.288 | 2.784 | 100 | 9.564 | 3.778 | 2.014 | 0.197 | 3.021 | 3.590 |
| | SI-Adv | 100 | 9.399 | 2.377 | 1.422 | 0.185 | 1.061 | 2.684 | 100 | 9.266 | 3.233 | 1.535 | 0.203 | **1.146** | 2.811 |
| | ITA | 100 | 6.792 | 0.708 | 0.998 | 0.121 | 3.533 | 2.272 | 100 | 5.202 | 0.802 | 0.999 | 0.110 | 3.423 | 2.152 |
| | Ours | 100 | **1.156** | **0.325** | **0.337** | **0.110** | **0.373** | **0.491** | 100 | **3.067** | **0.729** | **0.349** | **0.102** | 1.701 | **1.295** |
| DGCNN | PGD | 100 | 19.968 | 5.098 | 1.933 | 0.267 | 4.924 | 4.785 | 100 | 63.556 | 27.557 | 5.224 | 0.511 | 7.275 | 9.233 |
| | IFGM | 100 | 15.791 | 12.391 | 1.622 | 0.363 | 2.849 | 3.777 | 100 | 19.623 | 26.040 | 2.069 | 0.504 | 4.954 | 4.387 |
| | GeoA$^3$ | 100 | 7.566 | 0.546 | 1.585 | 0.119 | 0.741 | 3.083 | 100 | 27.612 | 3.748 | 5.798 | 0.199 | 1.695 | 7.502 |
| | 3d-Adv | 100 | 10.345 | 3.807 | 3.589 | 0.227 | 5.997 | 6.685 | 100 | 21.553 | 8.531 | 2.258 | 0.282 | 5.119 | 4.628 |
| | SI-Adv | 100 | 7.146 | 1.691 | 1.087 | 0.143 | 0.666 | 2.495 | 100 | 11.685 | **3.019** | 1.772 | 0.160 | 2.054 | 3.646 |
| | ITA | 100 | 3.249 | 0.524 | 0.552 | 0.114 | 0.971 | 1.359 | 100 | 27.633 | 4.597 | 2.492 | 0.244 | 3.847 | 4.696 |
| | Ours | 100 | **2.576** | **0.420** | **0.540** | **0.100** | **0.556** | **1.027** | 100 | **9.003** | 4.693 | **1.653** | **0.135** | **0.112** | **3.465** |
| PointConv | PGD | 100 | 14.551 | 2.216 | 1.442 | 0.184 | 3.491 | 3.862 | 100 | 42.202 | 9.949 | 3.784 | 0.252 | 6.866 | 7.277 |
| | IFGM | 100 | 7.959 | 2.608 | 1.015 | 0.184 | 1.741 | 2.427 | 100 | 16.139 | 8.776 | 1.812 | 0.231 | 3.526 | 3.807 |
| | GeoA$^3$ | 100 | 6.809 | 0.644 | 2.169 | 0.119 | 1.119 | 3.556 | 100 | 9.383 | 1.222 | 4.224 | 0.120 | **1.190** | 5.391 |
| | 3d-Adv | 100 | 11.213 | 1.763 | 1.179 | 0.163 | 3.279 | 2.807 | 100 | 21.034 | 3.687 | 2.277 | 0.193 | 4.912 | 4.548 |
| | SI-Adv | 100 | 6.060 | 1.784 | 0.977 | 0.144 | 0.576 | 2.081 | 100 | 11.281 | 3.500 | 1.741 | 0.165 | 1.949 | 3.514 |
| | ITA | 100 | 5.539 | 0.480 | 0.833 | 0.111 | 1.904 | 1.971 | 100 | 9.082 | 1.452 | 1.375 | 0.146 | 3.645 | 2.925 |
| | Ours | 100 | **2.139** | **0.344** | **0.478** | **0.107** | **0.301** | **0.975** | 100 | **5.034** | **1.194** | **0.817** | **0.117** | 1.288 | **1.186** |

the percentage of adversarial point clouds that successfully fool the victim model. In this maximal adversarialness condition [37, 42], we assess the imperceptibility of attacks. Specifically, we employ six widely recognized metrics: Chamfer distance (CD) [9], Hausdorff distance (HD) [35], $l_2$-norm ($l_2$), curvature (Curv), geometric regularity (GR) [46], and earth mover's distance (EMD) [32]. Unless explicitly mentioned, all discussions regarding attack results are assumed to be about untargeted attacks.

### 5.2 Performance Comparison and Analysis

**Performance of Untargeted Attacks.** We evaluate the ASR and imperceptibility of various adversarial attack methods in an untargeted setting on the ModelNet40 and ShapeNet Part datasets, with results detailed in Tab. 1. It is observed that all these adversarial attack methods can achieve 100% ASR. However, methods like PGD, which impose larger perturbations in a single iteration, result in greater distortion and thus underperform across most metrics due to their lack of subtlety. In contrast, direction-based methods such as SI-Adv and ITA demonstrate lower distortion, showcasing their efficacy in maintaining attack stealth. Particularly, by modulating the flux of perturbation vector fields

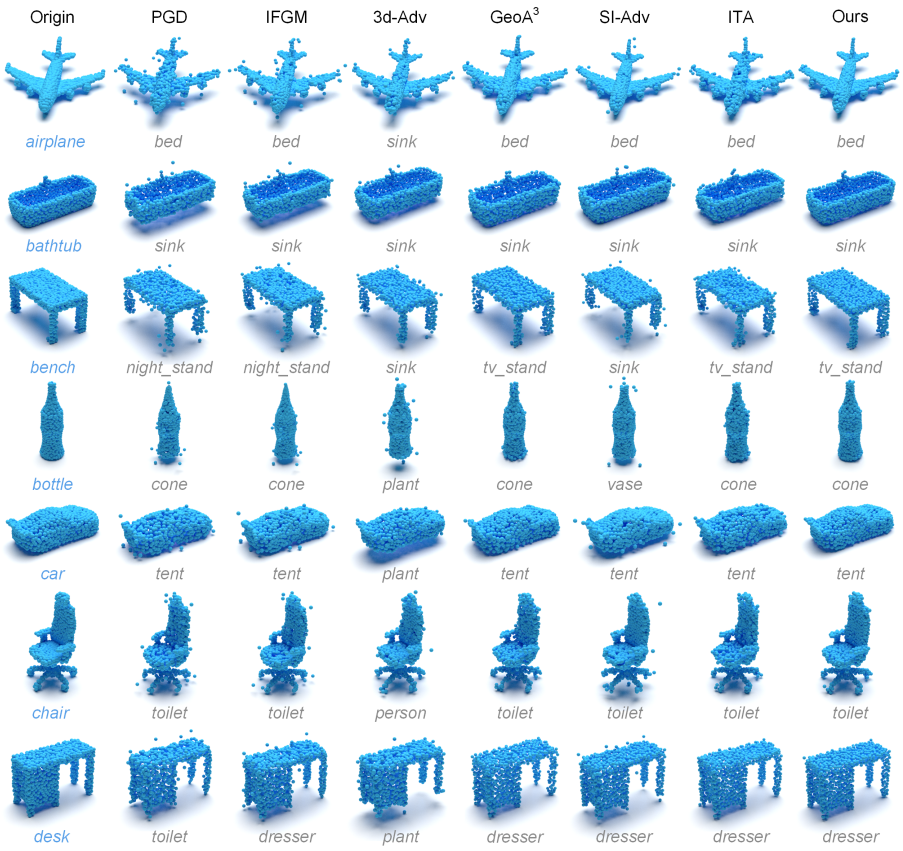


**Fig. 3:** Visualizations of original and adversarial point clouds generated to fool PointNet on ModelNet40 by various adversarial attack methods.

during attacks, our FLAT outperforms these state-of-the-art methods in the majority of metrics, confirming its effectiveness and superiority.

**Visualization of Adversarial Point Clouds.** To vividly showcase the enhanced imperceptibility afforded by our approach, we present visualizations of adversarial examples crafted using diverse attack methodologies for seven distinct classes from ModelNet40, designed to deceive PointNet, as depicted in Fig. 3. The adversarial point clouds generated by PGD and IFGM reveal pronounced outliers due to their relatively lax deformation constraints. In contrast, GeoA$^3$, which incorporates geometric constraints such as curvature, tends to produce samples with fewer outliers. SI-Adv and ITA, which utilize geometric characteristics for perturbation along tangential and normal vectors respectively, also result in fewer discernible outliers. Particularly, by restricting the flux of the perturbation vector field during the attack process, FLAT generates adversarial point clouds with surfaces that are noticeably more uniform and nearly devoid

**Table 2:** Comparison of perturbation sizes needed by various methods to achieve their highest ASR in targeted attacks against PointNet and DGCNN on ModelNet40.

| Attack | PointNet | | | | | | | DGCNN | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ASR (%) | CD ($10^{-4}$) | HD ($10^{-2}$) | $l_2$ | GR | Curv ($10^{-2}$) | EMD ($10^{-2}$) | ASR (%) | CD ($10^{-4}$) | HD ($10^{-2}$) | $l_2$ | GR | Curv ($10^{-2}$) | EMD ($10^{-2}$) |
| PGD | 100 | 24.428 | 2.042 | 2.630 | 0.180 | 1.787 | 5.548 | 100 | 65.040 | 16.099 | 9.644 | 0.416 | 2.528 | 14.893 |
| IFGM | 100 | 3.832 | 1.002 | 0.676 | 0.140 | 0.905 | 1.534 | 100 | 8.912 | 3.633 | 1.142 | 0.199 | 0.694 | 2.976 |
| GeoA$^3$ | 100 | 6.501 | **0.620** | 4.029 | 0.100 | 0.314 | 4.213 | 100 | 63.106 | 10.464 | 6.272 | 0.318 | 1.769 | 9.234 |
| 3d-Adv | 100 | 3.248 | 1.594 | 0.614 | 0.165 | 0.282 | 1.440 | 100 | 7.266 | 3.066 | **0.902** | 0.172 | 0.466 | 2.939 |
| SI-Adv | 100 | 4.185 | 1.795 | 1.054 | 0.144 | 0.301 | 1.594 | 100 | 13.974 | 12.761 | 2.141 | 0.184 | 1.309 | 3.511 |
| ITA | 100 | 38.081 | 1.889 | 1.969 | 0.186 | 2.011 | 4.055 | 100 | 48.321 | 1.623 | 1.542 | 0.193 | 2.351 | 3.553 |
| Ours | 100 | **2.344** | 1.044 | **0.594** | **0.091** | **0.212** | **0.793** | 100 | **6.198** | **0.617** | 1.533 | **0.109** | **0.357** | **2.668** |

**Table 3:** Comparison of uniformity changes measured by SDM [14] during attacks on four classifiers across ModelNet40 and ShapeNet Part.

| Attack | ModelNet40 | | | | ShapeNet Part | | | |
|---|---|---|---|---|---|---|---|---|
| | PointNet | PointNet++ | DGCNN | PointConv | PointNet | PointNet++ | DGCNN | PointConv |
| IFGM | 0.1493 | 0.1766 | 0.5049 | 0.2980 | 0.1201 | 0.2849 | 1.6553 | 0.2029 |
| GeoA$^3$ | 0.1956 | 0.2723 | 0.2799 | 0.3079 | 0.3823 | 0.3540 | 2.0009 | 0.4162 |
| SI-Adv | 0.1211 | 0.4115 | 0.3599 | 0.3548 | 0.1072 | 0.3488 | 1.1416 | 0.3776 |
| ITA | 0.1207 | 0.1756 | 0.1649 | 0.2698 | 0.2995 | 0.1778 | 0.4937 | 0.2465 |
| Ours | **0.0972** | **0.0785** | **0.1550** | **0.1420** | **0.0580** | **0.1480** | **0.2886** | **0.0953** |

of outliers, thereby affirming the effectiveness and superiority of our method in terms of imperceptibility.

**Performance of Targeted Attacks.** To further corroborate the superiority of our approach, we extend our evaluation to the targeted attack setting. Specifically, we randomly select 25 instances from each of the 10 categories in the ModelNet40 test set. For each instance, we craft adversarial examples targeting the remaining nine classes, resulting in a total of 2250 targeted attack point clouds, following the methodology of [50]. The results of these targeted attacks on PointNet and DGCNN are summarized in Tab. 2. It is observed that targeted attacks necessitate larger perturbations compared to the untargeted attacks presented in Tab. 1. Although ITA performed well in the untargeted setting, its relatively fixed perturbation pattern led to larger CD values in the targeted scenario. Our method remains consistent, outperforming other techniques across the majority of metrics, which further validates its effectiveness and superiority.

### 5.3 Ablation Studies and Other Analysis

**Importance of Flux in Uniformity Preserving.** To underscore the essential role of flux in preserving the uniformity of point clouds during adversarial attacks, we compare the uniformity changes induced by our method with those brought by state-of-the-art methods. Specifically, we employ the symmetric density metric (SDM) introduced in [14] to measure differences in nearest neighbor counts and average distances between original and adversarial point clouds. The

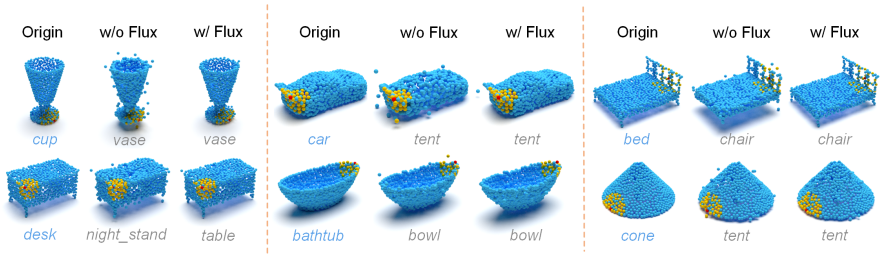


**Fig. 4:** Visualization of local regions of adversarial point clouds generated by FLAT with and without applying flux-based uniformity preservation in attacking PointNet.

**Table 4:** Comparison of our FLAT solution with a modified version, namely FLAT-SDM, that utilizes a uniformity-preserving constraint instead of flux adjustment.

| Attack | ASR (%) | CD ($10^{-4}$) | HD ($10^{-2}$) | $l_2$ | GR | Curv ($10^{-2}$) | EMD ($10^{-2}$) | SDM [14] | Iteration |
|---|---|---|---|---|---|---|---|---|---|
| FLAT-SDM | 100 | 2.502 | 1.713 | 0.608 | 0.214 | 0.452 | 0.565 | **0.084** | 15 |
| Ours | 100 | **1.539** | **0.371** | **0.426** | **0.114** | **0.249** | **0.460** | 0.097 | **8** |

results presented in Tab. 3 demonstrate that our method is more effective in preserving uniformity.

To better illustrate the effectiveness of our method in preserving uniformity, we present visualizations of local regions post-attack both with and without flux suppression. As depicted in Fig. 4, the center point is highlighted in red, while the surrounding points within its local region are marked in yellow. The adversarial samples generated with flux consideration exhibit only minimal deviations, closely resembling the original point cloud structure. Conversely, those produced without considering flux demonstrate noticeable deformations, emphasizing the crucial role of flux awareness in preserving point cloud uniformity.

**Flux-based vs. SDM-based Uniformity Preserving.** To further demonstrate the importance of our flux-based solution, we compare it with a variant of FLAT, named FLAT-SDM, that incorporates an SDM constraint [14] in the initial perturbation generation phase to create initial directions with better awareness of uniformity preservation, yet does not utilize flux for perturbation direction adjustment. The results in Tab. 4 indicate that directly applying constraints can indeed enhance uniformity. Nevertheless, the strong nature of the constraint complicates the execution of successful attacks. As a result, it requires 15 iterations, as opposed to the original 8, with these additional iterations leading to a deterioration in other imperceptibility metrics.

**Generalization of Flux-based Uniformity Preserving.** To evaluate the generalizability of our flux-based uniformity preservation approach, we incorporate it into four established iterative adversarial attack methods: PGD, IFGM [8], SI-Adv [16], and ITA [25]. As demonstrated in Tab. 5, these methods, when augmented with our flux-aware technique, exhibit marked improvements across most performance metrics under identical parameter settings. To illustrate the impact

**Table 5:** Comparison of perturbation sizes required by different methods, both with and without flux-based uniformity preservation, to achieve their highest ASR in untargeted attacks against PointNet and DGCNN on ModelNet40.

| Attack | PointNet | | | | | | | DGCNN | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ASR (%) | CD ($10^{-4}$) | HD ($10^{-2}$) | $l_2$ | GR | Curv ($10^{-2}$) | EMD ($10^{-2}$) | ASR (%) | CD ($10^{-4}$) | HD ($10^{-2}$) | $l_2$ | GR | Curv ($10^{-2}$) | EMD ($10^{-2}$) |
| PGD | 100 | 7.155 | 5.025 | 0.981 | 0.302 | 1.624 | 2.315 | 100 | 19.968 | 5.098 | 1.933 | 0.267 | 4.924 | 4.785 |
| PGD+flux | 100 | **6.259** | **3.721** | **0.894** | **0.265** | **1.529** | **2.220** | 100 | **18.100** | **3.285** | **1.799** | **0.221** | **4.678** | **4.549** |
| IFGM | 100 | 4.039 | 5.565 | 0.789 | 0.314 | 0.775 | 0.864 | 100 | 15.791 | 12.391 | 1.622 | 0.363 | 2.849 | 3.777 |
| IFGM+flux | 100 | **2.944** | **4.036** | **0.662** | **0.276** | **0.648** | **0.712** | 100 | **12.270** | **9.869** | **1.167** | **0.244** | **1.982** | **2.287** |
| SI-Adv | 100 | 2.768 | 2.595 | 0.731 | 0.220 | 0.271 | 0.725 | 100 | 7.146 | 1.691 | 1.087 | 0.143 | 0.666 | 2.495 |
| SI-Adv+flux | 100 | **2.165** | **1.914** | **0.580** | **0.187** | **0.202** | **0.609** | 100 | **7.055** | **1.676** | **1.037** | **0.126** | **0.658** | **2.309** |
| ITA | 100 | 2.747 | **0.414** | 0.534 | 0.122 | 0.555 | 1.214 | 100 | 3.249 | 0.524 | 0.552 | 0.114 | 0.971 | 1.359 |
| ITA+flux | 100 | **2.253** | 0.579 | **0.506** | **0.119** | **0.534** | **1.022** | 100 | **2.845** | **0.422** | **0.431** | **0.103** | **0.871** | **1.207** |

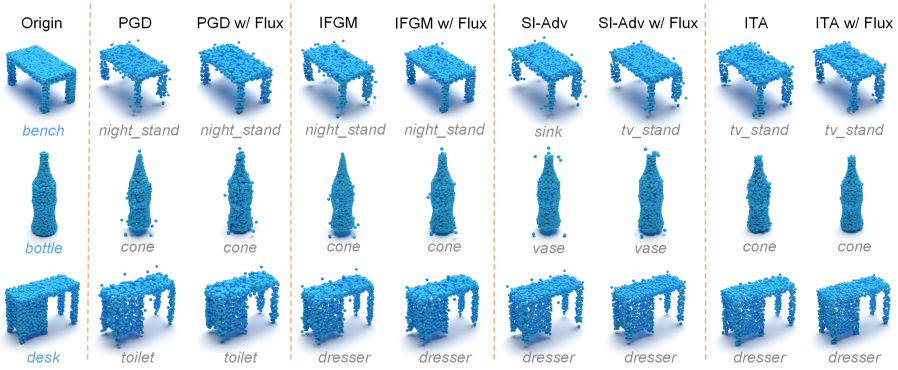


**Fig. 5:** Visualization of adversarial point clouds generated by various attack methods in attacking PointNet, with and without integrating flux-based uniformity preservation.

of flux-based uniformity preservation more vividly, we visualize the adversarial point clouds generated with and without this module in Fig. 5. It is evident that the integration of flux-based uniformity preservation significantly enhances the imperceptibility of the attack, exemplified by a substantial reduction in outliers. Consequently, we affirm the broad applicability of our flux-based uniformity preservation strategy.

**Impact of Initial Perturbation Directions.** We assess the influence of initial perturbation directions on the effectiveness of adversarial attacks by considering three alternatives: (1) gradient descent direction as implemented in IFGM [8]; (2) tangent plane direction as utilized in SI-Adv [16]; (3) normal vector direction as adopted in ITA [25]. The comparative analysis, depicted in Fig. 6, reveals only slight variations in the resulting distortions across these initial directions, underscoring our method's robustness. Notably, the gradient descent direction, as derived from IFGM, yield the best results; hence, we select it in our solution.

**Impact of Local Region Number and Size.** We evaluate the impact of local region number and size on the performance of our flux-based uniformity

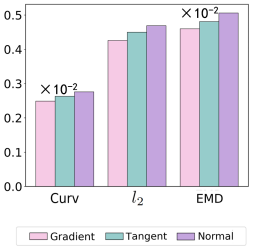


**Fig. 6:** Comparison of FLAT perturbation sizes with various initial directions.

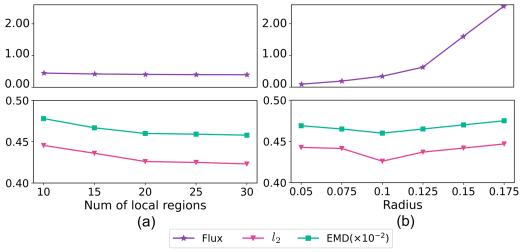


**Fig. 7:** Comparison of flux values and perturbation sizes for FLAT across varying numbers of local regions and different radii.

**Table 6:** Comparison of perturbation size and running time for FLAT in attacking PointNet using both formal and simplified flux methods.

| Flux | ASR (%) | CD ($10^{-4}$) | HD ($10^{-2}$) | $l_2$ | GR | Curv ($10^{-2}$) | EMD ($10^{-2}$) | Time (s) |
|---|---|---|---|---|---|---|---|---|
| Formal flux | 100 | 1.73 | 1.03 | 0.41 | 0.20 | 0.27 | 0.34 | 150.07 |
| Ours | 100 | 1.54 | 0.37 | 0.43 | 0.11 | 0.25 | 0.46 | 2.02 |

preservation. As illustrated in Fig. 7(a), augmenting the number of local regions corresponds to reduced $l_2$ and EMD metrics, indicating enhanced flux suppression. This trend stabilizes upon reaching 20 regions. As for the simplified flux values, we do not observe any significant variation. Furthermore, we analyze the influence of the radius in Fig. 7(b). It reveals an initial reduction in $l_2$ and EMD metrics with radius expansion, suggesting a wider influence on point suppression. This trend, however, inverts at a radius exceeding 0.1, where metrics begin to climb again. Notably, flux magnitude continuously grows with the inclusion of more points. Hence, we adopt 20 local regions with radius of 0.1 in our solution.

**Formal Flux Based on Continuous Field.** To assess the performance impact of our simplified flux approximation, we employ a linear radial basis function to interpolate a continuous field within each local region. This interpolation is based on the Euclidean distances between virtual points within the region and the fixed discretized points. As reported in Tab. 6, the results from attacking PointNet indicate that our simplified version performs comparably to the formal flux based on a continuous field. Moreover, our method incurs significantly lower computational costs than the formal flux. These findings collectively validate the efficiency and effectiveness of our simplified flux.

**Analysis on Undefendability and Transferability.** To evaluate the resilience of our method against different defense mechanisms, we compare it with other adversarial attack techniques targeting PointNet, under two defense strategies: statistical outlier removal (SOR) and DUP-Net [59]. The results, depicted in Fig. 8(a,b), show that all attack methods, FLAT included, experience reduced ASR when countered with these defenses, with DUP-Net causing a more significant decrease. While methods like IFGM and PGD achieve higher ASRs, they

**Table 7:** The average time required by different methods to generate an adversarial example to attack PointNet on ModelNet40.

| Attack | PGD | IFGM | GeoA$^3$ | 3d-Adv | SI-Adv | ITA | Ours |
|---|---|---|---|---|---|---|---|
| Time (s) | 0.440 | 0.322 | 66.093 | 26.815 | 1.059 | 44.529 | 2.020 |

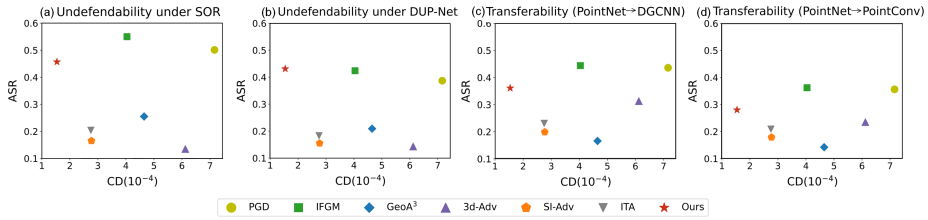


**Fig. 8:** Visualization of (a-b) the undefendability of different methods under the SOR and DUP-Net defense [59], and (c-d) the transferability of different methods from PointNet to DGCNN and from PointNet to PointConv.

do so at the cost of greater distortion. Conversely, our approach results in lower distortion while still achieving comparatively high ASR, showcasing its superior ability to withstand defenses.

Additionally, we assess the transferability of FLAT and other methods by launching attacks on one classifier and testing the generated adversarial point clouds on other classifiers. The outcomes, showcased in Fig. 8(c,d), highlight a notable decline in ASR after transfer, with all methods dropping below 50%. Our method not only shows a competitive ASR but also the least amount of distortion, validating its capability to strike a balance between successful attacks and maintaining imperceptibility.

**Analysis on Time Complexity.** To assess the time efficiency of our method, we report the average minimal time required by different methods to successfully generate an adversarial example to attack PointNet on ModelNet40, as shown in Tab. 7. The results demonstrate that our method's time complexity is competitive compared to existing approaches.

## 6 Conclusion

In this paper, we have proposed FLAT, a novel flux-aware attack framework for generating imperceptible adversarial point clouds. The rationale involves modeling adversarial perturbations as a vector field and subsequently suppressing their flux within localized regions to preserve point cloud uniformity. Extensive experiments validate that FLAT generates adversarial point clouds with enhanced imperceptibility. We hope our work can inspire further research into enhancing the imperceptibility of 3D adversarial attacks. In the future, we plan to delve deeper into field-based factors to continue advancing the imperceptibility.

## Acknowledgements

This work was supported in part by National Natural Science Foundation of China (62102105, U20B2046, 62025207), Guangdong Basic and Applied Basic Research Foundation (2022A1515011501, 2022A1515010138, 2024A1515012064), Guangzhou Sci-Tech Program (SL2022A04J01112), Guangzhou University Program (YJ2023048), and Academician Binxing Fang's Specialized Class.

## References

1. Aris, R.: Vectors, tensors and the basic equations of fluid mechanics. Courier Corporation (2012)
2. Berger, M., Tagliasacchi, A., Seversky, L.M., Alliez, P., Guennebaud, G., Levine, J.A., Sharf, A., Silva, C.T.: A survey of surface reconstruction from point clouds. In: Computer graphics forum. vol. 36, pp. 301–329 (2017)
3. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: going beyond euclidean data. IEEE Signal Processing Magazine **34**(4), 18–42 (2017)
4. Cai, G., He, L., Zhou, M., Alhumade, H., Hu, D.: Learning smooth representation for unsupervised domain adaptation. IEEE TNNLS (2021)
5. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: IEEE Symposium on Security and Privacy. pp. 39–57 (2017)
6. Chen, L., Zhang, Q.: Ddgcn: graph convolution network based on direction and distance for point cloud learning. The Visual Computer **39**(3), 863–873 (2023)
7. Chen, Y., Peng, W., Tang, K., Khan, A., Wei, G., Fang, M.: Pyrapvconv: efficient 3d point cloud perception with pyramid voxel convolution and sharable attention. Computational Intelligence and Neuroscience **2022**(1), 2286818 (2022)
8. Dong, X., Chen, D., Zhou, H., Hua, G., Zhang, W., Yu, N.: Self-robust 3d point recognition via gather-vector guidance. In: CVPR. pp. 11513–11521 (2020)
9. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3d object reconstruction from a single image. In: CVPR. pp. 605–613 (2017)
10. Fang, X., Liu, D., Zhou, P., Hu, Y.: Multi-modal cross-domain alignment network for video moment retrieval. IEEE TMM **25**, 7517–7532 (2022)
11. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: ICLR (2015)
12. Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, M.: Deep learning for 3d point clouds: A survey. IEEE TPAMI **43**(12), 4338–4364 (2020)
13. Han, X.F., Jin, J.S., Wang, M.J., Jiang, W., Gao, L., Xiao, L.: A review of algorithms for filtering the 3d point cloud. Signal Processing: Image Communication **57**, 103–112 (2017)
14. He, Y., Ren, X., Tang, D., Zhang, Y., Xue, X., Fu, Y.: Density-preserving deep point cloud compression. In: CVPR. pp. 2333–2342 (2022)
15. Huang, H., Li, D., Zhang, H., Ascher, U., Cohen-Or, D.: Consolidation of unorganized point clouds for surface reconstruction. ACM TOG **28**(5), 1–7 (2009)
16. Huang, Q., Dong, X., Chen, D., Zhou, H., Zhang, W., Yu, N.: Shape-invariant 3d adversarial point clouds. In: CVPR. pp. 15335–15344 (2022)
17. Ioannidou, A., Chatzilari, E., Nikolopoulos, S., Kompatsiaris, I.: Deep learning advances in computer vision with 3d data: A survey. ACM computing surveys **50**(2), 1–38 (2017)

18. Kim, J., Hua, B.S., Nguyen, T., Yeung, S.K.: Minimal adversarial examples for deep learning on 3d point clouds. In: ICCV. pp. 7797–7806 (2021)
19. Landrieu, L., Simonovsky, M.: Large-scale point cloud semantic segmentation with superpoint graphs. In: CVPR. pp. 4558–4567 (2018)
20. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
21. Lee, K., Chen, Z., Yan, X., Urtasun, R., Yumer, E.: Shapeadv: Generating shape-aware adversarial 3d point clouds. arXiv preprint arXiv:2005.11626 (2020)
22. Li, B.: 3d fully convolutional network for vehicle detection in point cloud. In: IROS. pp. 1513–1518 (2017)
23. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: Pointcnn: Convolution on $\chi$-transformed points. In: NeurIPS. pp. 820–830 (2018)
24. Li, Y., Ma, L., Zhong, Z., Liu, F., Chapman, M.A., Cao, D., Li, J.: Deep learning for lidar point clouds in autonomous driving: A review. IEEE TNNLS **32**(8), 3412–3432 (2020)
25. Liu, D., Hu, W.: Imperceptible transfer attack and defense on 3d point cloud classification. IEEE TPAMI **45**(4), 4727–4746 (2022)
26. Liu, D., Yu, R., Su, H.: Extending adversarial attacks and defenses to deep 3d point cloud classifiers. In: ICIP. pp. 2279–2283 (2019)
27. Luo, S., Hu, W.: Score-based point cloud denoising. In: ICCV. pp. 4583–4592 (2021)
28. Maturana, D., Scherer, S.: Voxnet: A 3d convolutional neural network for real-time object recognition. In: IROS. pp. 922–928 (2015)
29. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: NeurIPS. vol. 32 (2019)
30. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: CVPR. pp. 652–660 (2017)
31. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: deep hierarchical feature learning on point sets in a metric space. In: NeurIPS. pp. 5105–5114 (2017)
32. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover's distance as a metric for image retrieval. IJCV **40**, 99–121 (2000)
33. Shi, W., Rajkumar, R.: Point-gnn: Graph neural network for 3d object detection in a point cloud. In: CVPR. pp. 1711–1719 (2020)
34. Sun, Y., Miao, Y., Chen, J., Pajarola, R.: Pgcnet: patch graph convolutional network for point cloud segmentation of indoor scenes. The Visual Computer **36**(10), 2407–2418 (2020)
35. Taha, A.A., Hanbury, A.: Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool. BMC medical imaging **15**(1), 1–28 (2015)
36. Tang, K., Chen, Y., Peng, W., Zhang, Y., Fang, M., Wang, Z., Song, P.: Reppv-conv: attentively fusing reparameterized voxel features for efficient 3d point cloud perception. The Visual Computer **39**(11), 5577–5588 (2023)
37. Tang, K., He, X., Peng, W., Wu, J., Shi, Y., Liu, D., Zhou, P., Wang, W., Tian, Z.: Manifold constraints for imperceptible adversarial attacks on point clouds. In: AAAI. vol. 38, pp. 5127–5135 (2024)
38. Tang, K., Lou, T., Peng, W., Chen, N., Shi, Y., Wang, W.: Effective single-step adversarial training with energy-based models. IEEE TETCI (2024). https://doi.org/10.1109/TETCI.2024.3378652
39. Tang, K., Ma, Y., Miao, D., Song, P., Gu, Z., Tian, Z., Wang, W.: Decision fusion networks for image classification. IEEE TNNLS (2022). https://doi.org/10.1109/TNNLS.2022.3196129

40. Tang, K., Shi, Y., Lou, T., Peng, W., He, X., Zhu, P., Gu, Z., Tian, Z.: Rethinking perturbation directions for imperceptible adversarial attacks on point clouds. IEEE Internet of Things Journal **10**(6), 5158–5169 (2023)
41. Tang, K., Shi, Y., Wu, J., Peng, W., Khan, A., Zhu, P., Gu, Z.: Normalattack: Curvature-aware shape deformation along normals for imperceptible point cloud attack. Security and Communication Networks **2022**(1), 1186633 (2022)
42. Tang, K., Wu, J., Peng, W., Shi, Y., Song, P., Gu, Z., Tian, Z., Wang, W.: Deep manifold attack on point clouds via parameter plane stretching. In: AAAI. vol. 37, pp. 2420–2428 (2023)
43. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J.: Kpconv: Flexible and deformable convolution for point clouds. In: ICCV. pp. 6411–6420 (2019)
44. Wang, R., Peethambaran, J., Chen, D.: Lidar point clouds to 3-d urban models: A review. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **11**(2), 606–627 (2018)
45. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM Transactions on Graphics **38**(5), 1–12 (2019)
46. Wen, Y., Lin, J., Chen, K., Chen, C.P., Jia, K.: Geometry-aware generation of adversarial point clouds. IEEE TPAMI **44**(6), 2984–2999 (2020)
47. Wicker, M., Kwiatkowska, M.: Robustness of 3d deep learning in an adversarial setting. In: CVPR. pp. 11767–11775 (2019)
48. Wu, W., Qi, Z., Fuxin, L.: Pointconv: Deep convolutional networks on 3d point clouds. In: CVPR. pp. 9621–9630 (2019)
49. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: CVPR. pp. 1912–1920 (2015)
50. Xiang, C., Qi, C.R., Li, B.: Generating 3d adversarial point clouds. In: CVPR. pp. 9136–9144 (2019)
51. Xu, M., Ding, R., Zhao, H., Qi, X.: Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. CVPR (2021)
52. Yang, J., Zhang, Q., Fang, R., Ni, B., Liu, J., Tian, Q.: Adversarial attack and defense on point sets. arXiv preprint arXiv:1902.10899 (2019)
53. Yi, L., Kim, V.G., Ceylan, D., Shen, I.C., Yan, M., Su, H., Lu, C., Huang, Q., Sheffer, A., Guibas, L.: A scalable active framework for region annotation in 3d shape collections. ACM TOG **35**(6), 1–12 (2016)
54. Zhang, J., Jiang, C., Wang, X., Cai, M.: Td-net: Topology destruction network for generating adversarial point cloud. In: ICIP. pp. 3098–3102 (2021)
55. Zhao, H., Jiang, L., Fu, C.W., Jia, J.: Pointweb: Enhancing local neighborhood features for point cloud processing. In: CVPR. pp. 5565–5573 (2019)
56. Zhao, Y., Wu, Y., Chen, C., Lim, A.: On isometry robustness of deep 3d point cloud models under adversarial attacks. In: CVPR. pp. 1201–1210 (2020)
57. Zheng, T., Chen, C., Yuan, J., Li, B., Ren, K.: Pointcloud saliency maps. In: ICCV. pp. 1598–1606 (2019)
58. Zhou, H., Chen, D., Liao, J., Chen, K., Dong, X., Liu, K., Zhang, W., Hua, G., Yu, N.: Lg-gan: Label guided adversarial network for flexible targeted attack of point cloud based deep networks. In: CVPR. pp. 10356–10365 (2020)
59. Zhou, H., Chen, K., Zhang, W., Fang, H., Zhou, W., Yu, N.: Dup-net: Denoiser and upsampler network for 3d adversarial point clouds defense. In: ICCV. pp. 1961–1970 (2019)