Digital Object Identifier

Adversarial Attacks and Defenses on 3D Point Cloud Classification: A Survey

HANIEH NADERI[1]
IVAN V. BAJIĆ[2]
[1]Department of Computer Engineering, Sharif University of Technology Tehran (e-mail: hanieh.naderii@gmail.com)
[2]School of Engineering Science, Simon Fraser University, Burnaby, BC, Canada (e-mail: ibajic@ensc.sfu.ca)

ABSTRACT

Deep learning has successfully solved a wide range of tasks in 2D vision as a dominant AI technique. Recently, deep learning on 3D point clouds is becoming increasingly popular for addressing various tasks in this field. Despite remarkable achievements, deep learning algorithms are vulnerable to adversarial attacks. These attacks are imperceptible to the human eye but can easily fool deep neural networks in the testing and deployment stage. To encourage future research, this survey summarizes the current progress on adversarial attack and defense techniques on point cloud classification. This paper first introduces the principles and characteristics of adversarial attacks and summarizes and analyzes the adversarial example generation methods in recent years. Besides, it classifies defense strategies as input transformation, data optimization, and deep model modification. Finally, it presents several challenging issues and future research directions in this domain.

INDEX TERMS 3D deep learning, deep neural network, adversarial examples, adversarial defense, machine learning security, 3D point clouds.

I. INTRODUCTION

DEEP learning (DL) [1] is a subset of machine learning (ML) and artificial intelligence (AI) that analyzes large amounts of data using a structure roughly similar to the human brain. Deep learning is characterized by the use of multiple layers of neural networks, which process and analyze large amounts of data. These neural networks are trained on large datasets, which allows them to learn patterns and make decisions on their own. DL has achieved impressive results in the fields of image recognition [2, 3], semantic analysis [4, 5], speech recognition [6, 7] and natural language processing [8] in recent years.

Despite the tremendous success of DL, in 2013 Szegedy *et al.* [9] found that deep models are vulnerable to adversarial examples in image classification tasks. Adversarial examples are inputs to a deep learning model that have been modified in a way that is intended to mislead the model. In the context of image classification, for example, an adversarial example might be a picture of a panda that has been slightly modified in a way that is imperceptible to the human eye but that causes a deep learning model to classify the image as a gibbon. Adversarial examples can be created in two or three dimensions. In the case of 2D adversarial examples, the input is an image, and the modification is applied to the pixels of the image. These modifications can be small perturbations added to the image pixels [10, 11, 12, 13, 14, 15, 16] or they can be more significant changes to the structure of the image [17, 18, 19, 20].

Thanks to the rapid development of 3D acquisition technologies, various types of 3D scanners, LiDARs, and RGB-D cameras have become increasingly affordable. 3D data is often used as an input for Deep Neural Networks (DNNs) in healthcare [21], self-driving cars [22], drones [23], robotics [24], and many other applications. These 3D data, compared to 2D counterparts, capture more information from the environment, thereby allowing more sophisticated analysis. There are different representations of 3D data, like voxels [25], meshes [26], and point clouds [27]. Since point clouds can be received directly from scanners, they can precisely capture shape details. Therefore, it is the preferred representation for many safety-critical applications. Due to this, in the case of 3D adversarial examples, the input is a point cloud, and the modification is applied to the points in the cloud. These examples can be created by adding, dropping, and shifting some points in the input point clouds, or by generating entirely new point clouds with predefined target labels using methods such as Generative Adversarial Networks (GANs) or other transformation techniques. It is

typically easier to create adversarial examples in 2D space than in 3D space because the input space is smaller and there are fewer dimensions to perturb. In general, adversarial examples exploit the vulnerabilities or weaknesses in the model's prediction process, and they can be very difficult to detect because they are often indistinguishable from normal examples to the human eye. As a result, adversarial examples can pose a serious threat to the security and reliability of DL models. Therefore, it is important to have effective methods for defending against adversarial examples in order to ensure the robustness and reliability of DL models.

Adversarial defense in the 2D image and the 3D point clouds both seek to protect DL models from being fooled by adversarial examples. However, there are some key differences between the approaches used to defend against adversarial images and adversarial point clouds. Some of the main differences include the following:

- Input data: Adversarial images are 2D data representations, while adversarial point clouds are 3D data representations. This means that the approaches used to defend against adversarial images and point clouds may need to take into account the different dimensions and characteristics of the input data.
- Adversarial perturbations: Adversarial images may be modified using small perturbations added to the image pixels, while adversarial point clouds may be modified using perturbations applied to individual points or groups of points in the point cloud. This means that the approaches used to defend against adversarial images and point clouds may need to be tailored to the specific types of adversarial perturbations that are being used.
- Complexity: Adversarial point clouds may be more complex to defend against than adversarial images, as the perturbations applied to point clouds may be more difficult to identify and remove. This may require the use of more sophisticated defenses, such as methods that are able to detect and remove adversarial perturbations from the input point cloud.

On the whole, adversarial point clouds can be challenging to identify and defend against, as they may not be easily recognizable in the 3D point cloud data. Adversarial point clouds may be more harmful and harder to defend against, because their changes may be less obvious to humans due to the lack of familiarity compared to images. As a result, it is important to conduct a thorough survey of adversarial attacks and defenses on 3D point clouds in order to identify the challenges and limitations of current approaches and to identify opportunities for future research in this area. There are a number of published surveys that review adversarial attacks and defenses in general, including in the context of computer vision, machine learning, and deep learning systems. These surveys provide an overview of the various types of attacks and defenses that have been proposed, as well as their strengths and limitations. However, there is a lack of surveys specifically focused on 3D point cloud attacks and defenses. Some published surveys do mention 3D attacks and defenses briefly [28], but there is a need for more comprehensive surveys that delve deeper into this topic. Table 1 refers to a summary or overview of published surveys of adversarial attacks and defenses. Some of these surveys focus on specific domains, such as computer vision [28, 29, 30], text [31], and images [32, 33, 34, 35] while others provide a more general overview of adversarial attacks and defenses in the field of artificial intelligence [36, 37].

Our key contributions are as follows:

- A review of the different types of adversarial point clouds that have been proposed and the methods that have been used to generate them, and proposing a taxonomy of these methods.
- A review of the various methods that have been proposed for defending against adversarial point clouds, including data optimization, input transformation methods, and deep model modification.
- Categorization of the most important datasets and models used by researchers in this field.
- An assessment of the challenges and limitations of current approaches to adversarial attacks and defenses on 3D point clouds, and identification of opportunities for future research in this area.

An overview of the categorization of adversarial attack and defense approaches on 3D point clouds is shown in Fig. 1. The rest of this paper is organized as follows. Section II introduces a list of notations, terms and measurements used in the paper. We discuss adversarial attacks on deep models for 3D point cloud classification in Section III. Section IV provides a detailed review of the existing adversarial defense methods. In Section V, we summarize commonly used 3D datasets and present a taxonomy of datasets and victim models used in recent studies. We discuss current challenges and potential solutions related to adversarial attacks in Section VI. Finally, Section VII concludes the survey.

II. BACKGROUND

In this section, we provide the necessary background in terms of notation, terminology, and point cloud distance measures used in the field of 3D adversarial attacks. By establishing clear definitions, researchers can more accurately compare the effectiveness of different approaches and identify trends or patterns in the methods.

A list of symbols used in the paper is given in Table 2, along with their explanations. These symbols are used to represent various quantities related to point cloud adversarial attacks. The table provides a brief description of each symbol to help readers understand and follow the discussions and equations in the paper. Next, we briefly introduce the terminology and distance measures used in the field of adversarial attacks and defenses on 3D point clouds.

A. DEFINITION OF TERMS

It is crucial to define the technical terms used in the literature in order to provide a consistent discussion of the various

TABLE 1: A review of published surveys of adversarial attacks and defenses .

Surveys	Year
Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey [29]	2018
Adversarial Examples: Attacks and Defenses for Deep Learning [33]	2018
Review of artificial intelligence adversarial attack and defense technologies [36]	2019
Adversarial Examples in Modern Machine Learning: A Review [38]	2019
A survey on adversarial attacks and defences [34]	2021
Advances in Adversarial Attacks and Defenses in Computer Vision: A Survey [28]	2021
A survey on the vulnerability of deep neural networks against adversarial attacks [35]	2022
Adversarial Attack and Defense Strategies of Speaker Recognition Systems: A Survey [39]	2022
Adversarial attack and defense technologies in natural language processing: A survey [31]	2022
Adversarial Attack and Defense: A Survey [40]	2022
A Review of Adversarial Attack and Defense for Classification [41]	2022
Adversarial Attacks and Defenses for Deployed AI Models [37]	2022
Physically Adversarial Attacks and Defenses in Computer Vision: A Survey [42]	2022
Physical Adversarial Attack meets Computer Vision: A Decade Survey [30]	2022
Adversarial Examples based on Object Detection tasks: A Survey [43]	2022
Adversarial Deep Learning: A Survey on Adversarial Attacks and Defense Mechanisms on Image Classification [32]	2022

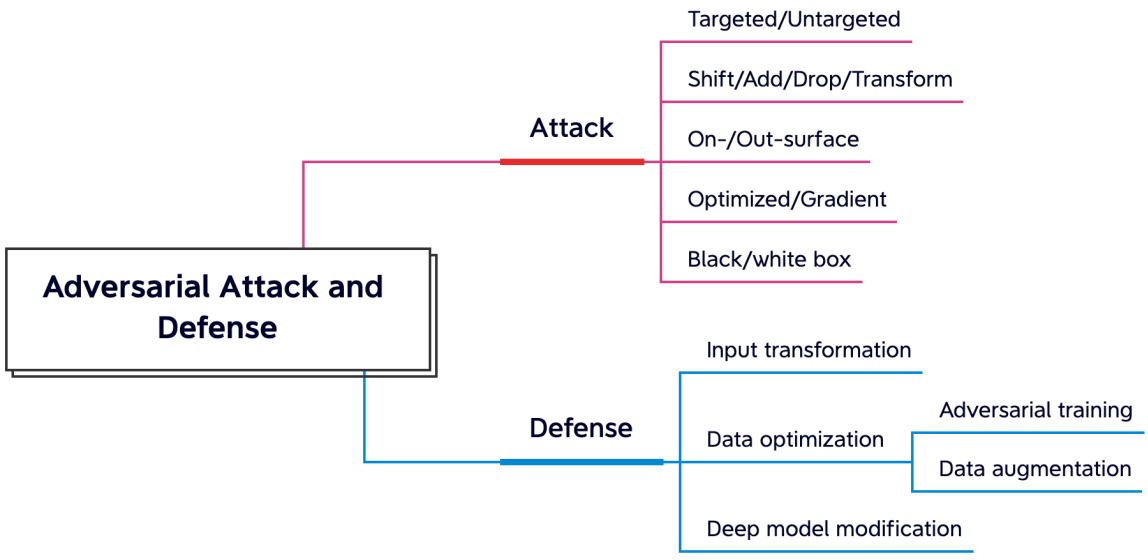

FIGURE 1: Categorization of adversarial attack and defense approaches on 3D point clouds.

methods and approaches. The definitions of these terms appear below. The rest of the paper follows the same definitions throughout.

- **3D point cloud** is a set of points in 3D space, typically representing a 3D shape or scene.
- **Adversarial point cloud** is a 3D point cloud that has been intentionally modified in order to mislead a DL model that analyzes 3D point clouds. We focus on geometric modifications, rather than attribute (e.g., color) modifications, since these are predominant in the literature on adversarial point clouds.
- **Adversarial attack** is a technique that intentionally introduces perturbations or noise to an input point cloud in order to fool a DL model, causing it to make incorrect predictions or decisions.
- **Black-box attacks** are a type of adversarial attack in which the attacker only has access to the model's input and output, and has no access to the structure of the DL model being attacked.
- **White-box attacks** are a type of adversarial attack in which the attacker knows all the details about the DL model's architecture and parameters.
- **Targeted attacks** involve manipulating the input point cloud in a way that causes the model to output a specific target label when presented with the modified input.
- **Non-targeted attacks** involve manipulating the input point cloud in a way that causes the model to output a wrong label, regardless of what that label is.
- **Point addition attacks** involve adding points to the point cloud to fool the DL model.

TABLE 2: Symbols and their explanations.

Symbol	Description
$\mathcal{P}$	An instance of an original (input) point cloud
$\mathcal{P}^{adv}$	An instance of an adversarial point cloud
p_i	i-th point in the original (input) point cloud
p_i^{adv}	i-th point in the adversarial point cloud
η	Perturbation vector (difference between the original and adversarial point cloud)
ϵ	Perturbation threshold
α	Scale parameter
n	Total number of points in a point cloud
Y	ground-truth label associated with original input
Y'	Wrong label associated with an adversarial example that deep model predicts
T	Target attack label
$f(\cdot)$	Mapping from the input point cloud to the output label implemented by the deep model
θ	Parameters of model f
$J(\cdot,\cdot)$	Loss function used for model f
∇	Gradient
sign$(\cdot)$	Sign function
P	Parameter of the ℓ_P-norm; typical values of P are $1, 2$ and ∞.
D_{ℓ_P}	ℓ_P-norm distance
D_H	Hausdorff distance
D_C	Chamfer distance
k	Number of nearest neighbors of a point
κ	Confidence constant
z	Latent space of a point autoencoder
$g(\cdot)$	Objective function
t	Number of iterations
μ	Mean of k nearest neighbor distance of all points in a point cloud
σ	Standard deviation of k nearest neighbor distance of all points in a point cloud

- **Point shift attacks** involve shifting points of the point cloud to fool the DL model, while the number of points remains the same as in the original point cloud.
- **Point drop attacks** involve dropping points from the point cloud to fool the DL model.
- **Optimization-based attacks** are a type of attack in which the creation of an adversarial point cloud is formulated and solved as an optimization problem.
- **Gradient-based attacks** are a type of attack in which the gradients of the cost function corresponding to each input point are used to generate an adversarial point cloud with higher tendency toward being misclassified.
- **On-surface perturbation attacks** are a type of attack that involves modifying points along the object's surface in the point cloud.
- **Out-of-surface perturbation attacks** are a type of attack that involves modifying points outside the object surface in the point cloud.
- **Transferability** refers to the ability of adversarial examples generated for one DL model to be successful in causing misclassification for another DL model.
- **Adversarial defense** is a set of techniques that aim to mitigate the impact of adversarial attacks and improve the robustness of the DL model against them.
- **Attack success rate** refers to the percentage of times that an adversarial attack on a DL model is successful.

B. DISTANCE MEASURES

The objective of adversarial attacks is to modify points of $\mathcal{P}$, creating an adversarial point cloud $\mathcal{P}^{adv}$, which could fool a DL model to output wrong results. Geometric 3D adversarial attacks can be achieved by adding, dropping, or shifting points in $\mathcal{P}$. If the adversarial point cloud is generated by shifting points, $\boldsymbol{\ell_P}$**-norms** can be used to measure the distance between $\mathcal{P}$ and $\mathcal{P}^{adv}$, as the two point clouds have the same number of points. In this case, we can talk about the vector difference (perturbation) $\eta = \mathcal{P} - \mathcal{P}^{adv}$, and consider $\|\eta\|_P$ as the distance between $\mathcal{P}$ and $\mathcal{P}^{adv}$. The typical choices for P are $P \in \{0, 2, \infty\}$, and the equation is:

$$D_{\ell_P}(\mathcal{P}, \mathcal{P}^{adv}) = \|\eta\|_P = \left(\sum_{i=1}^{n} \|p_i - p_i^{adv}\|_P^P\right)^{1/P} \quad (1)$$

where $\mathcal{P} \in \mathbb{R}^{n\times 3}$ is the original point cloud consisting of n points in 3D space, $\mathcal{P} = \{p_i \,|\, i = 1, 2, ..., n\}$ and the i^{th} point, $p_i = (x_i, y_i, z_i)$, is a 3D vector of coordinates. $\mathcal{P}^{adv}$ is the adversarial point cloud formed by adding the adversarial perturbation $\eta = (\eta_1, \eta_2, ..., \eta_n), \eta_i \in \mathbb{R}^3$, to $\mathcal{P}$. The three common ℓ_P norms have the following interpretations:

- $\boldsymbol{\ell_0}$**-norm** or $\|\eta\|_0$ counts the number of non-zero elements in η, so it indicates how many points in $\mathcal{P}^{adv}$ have changed compared to $\mathcal{P}$.
- $\boldsymbol{\ell_2}$**-norm** or $\|\eta\|_2$ is the Euclidean distance between $\mathcal{P}^{adv}$ and $\mathcal{P}$.
- $\boldsymbol{\ell_\infty}$**-norm** or $\|\eta\|_\infty$ is the maximum difference between the points in $\mathcal{P}^{adv}$ and $\mathcal{P}$.

As mentioned above, ℓ_P-norm distance criteria require that $\mathcal{P}^{adv}$ and $\mathcal{P}$ have the same number of points. Hence, these distance measures cannot be used for attacks that involve adding or dropping points. To quantify the dis-similarity between two point clouds that don't have the same number of

points, **Hausdorff distance** D_H and **Chamfer distance** D_C are commonly used. Hausdorff distance is defined as follows:

$$D_H(\mathcal{P}, \mathcal{P}^{adv}) = \max_{p \in \mathcal{P}} \min_{p^{adv} \in \mathcal{P}^{adv}} \|p - p^{adv}\|_2^2 \quad (2)$$

It locates the nearest original point p for each adversarial point p^{adv} and then finds the maximum squared Euclidean distance between all such nearest point pairs. Chamfer distance is similar to Hausdorff distance, except that it sums the distances among all pairs of closest points, instead of taking the maximum:

$$\begin{aligned} D_C(\mathcal{P}, \mathcal{P}^{adv}) = & \sum_{p^{adv} \in \mathcal{P}^{adv}} \min_{p \in \mathcal{P}} \|p - p^{adv}\|_2^2 \\ & + \sum_{p \in \mathcal{P}} \min_{p^{adv} \in \mathcal{P}^{adv}} \|p - p^{adv}\|_2^2 \end{aligned} \quad (3)$$

Optionally, Chamfer distance can be averaged with respect to the number of points in the two point clouds.

Besides the distance measures mentioned above, there are other distance measures for point clouds, such as point-to-plane distance [44], that are used in point cloud compression. However, these are not commonly encountered in the literature on 3D adversarial attacks, so we don't review them here.

III. ADVERSARIAL ATTACKS

This section describes the seven most common approaches for generating adversarial point clouds. Our discussion encompasses the technicalities of these seven widely used methods and also briefly touches upon similar approaches related to these seven attacks. Some of the approaches [45, 46] described in this section are extended versions of adversarial examples for 2D data, adapted for use with 3D point clouds. These approaches may face new challenges due to the additional dimension of the data. Other approaches [47] are specifically designed for 3D data and may be more effective at generating adversarial point clouds than methods that are simply adapted from 2D data. These approaches may consider the unique characteristics of 3D point clouds and the deep models that process them. Overall, the goal of these approaches is to understand better how adversarial point clouds could affect current deep 3D models. The most popular approaches are also summarized in Table 3 and we explain how adversarial attacks and attack categories relate in the context of adversarial examples for point cloud classification tasks.

A. 3D FAST GRADIENT SIGN METHOD (3D FGSM)

The fast gradient sign method (FGSM) presented by Goodfellow *et al.* [61]. In accordance with standard FGSM, the method adds an adversarial perturbation η to each point of given point cloud $\mathcal{P}$ in order to create an adversarial point cloud as $\mathcal{P}^{adv} = \mathcal{P} + \eta$. Perturbations are generated according to the direction of the sign of gradient at each point. The perturbation can be expressed as

$$\eta = \epsilon sign(\nabla_{\mathcal{P}} J(f(\mathcal{P} : \theta), Y) \quad (4)$$

where f is deep model that is parameterized by θ and takes an input point cloud $\mathcal{P}$ and Y denotes the label associated with $\mathcal{P}$. $\Delta_x J(.,.)$ is gradient of loss function of model w.r.t to $\mathcal{P}$ and $sign(.)$ denotes the sign function. The ϵ value is an adjusting hyperparameter that determines the ℓ_∞-norm of the difference between the original and adversarial inputs.

The FGSM was extended by Liu *et al.* [54] to 3D data. There are three different ways were introduced [54] to define ϵ value as a constraint for η as follows

1) Constraining the ℓ_2-norm between each dimension of points $\mathcal{P}$ and $\mathcal{P}^{adv}$..
2) Constraining the ℓ_2-norm between each point $\mathcal{P}$ and $\mathcal{P}^{adv}$.
3) Constraining the ℓ_2-norm between all points $\mathcal{P}$ and $\mathcal{P}^{adv}$.

Due to the first method severely limiting the movement of points, the authors suggest the second and third methods. However, all three methods have shown little difference in the attack success rates.

Yang *et al.* [45] used the Chamfer distance (instead of ℓ_2-norm) between the original point cloud and the adversarial counterpart to extend FGSM to a 3D domain. Using this approach, each point in the adversarial point clouds is perturbed slightly. There is a trade-off between the chamfer distance and the attack success rate because, as the chamfer distance decreases, it may become more difficult for an adversarial attack to achieve a high attack success rate. However, if the chamfer distance is set too high, the model may be more vulnerable to adversarial attacks. Finding the right balance between these two factors can be challenging, and it may depend on the specific characteristics of the point cloud model and the type of adversarial attack being used. Figure 2 illustrates an example of an FGSM adversarial point cloud with Chamfer distances varying from 0.01 to 0.05 between the two point clouds. The author in [45] sets it to 0.02 as an "appreciate distance".

Apart from the FGSM attack, Yang *et al.* [45] introduced another attack called "Momentum-Enhanced Pointwise Gradient (**MPG**)." The MPG attack, similar to [62], integrates momentum into iterative FGSM. The MPG attack produces more transferable adversarial examples.

B. 3D CARLINI AND WAGNER ATTACK (3D C&W)

The C&W attack is presented by Carlini and Wagner [63]. They provided three kinds of attacks with three different distance measures, ℓ_0-norm, ℓ_2-norm, and ℓ_∞-norm. As a general rule, generating the C&W attack can describe as an optimization problem to find minimum perturbation η such that the label of the adversarial input $\mathcal{P}^{adv}$ is changed to the target label T by the objective function g.

$$\begin{aligned} \min_{\eta} \quad & D(\mathcal{P}, \mathcal{P}^{adv}) + c.g(\mathcal{P} + \eta) \\ s.t. \quad & f(\mathcal{P}^{adv}) = T \end{aligned} \quad (5)$$

where $D(.)$ refers to distance measure (it can be defined using different distance measures like ℓ_P-norm, Chamfer

TABLE 3: Relationship between adversarial attacks and attack categories.

Ref	Attack Name	Categories				
		Targeted/Non-targeted	Shift/Add/Drop/Transform	On-/Out-surface	Optimized/Gradient	Black-/White-box
[46]	Perturbation	Targeted	Shift	Out	Optimized	White
	Independent points	Targeted	Add	Out	Optimized	White
	Clusters	Targeted	Add	Out	Optimized	White
	Objects	Targeted	Add	Out	Optimized	White
[48]	Drop100	Non-Targeted	Drop	On	Gradient	White
	Drop200	Non-Targeted	Drop	On	Gradient	White
[49]	Advpc	Targeted	Transform	On	Optimized	White
[50]	ShapeAdv	Targeted	Shift	On	Optimized	White
[51]	LG-GAN	Targeted	Transform	On	-	White
[52]	$GeoA^3$	Targeted	Shift	On	Optimized	White
[53]	KNN	Targeted	Shift	On	Optimized	White
[54]	Extended FGSM	Non-Targeted	Shift	Out	Gradient	White
[55]	VSA	Non-Targeted	Add	On	Optimized	White
[56]	Distributional attack	Non-Targeted	Shift	On	Gradient	White
	Perturbation resampling	Non-Targeted	Add	Out	Gradient	White
	Adversarial sticks	Non-Targeted	Add	Out	Gradient	White
	Adversarial sinks	Non-Targeted	Add	Out	Gradient	White
[57]	Minimal	Non-Targeted	Shift	Out	Optimized	White
	Minimal	Non-Targeted	Add	Out	Optimized	White
[58]	JGBA	Targeted	Shift	On	Optimized	White
[59]	ITA	Targeted	Shift	On	Optimized	Black
[45]	FGSM	Non-Targeted	Shift	Out	Gradient	White
	MPG	Non-Targeted	Shift	Out	Gradient	White
	Point-attachment	Non-Targeted	Add	Out	Gradient	White
	Point-detachment	Non-Targeted	Drop	On	Gradient	White
[60]	Wicker *et al.*	Both	Drop	On	Optimized	Both

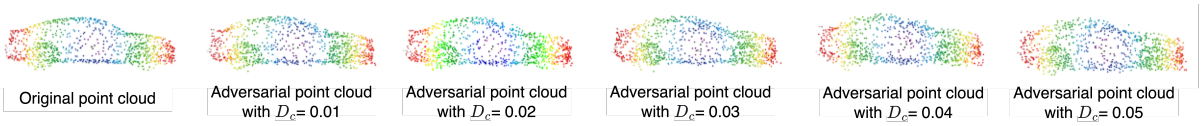

FIGURE 2: An example of original point cloud and 3D FGSM adversarial counterpart [45] with Chamfer distances varying from 0.01 to 0.05.

or Hausdorff distance), c is a suitably chosen constant and $g(\mathcal{P}^{adv}) \geq 0$ if and only if $f(\mathcal{P}^{adv}) = T$. By doing so, the distance and penalty term can be optimized more effectively. There were seven objective functions g listed by the authors [63]. An effective function evaluated by their experiments, which was also used in other papers, is as follows

$$g(\mathcal{P}^{adv}) = max(\max_{i=t}(Z(\mathcal{P}^{adv})_i) - Z(\mathcal{P}^{adv})_t, -\kappa) \quad (6)$$

where Z denotes the Softmax function, and κ represents a constant that controls confidence. In comparison with the FGSM attack, these attacks do not set a constraint for perturbation. In fact, the attacks search for minimal perturbation (without imposing any constraints) to change the label to the target label.

As the first instance, a 3D version of the C&W attack was developed by Xiang *et al.* [46]. According to the paper, [46], four types of attacks were proposed as follows. In Figure 3, you can see the four types of C&W attacks, where the bottle label has been misclassified as a result of these attacks.

1) **Adversarial perturbation** negligibly by using ℓ_2-norm (between all points $\mathcal{P}$ and $\mathcal{P}^{adv}$) as distance measure to shift points toward the point cloud's surface.
2) **Adding adversarial independent points** by using two different distance measures. 1. Chamfer distance between the original point cloud and the adversarial point cloud. 2. Hausdorff distance between the original point cloud and the adversarial point cloud.
   These measures are used to push independent points toward the point cloud's surface.
3) **Adding adversarial clusters** by the combination of three different distance measures. 1. Chamfer distance between the original point cloud and the adversarial cluster is used to push clusters toward the point cloud's surface. 2. The number of clusters added. Using this measure, only 1 to 3 clusters are added, so there is only a small number of clusters added. 3. Minimize the farthest distance. In this measure, the distance between the two most distant points in each cluster is minimized to constrain the added points clustered to be within small regions.
4) **Adding adversarial objects** by the combination of three different distance measures. 1. Chamfer distance between the original point cloud and the adversarial object is used to push adversarial objects toward the point cloud's surface. 2. The number of objects added. Using this measure, only 1 to 3 objects are added, so

there is only a small number of objects added. 3. ℓ_2-norm between a real-world object and an adversarial object is used to generate shapes similar to the real-world ones.

The first attack is based on shifting points, and three other attacks are based on adding points. Since directly adding points to the unbounded 3D space is not possible due to the vast search space, the last three attacks use the position of critical points as the initial positions of adversarial points (or clusters or objects). Critical points are like key points that are effective in classification results. An example of critical points in PointNet would be calculating the remaining points after max pooling.

Tsai *et al.* [53] developed a shifting point attack called K-Nearest Neighbor (**KNN**) attack that limits distances between adjacent points by adding an extra distance loss to 5, which calculates K-Nearest Neighbor distance for each point. By doing so, adversarial point clouds are restricted to becoming physical objects. They use Chamfer distance to measure the distance of two point clouds.

Wen *et al.* [52] considered a new distance measure named consistency of local curvatures to guide perturbed points lean towards object surfaces. Adopting the C&W attack framework, the authors use the combination of Chamfer distance, Hausdorff distance, and local curvature consistency distance as the distance measure to create a geometry-aware adversarial attack ($GeoA^3$). The generated $GeoA^3$ attack has smoothness and fairness surface properties, so the difference between it and the original point cloud is imperceptible to the human eye.

C. 3D PROJECTED GRADIENT DECENT METHOD (3D PGD)

One of the most potent attacks in the 2D literature is the Projected Gradient Descent (PGD), which has its roots in the pioneering paper of Madry *et al.* [64]. The iterative FGSM is considered a PGD method. Taking the iterative FGSM method, we can generate the adversarial point cloud as

$$\mathcal{P}_0^{adv} = x, \quad \mathcal{P}_{t+1}^{adv} = Clip_{\mathcal{P},\epsilon}[\mathcal{P}_t^{adv} + \alpha sign(\nabla_{\mathcal{P}} J(f(\mathcal{P}:\theta), Y)] \tag{7}$$

where $Clip_{\mathcal{P},\epsilon}$ limits the change of the generated adversarial input in each iteration and t refers to iteration.

The PGD attack try to increase the cost of the correct class Y, without specifying which of the incorrect classes the model should select. The PGD attack finds the perturbation that maximizes the cost function under the η constraint with ϵ.

$$\begin{aligned} &\max_{\eta} \quad J(f(\mathcal{P}:\theta), Y) \\ &s.t. \quad D(\mathcal{P}, \mathcal{P}^{adv}) \leq \epsilon - ball \end{aligned} \tag{8}$$

The 3D PGD attack is similar to the 2D version, but it usually uses different distance measures to calculate perturbations. In particular, Liu *et al.* [56] proposed a PGD attack named **Distributional attack** by using the Hausdorff distance between the triangular mesh (original point cloud surface approximate through a triangular mesh) and the adversarial point cloud as distance measure to push adversarial points toward the triangular mesh. This method is less sensitive to the density of points in $\mathcal{P}$ because it uses a mesh instead of a point cloud to measure perturbation. Figure 4 demonstrated two examples of adversarial point clouds generated by the distributional attack.

Ma *et al.* [58] proposed Joint Gradient Based Attack (**JGBA**) attack. They added an extra term to the optimization function of the PGD attack 8 to defeat the SOR (Statistical Outlier Remover), which removes outlier points. The term computes the gradient of the loss function of model w.r.t to points in $\mathcal{P}$ after removing outliers when the first term (term in 8) computes the gradient of the loss function of model w.r.t to all points in $\mathcal{P}$. These two terms are combined to solve the optimization problem. The JGBA attack takes ℓ_2-norm as the distance measure to constraint shifting of points.

D. SHAPE ATTACK

This type of attack attempts to morph the point cloud's shape. The concept of shape attacks can be compared to what is called unrestricted attacks in 2D images [65, 66, 67]. When such attacks occur, input data might change significantly while not changing the semantics. This adversarial attacks fool the classifier without making humans confused. In this regard, Liu *et al.* [56] proposed three shape attacks as follows. Figure 5 demonstrates these three shape attacks.

1) **Perturbation resampling** This attack resamples the certain number of points with the lowest gradients by farthest point sampling to ensure that all points are distributed approximately uniformly. The algorithm is iterated to generate an adversarial point cloud that deceives the model. The distance measure used to maintain the similarity between $\mathcal{P}$ and $\mathcal{P}^{adv}$ is Hausdorff distance.
2) **Adding adversarial sticks** During this attack, the algorithm adds four sticks to the point cloud so that one end of them is attached to the point cloud and the other end has a very small distance from the first end. The algorithm optimizes the two ends of the sticks so that the label of the point cloud be changed. Finally, it adds a few points between the two ends to make them look like sticks.
3) **Adding adversarial sinks** In this case, critical points (remaining points after max pooling in PointNet) selects as sink points, and points pull in the point cloud toward them. The goal of this attack is to minimize global changes to points that are not selected by the max pooling operation. The distance measure used to maintain the similarity between $\mathcal{P}$ and $\mathcal{P}^{adv}$ is ℓ_2-norm.

Lee *et al.* [50] also proposed Shape-aware adversarial attacks called **ShapeAdv** that are based on injecting an

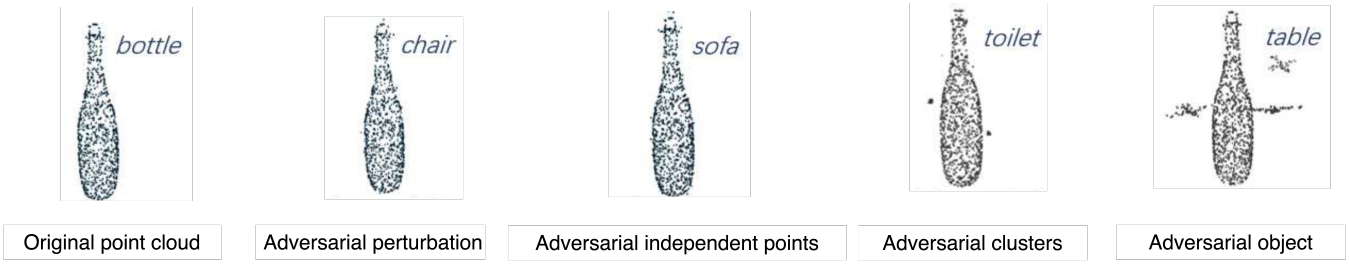

FIGURE 3: An example of original point cloud and four types of 3D C&W adversarial counterpart were proposed in [46].

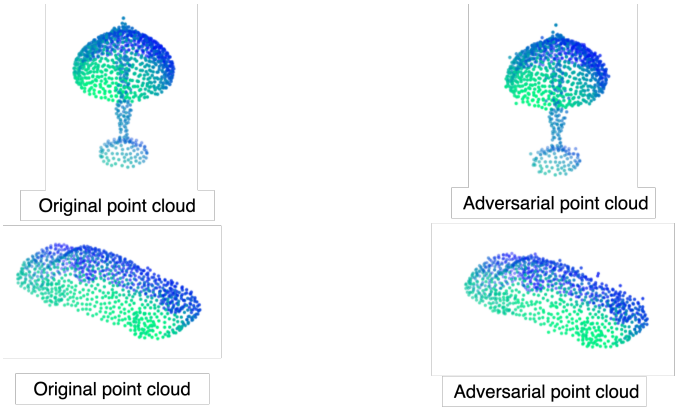

FIGURE 4: two example of original point clouds and distributional attacks (3D PGD adversarial counterparts) were proposed in [56].

adversarial perturbation η in the latent space z of a point cloud autoencoder. To be precise, the original point cloud is processed using an autoencoder to generate an adversarial point cloud, then the adversarial point cloud is fed to the classifier. Accordingly, Lee *et al.* [50] generated three attacks with varying distance measures. These measures are used as a term for C&W loss to maintain similarity between the original and the adversarial point clouds. All three attacks calculate gradient C&W loss w.r.t adversarial perturbation in the latent space z. The distance measures are defined as such for three types of attacks:

1) **Shape-aware attack in the latent space.** To make a more meaningful attack, the author minimizes the ℓ_2-norm between the latent space z and the adversarial latent space $z + \eta$. Using this approach, the generated adversarial point cloud is highly dissimilar from the original counterpart in terms of appearance.
2) **Shape-aware attack in the point space.** In this case, an attempt is being made to resolve the previous attack's problem. In order to maintain similarity between the original point cloud and the adversarial one, the distance measure is replaced by minimizing the Chamfer distance between the two.
3) **Shape-aware attack with auxiliary point clouds.** The attack minimizes the Chamfer distance between the adversarial point cloud and the average of k nearest neighbor, sampled from the original point cloud category. This attack aims to avoid adversarial perturbation in any direction in the latent space. To guide the direction in the latent space, it employs auxiliary point clouds sampled from the category of the original input.

1) Shape attacks via autoencoders and generative models

Hamdi *et al.* [49] proposed an attack called **Advpc** by using an autoencoder that could be transferred between networks effectively. This was achieved by introducing a new loss function and pipeline. Minimizing two losses was the goal of the Loss function. The first loss is C&W loss when adversarial point clouds are fed into deep models, and the second loss is C&W loss when adversarial point clouds are fed into deep models after reconstruction with a point cloud autoencoder. Using an autoencoder to generate an adversarial point cloud makes perturbations more meaningful. Consequently, their transferability from one network to another will be more promising. Lee *et al.* [50] also proposed Shape-aware attacks by injecting adversarial perturbation η in the latent space z of a point cloud autoencoder. In section III-D, this attack was described in detail. **LG-GAN** attack [51] is proposed to generate an adversarial point cloud based on GAN (Generative Adversarial Network). The GAN is fed with the original point clouds and target labels to learn how to generate adversarial point clouds to fool deep models. In detail, it extracts hierarchical features from original point clouds using one multi-branch adversarial network, then integrates the specified label information into multiple intermediate features using the label encoder. The encoded features will be fed into a reconstruction decoder to generate the adversarial point cloud. This attack is so fast because it only takes one forward pass to generate an adversarial point cloud. Figure 6 shows an instance of the LG-GAN attack.

Dai*et al.* [68] proposed a new type of attack based on GAN, which is created from noise rather than the original point cloud. In fact, the noise vector and the target label as the input are fed into a graph convolutional generator. It outputs the generated adversarial point cloud. The generator uses a loss function containing four parts (the objective loss, the discriminative loss, the outlier loss, and the uniform loss) to achieve a realistic adversarial attack that fools the victim network. The objective loss encourages the victim network to assign the target(incorrect) label to the adversarial point

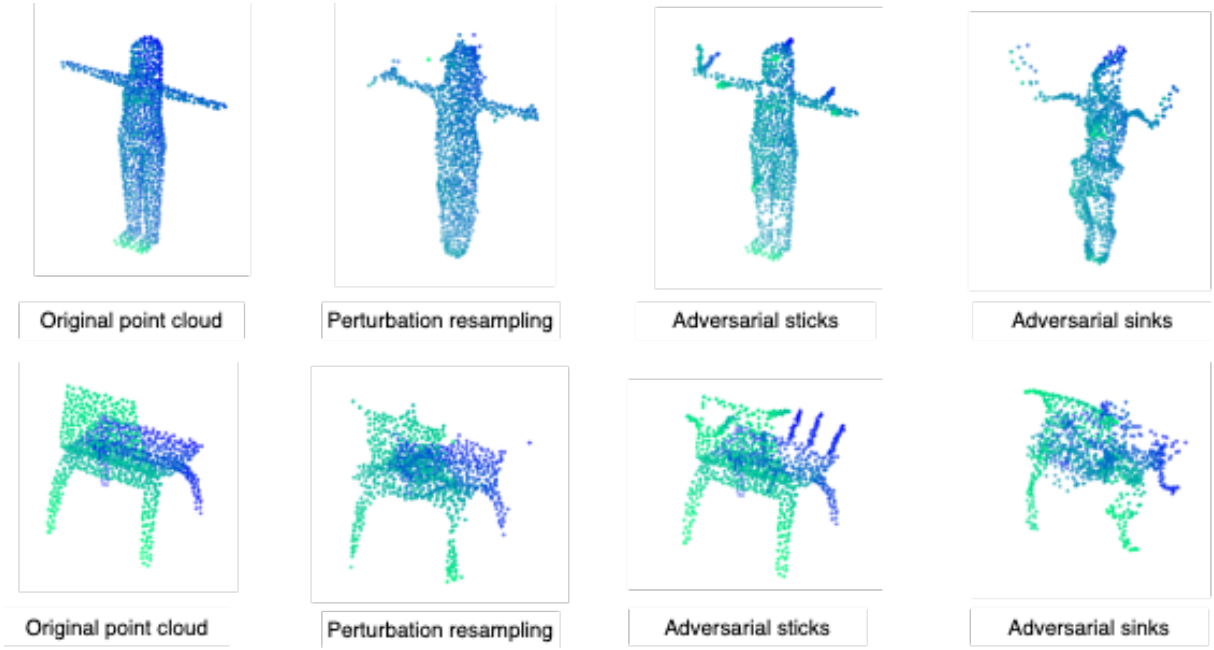

FIGURE 5: Two examples of original point clouds and three shape attacks were proposed in [56].

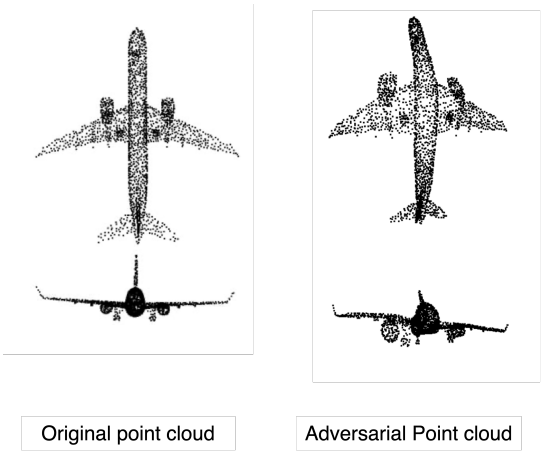

FIGURE 6: An example of original point cloud and LG-GAN attack were proposed in [51].

cloud while the discriminative loss encourages the auxiliary network to classify the adversarial point cloud correctly. The outlier loss and the uniform loss by removing outliers and generating a more uniform point cloud force the generator to preserve the point cloud shape.

Lang*et al.* [69] proposed a new type of adversarial attack that alters the reconstructed geometry of a 3D point cloud rather than just the predicted label, using an autoencoder trained on semantic shape classes.

Mariani*et al.* [70] proposed a method for creating adversarial attacks on surfaces embedded in 3D space, under weak smoothness assumptions on the perceptibility of the attack.

E. FREQUENCY ATTACK (ATTACK ON OTHER DOMAINS)

Liu *et al.* [71] have suggested an adversarial attack based on the frequency domain, which aims to enhance the transferability of generated adversarial examples. The author transformed points onto the frequency domain via graph Fourier transform (GFT). Then divide it into low-frequency components and high-frequency components, and apply perturbations to the low-frequency components to create an adversarial point cloud. In a contrasting way, Liu *et al.* [72] investigated the geometric structure of 3D point clouds by perturbing each of the three frequency components (low, mid, and high-frequency). They found that perturbing low-frequency components of point clouds significantly damaged their rough shape. To preserve the shape of the point cloud, they created an adversarial point cloud with constraints applying perturbations to the low-frequency components and guiding perturbations to the high-frequency components. Huang *et al.* [73] proposed a new attack based on applying reversible coordinate transformations to points in the original point cloud, which reduces one degree of freedom and limits their movement on the tangent plane. The best direction is calculated based on the gradients of the transformed point clouds. After that, all points are assigned a score to construct the sensitivity map. Finally, top-scoring points are selected to fool deep models. The authors in [74] suggest that by analyzing the eigenvalues and eigenvectors of the graph Laplacian matrix of a point cloud, it can be determined which areas of the model are particularly sensitive to perturbations. By focusing on these areas, the attack can be crafted more effectively.

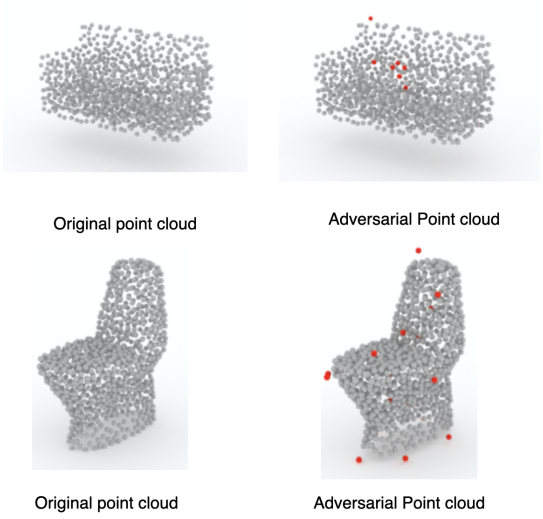

FIGURE 7: Two examples of original point cloud and minimal adversarial attack were proposed in [57].

F. MINIMAL LEVEL OF POINT MANIPULATIONS FOR ATTACKING

A special type of adversarial attacks exists in the 2D domain that focuses on perturbing a minimum number of pixels in adversarial attacks [63, 75, 76, 77, 78, 79, 80]. For instance, the one-pixel attack [75], which is the name given to the attack that can fool deep models by changing only one pixel, is a famous attack of this type. Taking inspiration from 2D attacks, Kim *et al.* [57] proposed adversarial attacks namely **minimal attack** that manipulate only a minimal number of points. To find an adversarial point cloud, they have modified the optimization function of the PGD attack 5 by adding a term. In this term, the number of changed points is kept to a minimum. Furthermore, they used two different distance measures, Hausdorff and Chamfer distance, to preserve the similarity between $\mathcal{P}$ and $\mathcal{P}^{adv}$. Figure 7 illustrates examples of minimal adversarial attack

In another attack called Variable Step-size Attack (**VSA**) [55], a hard boundary constraint on the number of modified points is incorporated into the optimization function of a PGD attack 5 to preserve the point cloud's appearance. In more concrete terms, certain points with the highest gradient norms (which have the most impact on classification tasks) are initialized as modified points. By controlling the step-size (large step-size (α) at the beginning and smaller at the end), this method escapes local optima and finds the most appropriate locations for the modified (adversarial) points.

Kim *et al.* [81] proposed a class of point cloud perturbation attacks called Nudge attacks that minimize point perturbation to flip 3D DNN results. The researchers generated adversarial point clouds using gradient-based and genetic algorithms with perturbations of up to 150 points in order to deceive DNNs. The attack can fool DNN even with a single point when the point has a large distance from the surface of 3D objects.

Yang *et al.* [45] provided a point-attachment attack by attaching a few points to the point cloud. A Chamfer distance is used to preserve a small distance between the newly added points and the original point cloud. Hard boundary constraints limit the number of points added in the point cloud, making it more difficult to detect.

Tan *et al.* [82] proposed a new type of attack called **One point attack** in which only a single point in the point cloud needs to be perturbed in order to fool the deep model. The authors also present an explainability method to identify the most important points in the point cloud for the attack

Shape Prior Guided Attack [83] is a method that uses a shape prior, or prior knowledge of the structure of the object, to guide the generation of the perturbations, or changes made to the point cloud to create the adversarial point cloud. The goal of this method is to create adversarial point clouds that have minimal perturbations while still being able to fool the target object detection model.

G. ATTACKS WITH DROP POINTS

Attacks described in the previous sections mostly revolved around shifting, adding, or transforming points (transforming points into another space and making changes there). This section reviews attacks that drop some points to generate adversarial point clouds. Depending on how points are dropped, these attacks can be made. The authors have provided various algorithms for removing critical points effectively. As an example, Zheng*et al.* [48] developed a method that by using a saliency map [84] finds critical points that are important in model decision-making and drops them. The points dropped by the saliency map are illustrated in red points in Figure 8. According to this method, every point is assigned a saliency score that reflects its contribution to the deep model recognition. By shifting high-saliency points towards the point cloud center, these points will not affect the surfaces much and practically operate in the same way as drop points. Consequently, the model can be deceived by shifting high-scoring points in a point cloud, resulting in adversarial point clouds. This method was proposed in two popular dropped attacks, **Drop100** and **Drop200**, which drop 100 and 200 points respectively.

An attack described in [47] identifies "adversarial drop points" in a 3D point cloud that, when dropped, significantly reduce a model's accuracy. These points are specified independently of the model by analyzing and combining fourteen-point cloud features and determining which features play key roles in the model's decision-making.

In [60], the critical points can be randomly determined and checked for dropping one by one. If a point increases the probability of changing the ground-truth label $f(\mathcal{P}) = Y$ is considered a critical point and, will be dropped. Otherwise, it will not be dropped. This procedure continues iteratively until the minimum critical points are dropped according to the following optimization problem

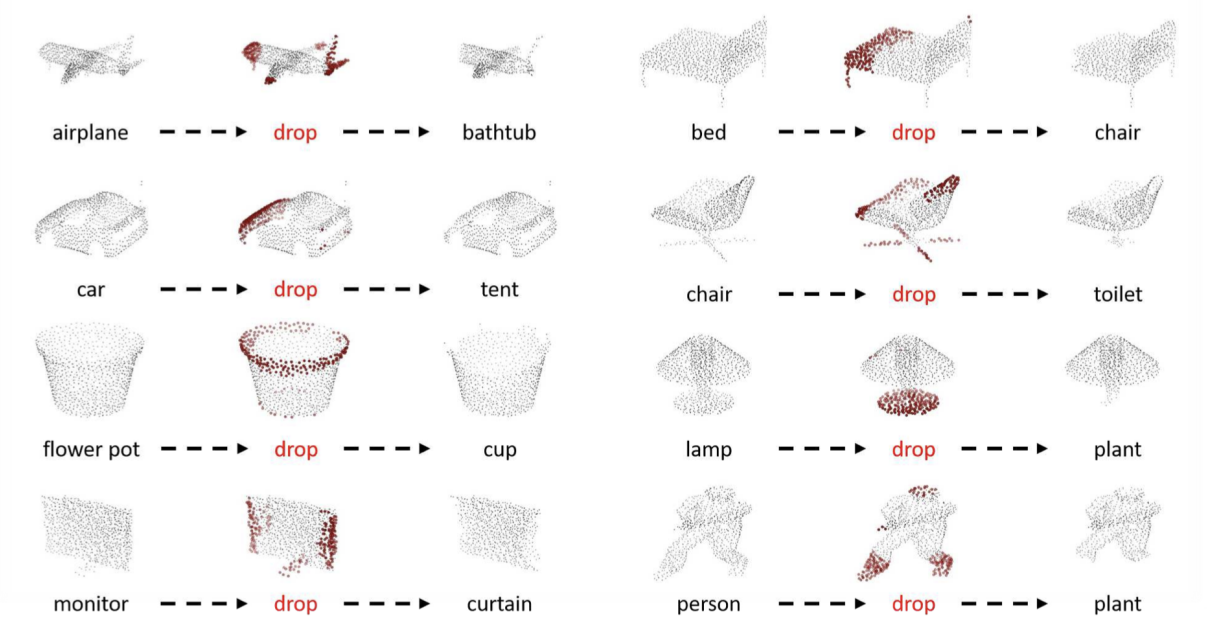

FIGURE 8: Original point clouds with labels(left), dropped points (red points) associated with highest scores(middle), and adversarial point clouds with estimated labels (right) were proposed in [48].

$$\begin{aligned} \min_{\mathcal{P}\subseteq\mathcal{P}^{adv}} \quad & (|\mathcal{P}^{adv}| - |\mathcal{P}|) \\ s.t. \quad & f(\mathcal{P}^{adv}) \neq f(\mathcal{P}) \end{aligned} \tag{9}$$

where $|\mathcal{P}^{adv}|$ and $|\mathcal{P}|$ are number points in the original point cloud and the adversarial one. The adversarial examples are generated by dropping critical points that optimize formula 9.

In order to determine the level of effectiveness of a given point in PointNet model decision-making, Yang*et al.* [45] introduced a Point-detachment attack that assigned a *class-dependent importance* to each point. A greedy strategy is employed to generate an adversarial point cloud, in which the most important point dependent on the true class are dropped iteratively. The *class-dependent importance* associated with a given point is determined by multiplying the two terms. The first term uses the PointNet feature matrix before max-pooling aggregation. (In this matrix, each row represents a point in the point cloud and each column represents a special feature). The second term uses from gradient the feature matrix w.r.t. the true class output, which is a sparse matrix with non-zero only at the critical points. If a given point has the largest value in some columns, the first term sums the difference between the first and second largest values in these columns. A bigger difference means more significance for the largest value. This means that a given point that corresponds to the largest value is more effective in the model decision. The second term sums up all values for a given point at a row level in the sparse matrix.

H. MISCELLANEOUS ATTACKS

Miao*et al.* [85] developed an adversarial point cloud based on rotation by applying an isometry matrix to the original point cloud. To find an appropriate isometry matrix the author used the Thompson Sampling method which can quickly find a suitable isometry matrix with a high attack rate.

Liu *et al.* [59] proposed an Imperceptible Transfer Attack (**ITA**) that enhances the imperceptibility of adversarial point clouds by shifting each point in the direction of its normal vector.

Zhang *et al.* [86] proposed a Mesh Attack that directly perturbs the mesh of a 3D object.

Tang *et al.* [87] presented a method called NormalAttack for generating imperceptible point cloud attacks. The method deforms objects along their normals by considering the object's curvature to make the modification less noticeable.

IV. DEFENSES AGAINST ADVERSARIAL ATTACKS

Adversarial defense methods for 3D point clouds can generally be divided into three categories: input transformation, data optimization, and deep model modification. The following sections discuss defense methods under each of these categories.

A. INPUT TRANSFORMATION

An input transformation is a preprocessing approach that involves applying some transformations to the input point cloud before it is fed into the deep model. This transformation could be designed to reduce the sensitivity of the model to adversarial attacks or to make it more difficult for an attacker to craft an adversarial point cloud. Input transformation methods are listed below.

1) Simple Random Sampling (SRS)

Simple random sampling [46] is a statistical technique commonly known as **SRS** that randomly drops a certain number of points (usually 500) from an input point cloud (with the same probability).

2) Statistical Outlier Removal (SOR)

Since there exist outliers in most adversarial attacks, Zhou *et al.* [88] proposed a statistical outlier removal (**SOR**) method that trimmed the points in an adversarial point cloud if the average distance a point to its k nearest neighbors falls outside the $(\mu + \sigma.\alpha)$, which μ is mean and σ is the standard deviation of k nearest neighbor distance of all points in the original point cloud. Depending on the size of the analyzed neighborhood, α will be determined. (In [88] α = 1.1 and k=2 are considered).

A similar defense method is used in [89]. The Euclidean distance between each point and its k-nearest neighbors is used to detect outliers. Points with High mean distances are discarded as outliers.

3) Salient points removal

This defense method [54] assumes that the adversarial points have fairly large gradient values. Taking this as true, this method calculated the saliency of each point based on the gradient output class of the model f w.r.t. to each point and points with high saliency were discarded.

4) Denoiser and Upsampler Network (DUP-Net)

The DUP-Net defense method consists of two steps. To remove outliers, it uses SOR as a denoiser in the first step. In the second step, the output of the first step is given to an upsampler network [90] to produce a denser point cloud. It is generally found that adversarial perturbations are missing critical points from original point clouds, so this defense uses a denser point cloud tracking the underlying surface of the point cloud with uniform distribution to recover these critical points.

5) IF-Defense

IF-Defense [91] is a preprocessing technique on the input point cloud. It first employs SOR to remove outliers from the input point cloud. In the next step, two losses are used to optimize input points' coordinates under geometry- and distribution-aware constraints. The geometry-aware loss tries to push points towards the surface in order to minimize outliers. To estimate the surfaces of objects, the authors train an implicit function network [92, 93] on original point clouds. Because output of implicit functions are continuous, the predicted surface is locally smooth. This reduces outlier effects. The distribution-aware loss encourages points to have an uniform distribution by maximizing the distance between each point and its k-nearest neighbors. Accordingly, the input point clouds are captured in a clean shape using If-Defense.

Figure 9 shows the results of three different defense methods against a Drop100 attack, including SOR, DUP-Net, and If-defense.

6) Miscellaneous Defenses

Dong *et al.* [94] proposed Gather-Vector Guidance (GvG) method which is sensitive to the change of local features. In case the adversarial perturbation changes the local features, the gather-vector will also change. This method learns to ignore noisy local features.

Liu *et al.* [95] developed PointGuard, a method that creates a number of random subsets of points in the original point cloud, then predicts the label of the original point cloud based on the majority vote among the labels of these random subsets.

Sun*et al.* [96] proposed a framework for evaluating the robustness of 3D point cloud classification models to adaptive attack.

Ada3Diff [97] is a method for defending against adversarial attacks on 3D point cloud models. It uses an adaptive diffusion process to smooth out perturbations in the point cloud, effectively reducing the impact of the adversarial attack.

B. DATA OPTIMIZATION

Another category is data optimization for training, which involves optimizing the training data to improve the robustness of the deep model to adversarial attacks. This could involve techniques such as data augmentation, which involves generating additional training examples by applying transformations to the existing training data, or adversarial training, which involves intentionally introducing adversarial examples into the training data in order to improve the model's robustness to such attacks. The following methods can be used to optimize data.

1) Adversarial Training

In terms of modified training sets, adversarial training [61] is an effective defense method, which augments the training set with adversarial examples to increase the model's robustness against attacks. To be precise, in standard training, the model is trained using only the original point clouds, while adversarial training uses both original and adversarial point clouds. The adversarial training for point clouds is described in [54] for the first time. The authors of [54] and [59] trained a deep model by augmenting the FGSM and ITA attacks. As a way to find a stronger adversarial training method, the authors in [98] used adaptive attacks. Using this new adversarial training, different types of attacks are added to the deep model by embedding a perturbation-injection module. This module is utilized to generate the perturbed features for adversarial training. Sun *et al.* [99] applied self-supervised learning to adversarial training with 3D point clouds.

In different tries, the authors in [45, 100] add Gaussian noise to each point by randomly sampling values from a Gaussian distribution. By doing so, the attacked models

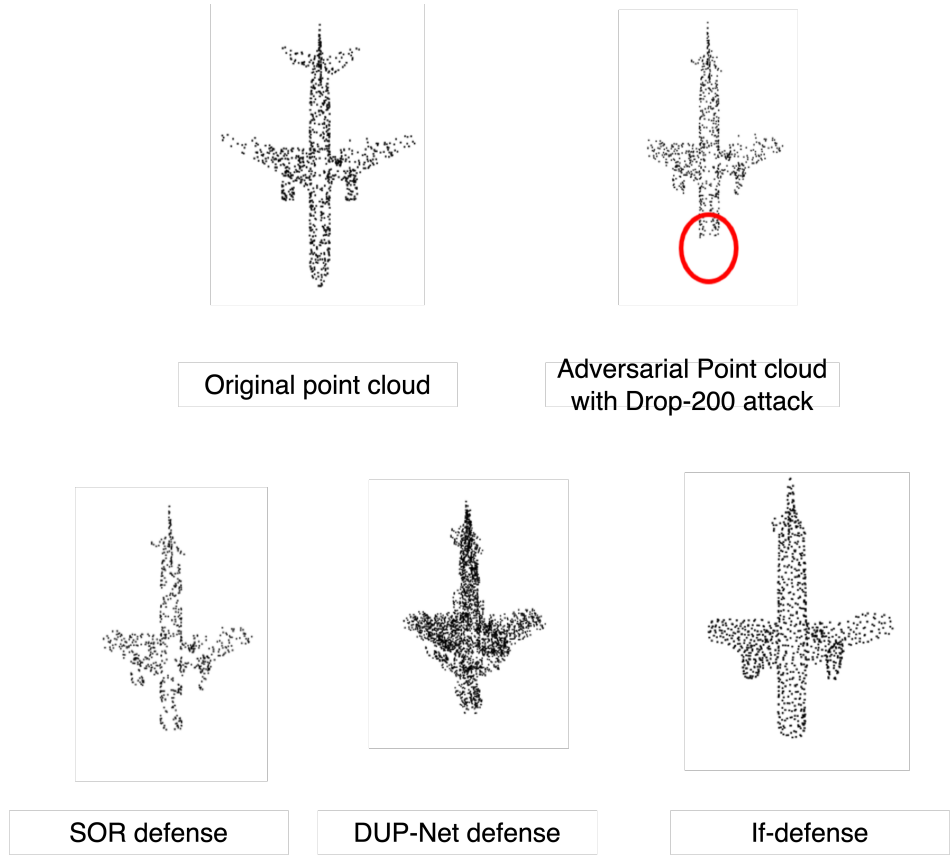

FIGURE 9: Results of three different defense methods on Drop100 attack. Figure taken from [91].

can escape from the narrow adversarial subspace. Also, they developed a Quantification Method for converting point cloud coordinates into low numerical precision with multiple quantification levels, which mitigates small variations in coordinates. These noisy point clouds are then used to augment training sets.

2) PointCutMix

Zhang *et al.* [101] proposed PointCutMix technique that generated a new training set by swapping points between two optimally aligned original point clouds and training a model with this new training set.

3) Low Pass Frequency-Defense (LPF-Defense)

In LPF-Defense [102], deep models are trained with the low-frequency version of the original point cloud. More specifically, with the Spherical Harmonic Transform (SHT) [103], original point clouds were transformed from the spatial to the frequency domain. The low-frequency version of the original point cloud is then retrieved back into the spatial domain by filtering the high-frequency input data components. This method is based on the assumption that 3D deep models are overly dependent on features with unnecessary information in the training sets, making them vulnerable to adversarial point clouds. Therefore it discards the unnecessary information from the training data by suppressing the high-frequency contents in the training phase.

C. DEEP MODEL MODIFICATION

Another category is deep model modifications, which refer to modifying the architecture of the deep model itself in order to improve its robustness to adversarial attacks. This could be achieved by making changes to the original deep neural network architecture during training. Examples of this category are given below.

1) Defense-PointNet

The authors in [104] have provided a defense method by splitting the PointNet deep model into two parts. The first part is the feature extractor, with a discriminator attached to its last layer enabling it to learn more powerful features. The feature extractor feeds a mini-batch of the original point cloud and the adversarial counterpart (generated by the FGSM attack) as input to extract features and also fool the discriminator. The second part is the PointNet classifier which is trained to classify each input correctly. The model parameters are optimized using three different loss functions: a classifier, a discriminator, and a feature extractor. While discriminator loss attempts to distinguish the original point cloud from the adversarial one, feature extractor loss misleads the discriminator to label every original/adversarial

vector as the original and classifier loss encourages the classifier to give correct predictions for each input.

2) Context-Consistency dynamic graph Network (CCN)

Li*et al.* [105] proposed two methodologies to improve the adversarial robustness of 3D point cloud classification models. The first methodology is the introduction of a novel point cloud architecture called Context-Consistency dynamic graph Network (CCN), which is designed to be more robust to adversarial attacks. The second methodology involves an in-depth analysis of the factors that affect the robustness of point cloud models, and the development of techniques to mitigate these factors. In order to provide a more robust model against adversarial point clouds, the authors integrate the two techniques

3) Lattice Point Classifier (LPC)

Li *et al.* [106] proposed embedding a declarative node into the networks to transform adversarial examples to the clean manifold. The authors proposed an effective instantiation, the Lattice Point Classifier (LPC), which projects each point cloud onto the lattice and generates a 2D image for classification using 2D CNNs. (Structured sparse coding in the permutohedral lattice is defined as the declarative node in LPC.). The declarative nodes defend the adversarial attacks through implicit gradients by leading them to wrong updating directions for inputs.

V. TAXONOMY OF DATASETS AND VICTIM MODELS

A variety of 3D point cloud datasets have been collected to evaluate shape classification on DNNs, including ModelNet [107], ShapeNet [108], ScanObjectNN [109], McGill Benchmark [110], ScanNet [111], Sydney Urban Objects [112]. A summary of the characteristics of these datasets is also provided in Table 4. Among all, 4 datasets namely ModelNet10 [107], ModelNet40 [107], ShapeNet [108] and ScanObjectNN [109] have mostly been used to evaluate attack and defense techniques.

Also, there is a taxonomy of datasets and victim models used in recent studies in Table 5.

VI. CHALLENGES AND DISCUSSIONS

This section discusses the current challenges that adversarial point clouds face, as well as the potential solutions that can be found. For both adversaries and researchers, adversarial point clouds are an interesting problem, which exploits the vulnerability of deep models and helps defenders avoid adversarial point clouds. Our discussion will focus on the following questions. What factors affect the attack on Point Cloud?

A. WHAT FACTORS AFFECT THE SUCCESS OF ADVERSARIAL ATTACKS ON 3D POINT CLOUDS?

There are some general factors that be more important for adversarial attacks on 3D point clouds including: The complexity and robustness of the model being attacked: When a deep model is less complex and less robust, it may be less immune to adversarial attacks and require a less sophisticated or weaker attack to fool it. The structure of the 3D point cloud: The distribution of points in the point cloud and the presence of outliers can potentially affect the success of most types of adversarial point clouds.

B. COMPARISON OF DIFFERENT DEFENSE METHODS

A 3D point cloud's distribution and outliers can significantly impact the effectiveness of defense methods against adversarial point clouds. For example, input transformation techniques are designed to make it more difficult for an attacker to craft adversarial point clouds. These techniques may rely on modifying the distribution of points in the point cloud or dropping outliers. By doing this, the structure of the original point cloud is disrupted. This makes it harder for the attacker to make successful modifications. On the other hand, other defense methods, such as adversarial training, may not rely as heavily on these factors and may not be as efficient. Adversarial training is one of the most powerful defenses in the 2D defense techniques, but it does not do well in 3D data. The paper [64] proves that the adversarial training maximizes the classifier loss by finding a worst-case example inside a constrained search space. This procedure can change decision boundaries so that the model gets more robust to different types of attacks. This proof is based on the regular structure of 2D data. Creating 2D attacks is performed by changing the pixel values. Note that in the 2D case, the data has a regular structure. But, a point cloud consists of a set of 3D data points that are placed irregularly in space. Furthermore, the point clouds used in the literatures are constructed by randomly sampling 1024 points from each 3D object. Therefore, points are not uniformly distributed across object's surface and any two point clouds from the same class (e.g., airplane) do not have the same regular structure, as opposed to the 2D cases. These structural differences result in different defense behaviors in the adversarial training phase. Therefore, training the model with the worst-case example inside a constrained search space can not guarantee robustness against other attacks. In other words, due to the irregular structure of point clouds, it is very challenging to model adversarial points to eliminate their impact on defense.

C. COMPARISON OF 3D POINT CLOUDS AND IMAGE DATA IN TERMS OF ATTACKS AND DEFENSES

There are several differences between 3D point clouds and images in terms of adversarial attacks and defenses: An adversarial attack on 3D point clouds can be more complex. Typically, an adversarial attack on an image data involves adding small perturbations to the pixel values. In contrast, adversarial attacks on 3D point clouds can involve more complex modifications, such as adding or dropping points, or changing the connectivity of the points in the point cloud. In fact, the structure of 3D point clouds is different from that of images. Images are typically represented as 2D arrays of pixel values, while 3D point clouds are represented as sets

TABLE 4: Summary of the commonly used 3D datasets.

Datasets				
Dataset Name	Year	Type	Classes	Samples(Training/Test)
McGill Benchmark [110]	2008	Synthetic	19	456(304/152)
Sydney Urban Objects [112]	2013	Real-World	14	588(/)
ShapeNet [108]	2015	Synthetic	55	51190(/)
ModelNet10 [107]	2015	Synthetic	10	4899(3991/605)
ModelNet40 [107]	2015	Synthetic	40	12311(9843/2468)
ScanNet [111]	2017	Real-World	17	12283(9677/2606)
ScanObjectNN [109]	2019	Real-World	15	2902(2321/581)

TABLE 5: Taxonomy of datasets and victim models used in attacks and defenses on 3D point clouds.

Datasets	ModelNet10 [107]	[60] ,[96],[113] ,[81]
	ModelNet40 [107]	[54],[60] ,[59] ,[57],[56],[55],[53],[51] ,[52],[48] [46],[91],[73] ,[114] ,[98],[115],[96],[116],[113],[86] [50],[94] ,[81],[117],[58],[88],[118],[106],[85],[72],[83],[119]
	ShapeNet [108]	[55],[51] ,[49] ,[91],[114],[117],[69],[104]
	ScanObjectNN [109]	[57] ,[115],[95], [96],[113]
	KITTI[120]	[60] ,[68],[121]
	ScanNet[27]	[95],[106]
	3DMNIST[122]	[48],[83]
Victim models	PointNet[27]	[54] , [60], [45],[59] ,[57],[56],[55],[51],[52] ,[49] [48],[46],[91] ,[98],[71],[115],[95],[96],[116],[113] [68],[86],[50],[94] ,[81],[117],[69],[104],[58],[88] [121],[118],[106],[85] ,[72],[83],[119]
	PointNet++[123]	[54] , [45] ,[59],[57] ,[56],[55] ,[53],[51],[52] ,[49] [48],[46],[91],[73],[114] ,[98],[71],[116],[68],[86] [50],[94],[117],[58],[88],[121],[85],[72],[83],[119]
	DGCNN[124]	[45] ,[59],[57] ,[56],[55],[51],[52] ,[49] ,[48],[46] [91],[73],[114] ,[98],[71],[95],[113],[68],[86],[50] [81],[117],[58],[118],[85],[72],[83],[119]
	PointConv[125]	[91],[114],[71]
	RS-CNN[126]	[91],[119]
	VoxNet[127]	[60]
	SpiderCNN[128]	[57]
	PointASNL[129]	[57]
	CurveNet[130]	[73]
	AtlasNet[131]	[69]
	PointTrans[132]	[72]
	PointMLP[133]	[72]

of 3D points. This difference in structure can make it more challenging to apply defense methods that were developed for image data to 3D point clouds. On the other hand, 3D point clouds can be more sensitive to perturbations. Because 3D point clouds are used to represent physical objects in the real world, even small perturbations to the point cloud can result in significant changes to the shape or appearance of the represented object. This sensitivity can make it more difficult to develop robust defense methods for 3D point clouds.

VII. CONCLUSION

Adversarial attacks on 3D point cloud classifications have become a significant concern in recent years. These attacks can successfully manipulate the classification of 3D point clouds, leading to incorrect decisions with potentially harmful consequences. Adversarial attacks on 3D point clouds can be categorized into several types, including drop attacks, add attacks, shift attacks, and transform attacks.

To defend against these attacks, researchers have proposed two main categories of approaches: input transformation and adversarial training. Input transformation methods aim to preprocess the input data in order to make it more robust to adversarial perturbations, while adversarial training involves augmenting the training data with adversarial examples in order to improve the model's robustness. For more robust protection against adversarial attacks, input transformation techniques can be combined with adversarial training.

Some potential future directions for research on adversarial attacks on 3D point clouds include optimizing attack methods by targeting only a subset of points in the point cloud and focusing on the local rather than global structure of the point cloud, as well as exploring the robustness of 3D point cloud classifiers to attacks that are specifically designed for 3D data rather than adapted from methods developed for 2D images.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," nature, vol. 521, no. 7553, pp. 436–444, 2015.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural

networks," Communications of the ACM, vol. 60, no. 6, pp. 84–90, 2017.
[4] N. Silberman and R. Fergus, "Indoor scene segmentation using a structured light sensor," in 2011 IEEE international conference on computer vision workshops (ICCV workshops). IEEE, 2011, pp. 601–608.
[5] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A review on deep learning techniques applied to semantic segmentation," arXiv preprint arXiv:1704.06857, 2017.
[6] G. Saon, H.-K. J. Kuo, S. Rennie, and M. Picheny, "The ibm 2015 english conversational telephone speech recognition system," arXiv preprint arXiv:1505.05899, 2015.
[7] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, "Speech recognition using deep neural networks: A systematic review," IEEE access, vol. 7, pp. 19 143–19 165, 2019.
[8] K. Chowdhary, "Natural language processing," Fundamentals of artificial intelligence, pp. 603–649, 2020.
[9] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2014.
[10] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," arXiv preprint arXiv:1611.01236, 2016.
[11] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2574–2582.
[12] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," arXiv preprint arXiv:1705.07204, 2017.
[13] F. Tramer, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," Advances in Neural Information Processing Systems, vol. 33, pp. 1633–1645, 2020.
[14] C. Xiao, J.-Y. Zhu, B. Li, W. He, M. Liu, and D. Song, "Spatially transformed adversarial examples," arXiv preprint arXiv:1801.02612, 2018.
[15] H. Zhao, T. Le, P. Montague, O. De Vel, T. Abraham, and D. Phung, "Perturbations are not enough: Generating adversarial examples with spatial distortions," arXiv preprint arXiv:1910.01329, 2019.
[16] T. Deng and Z. Zeng, "Generate adversarial examples by spatially perturbing on the meaningful area," Pattern Recognition Letters, vol. 125, pp. 632–638, 2019.
[17] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 1625–1634.
[18] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," arXiv preprint arXiv:1712.09665, 2017.
[19] X. Yang, F. Wei, H. Zhang, and J. Zhu, "Design and interpretation of universal adversarial patches in face detection," in European Conference on Computer Vision. Springer, 2020, pp. 174–191.
[20] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, and A. Madry, "Exploring the landscape of spatial robustness," 2017.
[21] M. Mozaffari-Kermani, S. Sur-Kolay, A. Raghunathan, and N. K. Jha, "Systematic poisoning attacks on and defenses for machine learning in healthcare," IEEE journal of biomedical and health informatics, vol. 19, no. 6, pp. 1893–1905, 2014.
[22] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. M. Paixao, F. Mutz et al., "Self-driving cars: A survey," Expert Systems with Applications, vol. 165, p. 113816, 2021.
[23] M. Hassanalian and A. Abdelkefi, "Classifications, applications, and design challenges of drones: A review," Progress in Aerospace Sciences, vol. 91, pp. 99–131, 2017.
[24] H. A. Pierson and M. S. Gashler, "Deep learning in robotics: a review of recent research," Advanced Robotics, vol. 31, no. 16, pp. 821–835, 2017.
[25] Z. Liu, H. Tang, Y. Lin, and S. Han, "Point-voxel cnn for efficient 3d deep learning," Advances in Neural Information Processing Systems, vol. 32, 2019.
[26] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," arXiv preprint arXiv:1312.6203, 2013.
[27] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 652–660.
[28] N. Akhtar, A. Mian, N. Kardan, and M. Shah, "Advances in adversarial attacks and defenses in computer vision: A survey," IEEE Access, vol. 9, pp. 155 161–155 196, 2021.
[29] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," Ieee Access, vol. 6, pp. 14 410–14 430, 2018.
[30] H. Wei, H. Tang, X. Jia, H. Yu, Z. Li, Z. Wang, S. Satoh, and Z. Wang, "Physical adversarial attack meets computer vision: A decade survey," arXiv preprint arXiv:2209.15179, 2022.
[31] S. Qiu, Q. Liu, S. Zhou, and W. Huang, "Adversarial attack and defense technologies in natural language processing: A survey," Neurocomputing, vol. 492, pp. 278–307, 2022.
[32] S. Y. Khamaiseh, D. Bagagem, A. Al-Alaj, M. Mancino, and H. W. Alomari, "Adversarial deep learning: A survey on adversarial attacks and defense mecha-

nisms on image classification," IEEE Access, 2022.
[33] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," IEEE transactions on neural networks and learning systems, vol. 30, no. 9, pp. 2805–2824, 2019.
[34] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "A survey on adversarial attacks and defences," CAAI Transactions on Intelligence Technology, vol. 6, no. 1, pp. 25–45, 2021.
[35] A. Michel, S. K. Jha, and R. Ewetz, "A survey on the vulnerability of deep neural networks against adversarial attacks," Progress in Artificial Intelligence, pp. 1–11, 2022.
[36] S. Qiu, Q. Liu, S. Zhou, and C. Wu, "Review of artificial intelligence adversarial attack and defense technologies," Applied Sciences, vol. 9, no. 5, p. 909, 2019.
[37] K. D. Gupta and D. Dasgupta, "Adversarial attacks and defenses for deployed ai models," IT Professional, vol. 24, no. 4, pp. 37–41, 2022.
[38] R. R. Wiyatno, A. Xu, O. Dia, and A. de Berker, "Adversarial examples in modern machine learning: A review," arXiv preprint arXiv:1911.05268, 2019.
[39] H. Tan, L. Wang, H. Zhang, J. Zhang, M. Shafiq, and Z. Gu, "Adversarial attack and defense strategies of speaker recognition systems: A survey," Electronics, vol. 11, no. 14, p. 2183, 2022.
[40] H. Liang, E. He, Y. Zhao, Z. Jia, and H. Li, "Adversarial attack and defense: A survey," Electronics, vol. 11, no. 8, p. 1283, 2022.
[41] Y. Li, M. Cheng, C.-J. Hsieh, and T. C. Lee, "A review of adversarial attack and defense for classification methods," The American Statistician, pp. 1–17, 2022.
[42] X. Wei, B. Pu, J. Lu, and B. Wu, "Physically adversarial attacks and defenses in computer vision: A survey," arXiv preprint arXiv:2211.01671, 2022.
[43] J.-X. Mi, X.-D. Wang, L.-F. Zhou, and K. Cheng, "Adversarial examples based on object detection tasks: A survey," Neurocomputing, 2022.
[44] D. Tian, H. Ochimizu, C. Feng, R. Cohen, and A. Vetro, "Geometric distortion metrics for point cloud compression," in Proc. IEEE ICIP, 2017, pp. 3460–3464.
[45] J. Yang, Q. Zhang, R. Fang, B. Ni, J. Liu, and Q. Tian, "Adversarial attack and defense on point sets," arXiv preprint arXiv:1902.10899, 2019.
[46] C. Xiang, C. R. Qi, and B. Li, "Generating 3d adversarial point clouds," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9136–9144.
[47] H. Naderi, C. Dinesh, I. V. Bajic, and S. Kasaei, "Model-free prediction of adversarial drop points in 3d point clouds," arXiv preprint arXiv:2210.14164, 2022.
[48] T. Zheng, C. Chen, J. Yuan, B. Li, and K. Ren, "Pointcloud saliency maps," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1598–1606.
[49] A. Hamdi, S. Rojas, A. Thabet, and B. Ghanem, "Advpc: Transferable adversarial perturbations on 3d point clouds," in European Conference on Computer Vision. Springer, 2020, pp. 241–257.
[50] K. Lee, Z. Chen, X. Yan, R. Urtasun, and E. Yumer, "Shapeadv: Generating shape-aware adversarial 3d point clouds," arXiv preprint arXiv:2005.11626, 2020.
[51] H. Zhou, D. Chen, J. Liao, K. Chen, X. Dong, K. Liu, W. Zhang, G. Hua, and N. Yu, "Lg-gan: Label guided adversarial network for flexible targeted attack of point cloud based deep networks," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10 356–10 365.
[52] Y. Wen, J. Lin, K. Chen, C. P. Chen, and K. Jia, "Geometry-aware generation of adversarial point clouds," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020.
[53] T. Tsai, K. Yang, T.-Y. Ho, and Y. Jin, "Robust adversarial objects against deep learning models," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 01, 2020, pp. 954–962.
[54] D. Liu, R. Yu, and H. Su, "Extending adversarial attacks and defenses to deep 3d point cloud classifiers," in 2019 IEEE International Conference on Image Processing (ICIP). IEEE, 2019, pp. 2279–2283.
[55] A. Arya, H. Naderi, and S. Kasaei, "Adversarial attack by limited point cloud surface modifications," in 2023 6th International Conference on Pattern Recognition and Image Analysis (IPRIA). IEEE, 2023, pp. 1–8.
[56] D. Liu, R. Yu, and H. Su, "Adversarial shape perturbations on 3d point clouds," in European Conference on Computer Vision. Springer, 2020, pp. 88–104.
[57] J. Kim, B.-S. Hua, T. Nguyen, and S.-K. Yeung, "Minimal adversarial examples for deep learning on 3d point clouds," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 7797–7806.
[58] C. Ma, W. Meng, B. Wu, S. Xu, and X. Zhang, "Efficient joint gradient based attack against sor defense for 3d point cloud classification," in Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 1819–1827.
[59] D. Liu and W. Hu, "Imperceptible transfer attack and defense on 3d point cloud classification," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022.
[60] M. Wicker and M. Kwiatkowska, "Robustness of 3d deep learning in an adversarial setting," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 11 767–11 775.
[61] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2015.
[62] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in Proceedings of the IEEE conference on computer

vision and pattern recognition, 2018, pp. 9185–9193.
[63] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in 2017 ieee symposium on security and privacy (sp). IEEE, 2017, pp. 39–57.
[64] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2019.
[65] T. B. Brown, N. Carlini, C. Zhang, C. Olsson, P. Christiano, and I. Goodfellow, "Unrestricted adversarial examples," arXiv preprint arXiv:1809.08352, 2018.
[66] H. Naderi, L. Goli, and S. Kasaei, "Generating unrestricted adversarial examples via three parameters," Multimedia Tools and Applications, vol. -, no. -, pp. –, 2022.
[67] Y. Song, R. Shu, N. Kushman, and S. Ermon, "Constructing unrestricted adversarial examples with generative models," Advances in Neural Information Processing Systems, vol. 31, 2018.
[68] X. Dai, Y. Li, H. Dai, and B. Xiao, "Generating unrestricted 3d adversarial point clouds," arXiv preprint arXiv:2111.08973, 2021.
[69] I. Lang, U. Kotlicki, and S. Avidan, "Geometric adversarial attacks and defenses on 3d point clouds," in 2021 International Conference on 3D Vision (3DV). IEEE, 2021, pp. 1196–1205.
[70] G. Mariani, L. Cosmo, A. M. Bronstein, and E. Rodola, "Generating adversarial surfaces via band-limited perturbations," in Computer Graphics Forum, vol. 39, no. 5. Wiley Online Library, 2020, pp. 253–264.
[71] B. Liu, J. Zhang, L. Chen, and J. Zhu, "Boosting 3d adversarial attacks with attacking on frequency," arXiv preprint arXiv:2201.10937, 2022.
[72] D. Liu, W. Hu, and X. Li, "Point cloud attacks in graph spectral domain: When 3d geometry meets graph signal processing," arXiv preprint arXiv:2207.13326, 2022.
[73] Q. Huang, X. Dong, D. Chen, H. Zhou, W. Zhang, and N. Yu, "Shape-invariant 3d adversarial point clouds," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 15 335–15 344.
[74] Q. Hu, D. Liu, and W. Hu, "Exploring the devil in graph spectral domain for 3d point cloud attacks," arXiv preprint arXiv:2202.07261, 2022.
[75] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," IEEE Transactions on Evolutionary Computation, vol. 23, no. 5, pp. 828–841, 2019.
[76] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in 2016 IEEE European symposium on security and privacy (EuroS&P). IEEE, 2016, pp. 372–387.
[77] A. Modas, S.-M. Moosavi-Dezfooli, and P. Frossard, "Sparsefool: a few pixels make a big difference," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 9087–9096.
[78] F. Croce and M. Hein, "Sparse and imperceivable adversarial attacks," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 4724–4732.
[79] N. Narodytska and S. P. Kasiviswanathan, "Simple black-box adversarial perturbations for deep networks," arXiv preprint arXiv:1612.06299, 2016.
[80] L. Schott, J. Rauber, M. Bethge, and W. Brendel, "Towards the first adversarially robust neural network model on mnist," arXiv preprint arXiv:1805.09190, 2018.
[81] Y. Zhao, I. Shumailov, R. Mullins, and R. Anderson, "Nudge attacks on point-cloud dnns," arXiv preprint arXiv:2011.11637, 2020.
[82] H. Tan and H. Kotthaus, "Explainability-aware one point attack for point cloud neural networks," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2023, pp. 4581–4590.
[83] Z. Shi, C. Zhi, X. Zhenbo, Y. Wei, Y. Zhidong, and L. Huang, "Shape prior guided attack: Sparser perturbations on 3d point clouds," in Proceedings of the AAAI Conference on Artificial Intelligence, 2022.
[84] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2921–2929.
[85] Y. Miao, Y. Dong, J. Zhu, and X.-S. Gao, "Isometric 3d adversarial examples in the physical world," arXiv preprint arXiv:2210.15291, 2022.
[86] J. Zhang, L. Chen, B. Liu, B. Ouyang, Q. Xie, J. Zhu, W. Li, and Y. Meng, "3d adversarial attacks beyond point cloud," arXiv preprint arXiv:2104.12146, 2021.
[87] K. Tang, Y. Shi, J. Wu, W. Peng, A. Khan, P. Zhu, and Z. Gu, "Normalattack: Curvature-aware shape deformation along normals for imperceptible point cloud attack," Security and Communication Networks, vol. 2022, 2022.
[88] H. Zhou, K. Chen, W. Zhang, H. Fang, W. Zhou, and N. Yu, "Dup-net: Denoiser and upsampler network for 3d adversarial point clouds defense," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1961–1970.
[89] ——, "Deflecting 3d adversarial point clouds through outlier-guided removal," arXiv preprint arXiv:1812.11017, 2018.
[90] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "Pu-net: Point cloud upsampling network," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 2790–2799.
[91] Z. Wu, Y. Duan, H. Wang, Q. Fan, and L. J. Guibas, "If-defense: 3d adversarial point cloud defense via implicit function based restoration," arXiv preprint

arXiv:2010.05272, 2020.
[92] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, "Convolutional occupancy networks," in European Conference on Computer Vision. Springer, 2020, pp. 523–540.
[93] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 4460–4470.
[94] X. Dong, D. Chen, H. Zhou, G. Hua, W. Zhang, and N. Yu, "Self-robust 3d point recognition via gather-vector guidance," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020, pp. 11 513–11 521.
[95] H. Liu, J. Jia, and N. Z. Gong, "Pointguard: Provably robust 3d point cloud classification," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 6186–6195.
[96] J. Sun, K. Koenig, Y. Cao, Q. A. Chen, and Z. Mao, "On the adversarial robustness of 3d point cloud classification," 2020.
[97] K. Zhang, H. Zhou, J. Zhang, Q. Huang, W. Zhang, and N. Yu, "Ada3diff: Defending against 3d adversarial point clouds via adaptive diffusion," arXiv preprint arXiv:2211.16247, 2022.
[98] Q. Liang, Q. Li, W. Nie, and A.-A. Liu, "Pagn: perturbation adaption generation network for point cloud adversarial defense," Multimedia Systems, pp. 1–9, 2022.
[99] J. Sun, Y. Cao, C. B. Choy, Z. Yu, A. Anandkumar, Z. M. Mao, and C. Xiao, "Adversarially robust 3d point cloud recognition using self-supervisions," Advances in Neural Information Processing Systems, vol. 34, 2021.
[100] Y. Zhang, J. Hou, and Y. Yuan, "A comprehensive study and comparison of the robustness of 3d object detectors against adversarial attacks," arXiv preprint arXiv:2212.10230, 2022.
[101] J. Zhang, L. Chen, B. Ouyang, B. Liu, J. Zhu, Y. Chen, Y. Meng, and D. Wu, "Pointcutmix: Regularization strategy for point cloud classification," Neurocomputing, vol. 505, pp. 58–67, 2022.
[102] H. Naderi, K. Noorbakhsh, A. Etemadi, and S. Kasaei, "Lpf-defense: 3d adversarial defense based on frequency analysis," Plos one, vol. 18, no. 2, p. e0271388, 2023.
[103] T. S. Cohen, M. Geiger, J. Köhler, and M. Welling, "Spherical cnns," arXiv preprint arXiv:1801.10130, 2018.
[104] Y. Zhang, G. Liang, T. Salem, and N. Jacobs, "Defense-pointnet: Protecting pointnet against adversarial attacks," in 2019 IEEE International Conference on Big Data (Big Data). IEEE, 2019, pp. 5654–5660.
[105] G. Li, G. Xu, H. Qiu, R. He, J. Li, and T. Zhang, "Improving adversarial robustness of 3d point cloud classification models," in European Conference on Computer Vision. Springer, 2022, pp. 672–689.
[106] K. Li, Z. Zhang, C. Zhong, and G. Wang, "Robust structured declarative classifiers for 3d point clouds: Defending adversarial attacks with implicit gradients," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 15 294–15 304.
[107] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1912–1920.
[108] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su et al., "Shapenet: An information-rich 3d model repository," arXiv preprint arXiv:1512.03012, 2015.
[109] M. A. Uy, Q.-H. Pham, B.-S. Hua, T. Nguyen, and S.-K. Yeung, "Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data," in Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 1588–1597.
[110] K. Siddiqi, J. Zhang, D. Macrini, A. Shokoufandeh, S. Bouix, and S. Dickinson, "Retrieving articulated 3-d models using medial surfaces," Machine vision and applications, vol. 19, no. 4, pp. 261–275, 2008.
[111] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 5828–5839.
[112] M. De Deuge, A. Quadros, C. Hung, and B. Douillard, "Unsupervised feature learning for classification of outdoor 3d scans," in Australasian conference on robitics and automation, vol. 2. University of New South Wales Kensington, Australia, 2013, p. 1.
[113] J. Sun, Y. Cao, C. Choy, Z. Yu, C. Xiao, A. Anandkumar, and Z. M. Mao, "Improving adversarial robustness in 3d point cloud classification via self-supervisions," in International Conference on Machine Learning Workshop (ICMLW), vol. 1, 2021.
[114] K. Tang, Y. Shi, T. Lou, W. Peng, X. He, P. Zhu, Z. Gu, and Z. Tian, "Rethinking perturbation directions for imperceptible adversarial attacks on point clouds," IEEE Internet of Things Journal, 2022.
[115] D. D. Denipitiyage, T. Ajanthan, P. Kamalaruban, and A. Weller, "Provable defense against clustering attacks on 3d point clouds," in The AAAI-22 Workshop on Adversarial Machine Learning and Beyond, 2021.
[116] Y. Sun, F. Chen, Z. Chen, and M. Wang, "Local aggressive adversarial attacks on 3d point cloud," in Asian Conference on Machine Learning. PMLR, 2021, pp. 65–80.
[117] Y. Zhao, Y. Wu, C. Chen, and A. Lim, "On isometry robustness of deep 3d point cloud models under adver-

sarial attacks," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1201–1210.
[118] C. Ma, W. Meng, B. Wu, S. Xu, and X. Zhang, "Towards effective adversarial attack against 3d point cloud classification," in 2021 IEEE International Conference on Multimedia and Expo (ICME). IEEE, 2021, pp. 1–6.
[119] F. He, Y. Chen, R. Chen, and W. Nie, "Point cloud adversarial perturbation generation for adversarial attacks," IEEE Access, vol. 11, pp. 2767–2774, 2023.
[120] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in 2012 IEEE conference on computer vision and pattern recognition. IEEE, 2012, pp. 3354–3361.
[121] R. Cheng, N. Sang, Y. Zhou, and X. Wang, "Universal adversarial attack against 3d object tracking," in 2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys). IEEE, 2021, pp. 34–40.
[122] "A 3d version of the mnist database of handwritten digits," https://www.kaggle.com/datasets/daavoo/3d-mnist, accessed: 2019-01-30.
[123] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," 2017.
[124] A. V. Phan, M. Le Nguyen, Y. L. H. Nguyen, and L. T. Bui, "Dgcnn: A convolutional neural network over large-scale labeled graphs," Neural Networks, vol. 108, pp. 533–543, 2018.
[125] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3d point clouds," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9621–9630.
[126] Y. Liu, B. Fan, S. Xiang, and C. Pan, "Relation-shape convolutional neural network for point cloud analysis," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 8895–8904.
[127] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2015, pp. 922–928.
[128] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao, "Spidercnn: Deep learning on point sets with parameterized convolutional filters," in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 87–102.
[129] X. Yan, C. Zheng, Z. Li, S. Wang, and S. Cui, "Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5589–5598.
[130] T. Xiang, C. Zhang, Y. Song, J. Yu, and W. Cai, "Walk in the cloud: Learning curves for point clouds shape analysis," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 915–924.
[131] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "A papier-mâché approach to learning 3d surface generation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 216–224.
[132] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 16 259–16 268.
[133] X. Ma, C. Qin, H. You, H. Ran, and Y. Fu, "Rethinking network design and local geometry in point cloud: A simple residual mlp framework," arXiv preprint arXiv:2202.07123, 2022.