IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 1 December 2023, accepted 17 December 2023, date of publication 19 December 2023,
date of current version 26 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3345000

SURVEY

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

HANIEH NADERI[1], (Student Member, IEEE), AND IVAN V. BAJIĆ[2], (Senior Member, IEEE)
[1]Department of Computer Engineering, Sharif University of Technology, Tehran 14588-89694, Iran
[2]School of Engineering Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada

Corresponding author: Ivan V. Bajić (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 has become 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 adversarial example generation methods in recent years. Additionally, it provides an overview of defense strategies, organized into data-focused and model-focused methods. Finally, it presents several current challenges and potential 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], [4], semantic analysis [5], [6], speech recognition [7], [8] and natural language processing [9] in recent years.

Despite the tremendous success of DL, in 2013 Szegedy et al. [10] 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 [11] or they can be more significant changes to the structure of the image [12].

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 [13], self-driving cars [14], drones [15], robotics [16], 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 [17], meshes [18], and point clouds [19]. 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,

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

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, ML, and AI systems. For example, Akhtar et al. [20] focus on adversarial attacks in computer vision, with a particular emphasis on image and video recognition systems. Yuan et al. [21] delve into both adversarial attacks and defense mechanisms within the domain of images. Qiu et al. [22] provide a comprehensive review of adversarial attacks and defenses in various AI domains, including image, video, and text. Wei et al. [23] survey both attacks and defenses against physical 2D objects. Zhai et al. [24] explore adversarial attacks and defenses within the context of graph-based data. Bountakas et al. [25] review domain-agnostic defense strategies across multiple domains, including audio, cybersecurity, natural language processing (NLP), and computer vision. Pavlitska et al. [26] focus on adversarial attacks within the specific domain of traffic sign recognition, which is relevant to autonomous vehicles and road safety. These and several other surveys of adversarial attacks and defenses in various domains have been summarized in Table 1. As seen in the table, there is a lack of surveys focused specifically on 3D point cloud attacks and defenses. Some published surveys do mention 3D attacks and defenses briefly, for example [27], but there is a need for more comprehensive surveys that delve deeper into this topic.

While our survey is focused on adversarial attacks and defenses on 3D point cloud classification, it is important to mention that there are existing general surveys on point cloud analysis and processing, which are not focused on adversarial attacks and defenses. For example, Guo et al. [28] provide a comprehensive overview of deep learning methods for point cloud analysis, including classification, detection, and segmentation. Xiao et al. [29] concentrate on unsupervised point cloud analysis. Xie et al. [30] and Xie et al. [30] specifically address point cloud segmentation tasks. Zhang et al. [31] focus on point cloud classification. Fernandes et al. [32] discuss point cloud processing in specialized tasks like self-driving, while Krawczyk et al. [33] tackle full human body geometry segmentation. Cao et al. [34] explore compression methods for 3D point clouds, essential for handling large data volumes. Although the focus of our survey is on 3D point cloud attacks and defenses, there is an intersection with some of the aforementioned surveys, especially in terms of models and datasets used for point cloud classification. We review the models and datasets that are relevant to the area of adversarial attacks and defenses, which can also be valuable resources for the broader community working on point cloud analysis and processing.

Our key contributions are as follows:

- A review of the different types of adversarial attacks on point clouds that have been proposed, including their methodologies and attributes, with specific examples from the literature.
- A review of the various methods that have been proposed for defending against adversarial attacks, organized into data-focused and model-focused methods, with examples from the literature.
- A summary of the most important datasets and models used by researchers in this field.
- An overview of the challenges and limitations of the current approaches to adversarial attacks and defenses

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

Survey	Application Domain	Focus on	Year
Akhtar *et al.* [20]	Computer Vision (Image & Video)	Attack	2018
Yuan *et al.* [21]	Image	Attack & defense	2018
Qiu *et al.* [22]	AI (Image & Video & Text)	Attack & defense	2019
Wiyatno *et al.* [35]	ML (Image)	Attack	2019
Xu *et al.* [36]	Image & Graph & Text	Attack & defense	2020
Martins *et al.* [37]	Cybersecurity	Attack	2020
Chakraborty *et al.* [38]	Image & Video	Attack & defense	2021
Rosenberg *et al.* [39]	Cybersecurity	Attack & defense	2021
Akhtar *et al.* [27]	Computer Vision (Image & Video)	Attack & defense	2021
Michel *et al.* [40]	Image	Attack & defense	2022
Tan *et al.* [41]	Audio	Attack & defense	2022
Qiu *et al.* [42]	Text	Attack & defense	2022
Liang *et al.* [43]	Image	Attack & defense	2022
Li *et al.* [44]	Image	Attack & defense	2022
Gupta *et al.* [45]	AI (All)	Attack & defense	2022
Wei *et al.* [46]	Physical 2D object	Attack & defense	2022
Wei *et al.* [23]	Physical 2D object	Attack	2022
Mi *et al.* [47]	Object	Attack	2022
Khamaiseh *et al.* [48]	Image	Attack & defense	2022
Pavlitska *et al.* [26]	Image (Traffic Sign Recognition)	Attack	2023
Kotyan *et al.* [49]	ML	Attack	2023
Zhai *et al.* [24]	Graph	Attack & defense	2023
Baniecki *et al.* [50]	AI (Image)	Attack & defense	2023
Han *et al.* [51]	Image	Attack	2023
Bountakas *et al.* [25]	Audio, cyber-security, NLP, & computer vision	Defense	2023

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 datasets for point cloud classification and present an overview of datasets and victim models used in the area of adversarial attacks and defenses on point clouds. We discuss current challenges and potential future directions 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 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 knowledge of 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.
- **Gray-box attacks** cover the spectrum between the extremes of black- and white-box attacks. Here, the attacker knows partial details about the DL model's architecture and parameters in addition to having access to its input and output.

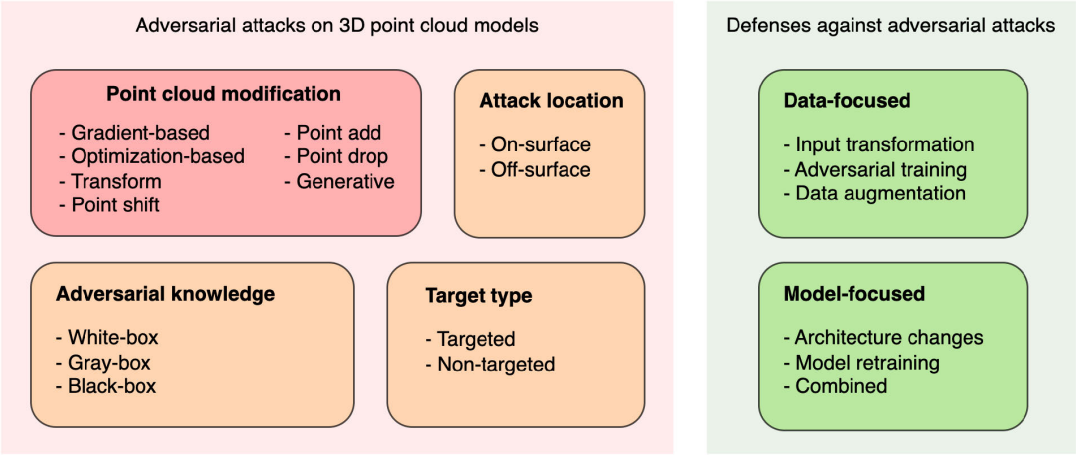

FIGURE 1. Categorization of adversarial attack and defense approaches on 3D point clouds.

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
$\text{sign}(\cdot)$	Sign function
P	Parameter of the ℓ_P-norm; typical values of P are $1, 2$ and ∞.
λ	Controls the trade-off between the two terms in the objective function
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
$g(\cdot)$	Penalty function
$S(\cdot)$	Statistical Outlier Removal (SOR) defense
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

- **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.
- **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 loss function corresponding to each input point are used to generate an adversarial point cloud with a 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.
- **Off-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 the points of $\mathcal{P}$, creating an adversarial point cloud $\mathcal{P}^{adv}$, which could fool a DL model to produce 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, **ℓ_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, $\mathcal{P} = \{p_i \,|\, i = 1, 2, \ldots, n\}$ 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 $\eta = (\eta_1, \eta_2, \ldots, \eta_n)$, $\eta_i \in \mathbb{R}^3$, to $\mathcal{P}$. The three common ℓ_P norms have the following interpretations:

- **ℓ_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 $\mathcal{P}$.
- **ℓ_2-norm** or $\|\eta\|_2$ is the Euclidean distance between $\mathcal{P}^{adv}$ and $\mathcal{P}$.
- **ℓ_∞-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 dissimilarity 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:

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

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

In addition to the distance measures mentioned above, there are other distance measures for point clouds, such as the point-to-plane distance [52], which are used in point cloud compression. However, these are not commonly encountered in the literature on 3D adversarial attacks, so we do not review them here.

III. ADVERSARIAL ATTACKS

Various techniques have been proposed to generate adversarial attacks on 3D point cloud models. This section presents a classification of these attacks based on several criteria, illustrated in Fig. 1. While different classifications are possible, ours is based on attack methodologies, such as gradient-based point cloud modification, etc., and attack attributes such as attack location (on-/off-surface), adversarial knowledge (white-box, gray-box, or black-box) and target type (targeted or non-targeted). In the following, we first present various methodologies, each with specific examples of the attack methods from that category. The discussion of various attack attributes is provided later in the section. The most popular attack approaches are also summarized in Table 3 for quick reference.

A. POINT CLOUD MODIFICATION STRATEGIES

1) GRADIENT-BASED STRATEGIES

DNNs are typically trained using the gradient descent method to minimize a specified loss function. Attackers targeting such models often take advantage of the fact that they can achieve their goals by maximizing this loss function along the gradient ascent direction. Specifically, attackers can create adversarial perturbations utilizing the gradient information of the model and iteratively adjusting the input to maximize the loss function. Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD) are the most commonly used gradient-based techniques for this purpose. We review each of them below.

a: 3D FAST GRADIENT SIGN METHOD (3D FGSM)

The inception of adversarial attacks on 3D data occurred in 2019 using gradient-based techniques. During this period, Liu et al. [60] and Yang et al. [66] extended the Fast Gradient Sign Method (FGSM), originally proposed by Goodfellow et al. [69], to 3D data. The 3D version of FGSM

TABLE 3. Popular adversarial attacks.

Reference	Attack Name	Targeted / Non-targeted	Shift / Add / Drop / Transform	On- / Off-surface	Optimized / Gradient	Black- / Gray- / White-box
Xiang *et al.* [53]	Perturbation	Targeted	Shift	Off	Optimized	White
	Independent points	Targeted	Add	Off	Optimized	White
	Clusters	Targeted	Add	Off	Optimized	White
	Objects	Targeted	Add	Off	Optimized	White
Zheng *et al.* [54]	Drop100	Non-Targeted	Drop	On	Gradient	White
	Drop200	Non-Targeted	Drop	On	Gradient	White
Hamdi *et al.* [55]	Advpc	Targeted	Transform	On	Optimized	White
Lee *et al.* [56]	ShapeAdv	Targeted	Shift	On	Optimized	White
Zhou *et al.* [57]	LG-GAN	Targeted	Transform	On	-	White
Wen *et al.* [58]	GeoA^3	Targeted	Shift	On	Optimized	White
Tsai *et al.* [59]	KNN	Targeted	Shift	On	Optimized	White
Liu *et al.* [60]	Extended FGSM	Non-Targeted	Shift	Off	Gradient	White
Arya *et al.* [61]	VSA	Non-Targeted	Add	On	Optimized	White
Liu *et al.* [62]	Distributional attack	Non-Targeted	Shift	On	Gradient	White
	Perturbation resampling	Non-Targeted	Add	Off	Gradient	White
	Adversarial sticks	Non-Targeted	Add	Off	Gradient	White
	Adversarial sinks	Non-Targeted	Add	Off	Gradient	White
Kim *et al.* [63]	Minimal	Non-Targeted	Shift	Off	Optimized	White
	Minimal	Non-Targeted	Add	Off	Optimized	White
Ma *et al.* [64]	JGBA	Targeted	Shift	On	Optimized	White
Liu *et al.* [65]	ITA	Targeted	Shift	On	Optimized	Black
Liu *et al.* [66]	FGSM	Non-Targeted	Shift	Off	Gradient	White
	MPG	Non-Targeted	Shift	Off	Gradient	White
	Point-attachment	Non-Targeted	Add	Off	Gradient	White
	Point-detachment	Non-Targeted	Drop	On	Gradient	White
Wicker *et al.* [67]	—	Both	Drop	On	Optimized	Both
He *et al.* [68]	—	Non-Targeted	Shift	On	Optimized	White

adds an adversarial perturbation η to each point in the given point cloud $\mathcal{P}$ to create an adversarial point cloud $\mathcal{P}^{adv} = \mathcal{P} + \eta$. Perturbations are generated according to the direction of the sign of the gradient at each point. The perturbation can be expressed as

$$\eta = \epsilon \cdot \text{sign}\left[\nabla_{\mathcal{P}} J(f(\mathcal{P}; \theta), Y)\right] \tag{4}$$

where f is the deep model that is parameterized by θ and takes an input point cloud $\mathcal{P}$, and Y denotes the label associated with $\mathcal{P}$. J is the loss function, $\nabla_{\mathcal{P}} J$ is its gradient with respect to $\mathcal{P}$ and sign$(\cdot)$ denotes the sign function. The ϵ value is an adjustable hyperparameter that determines the ℓ_p-norm of the difference between the original and adversarial inputs.

Liu et al. [60] introduced three different ways to define ϵ as a constraint for η as follows

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

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

Yang et al. [66] used the Chamfer distance (instead of the ℓ_2-norm) between the original point cloud and the adversarial counterpart to extend the FGSM to 3D. 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 authors in [66] set it to 0.02.

Apart from the FGSM attack, Yang et al. [66] introduced another attack called Momentum-Enhanced Pointwise Gradient (**MPG**). The (3D) MPG attack, similar to its 2D version [70], integrates momentum into the iterative FGSM. The MPG attack produces more transferable adversarial examples because the integration of momentum into the iterative FGSM process enhances its ability to escape local minima and generate effective perturbations.

b: 3D PROJECTED GRADIENT DECENT (3D PGD)

One of the most potent attacks on 3D data is the Projected Gradient Descent (PGD), whose foundation is the pioneering work by Madry et al. [71]. The iterative FGSM is considered

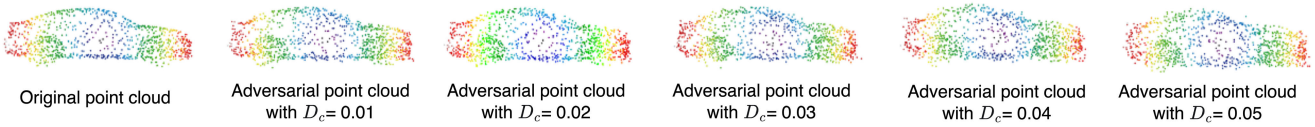

FIGURE 2. An example of original point cloud and 3D FGSM adversarial counterpart [66] with Chamfer distances D_c varying from 0.01 to 0.05. (Image source: [66]; use permitted under the Creative Commons Attribution License CC BY 4.0.)

a basis for PGD. Taking the iterative FGSM method, we can generate the adversarial point cloud as

$$\mathcal{P}_0^{adv} = \mathcal{P},$$
$$\mathcal{P}_{t+1}^{adv} = \text{clip}_{\mathcal{P},\epsilon}\left[\mathcal{P}_t^{adv} + \alpha \cdot \text{sign}(\nabla_{\mathcal{P}} J(f(\mathcal{P};\theta), Y)\right], \quad (5)$$

where t is the iteration number and $\text{clip}_{\mathcal{P},\epsilon}[\cdot]$ limits the change of the generated adversarial input to be within ϵ distance of $\mathcal{P}$, according to a chosen distance measure.

The PGD attack is based on increasing the cost of the correct class Y, without specifying which of the incorrect classes the model should select. To do this, the PGD attack finds the perturbation η that maximizes the loss function under the perturbation constraint controlled by ϵ. The optimization problem can be formulated as:

$$\max_{\eta} \quad J(f(\mathcal{P}+\eta;\theta), Y)$$
$$\text{such that} \quad D(\mathcal{P}, \mathcal{P}+\eta) \leq \epsilon \quad (6)$$

where J is the loss function and ϵ controls how far the adversarial point cloud can be from the original one according to the chosen distance measure D.

Liu et al. [62] proposed the following four flavors of the PGD attack.

1) **Perturbation resampling** This attack resamples a 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. Hausdorff distance is used to maintain the similarity between $\mathcal{P}$ and $\mathcal{P}^{adv}$.
2) **Adding adversarial sticks** During this attack, the algorithm adds four sticks to the point cloud, such that one end is attached to the point cloud while the other end is a small distance away. The algorithm optimizes the two ends of the sticks so that the label of the point cloud is changed.
3) **Adding adversarial sinks** In this case, critical points (the points remaining after max pooling in PointNet) are selected as "sink" points, which pull the other points towards them until the point cloud label is changed. The goal is to minimize global changes to non-critical points. The distance measure used to maintain the similarity between $\mathcal{P}$ and $\mathcal{P}^{adv}$ is ℓ_2-norm.
4) **Distributional attack** This attack uses the Hausdorff distance between the adversarial point cloud and the triangular mesh fitted over the original point cloud, to push adversarial points towards the triangular mesh. This method is less sensitive to the density of points in $\mathcal{P}$ because it uses a mesh instead of the point cloud itself to measure the perturbation. Figure 3 shows two examples of adversarial point clouds generated by the distributional attack.

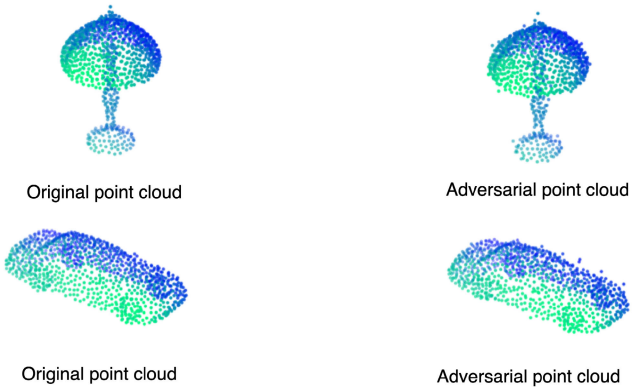

FIGURE 3. Two examples of the original point clouds (left) and adversarial point clouds generated by the distributional attack (right). [62] (Image source: [62]; use permitted under the Creative Commons Attribution License CC BY 4.0.)

Ma et al. [64] proposed the Joint Gradient Based Attack (**JGBA**). They added an extra term to the objective function of the PGD attack (6) to defeat statistical outlier removal (SOR), a common defense against attacks. Specifically, their optimization problem is:

$$\max_{\eta} \quad J(f(\mathcal{P}+\eta;\theta), Y) + \lambda \cdot J(f(S(\mathcal{P}+\eta);\theta), Y)$$
$$\text{such that} \quad D_{\ell_2}(\mathcal{P}, \mathcal{P}+\eta) \leq \epsilon \quad (7)$$

where $S(\cdot)$ denotes SOR and λ is a hyperparameter that controls the trade-off between the two terms in the objective function. This way, the adversarial point cloud becomes more resistant against the SOR defense.

Kim et al. [63] proposed a so-called **minimal attack** that aims to manipulate a minimal number of points in a point cloud. This can be thought of as minimizing $D_{\ell_0}(\mathcal{P}, \mathcal{P}^{adv})$. To find an adversarial point cloud, Kim et al. modify the loss function of the PGD attack (8) by adding a term that tries to keep the number of changed points to a minimum. Furthermore, they used Hausdorff and Chamfer distances to preserve the similarity between $\mathcal{P}$ and $\mathcal{P}^{adv}$. Figure 4 illustrates examples of minimal adversarial attack, where the altered points are indicated in red.

2) OPTIMIZATION-BASED STRATEGIES

While gradient-based strategies rely on model gradients, optimization-based methods instead utilize model output

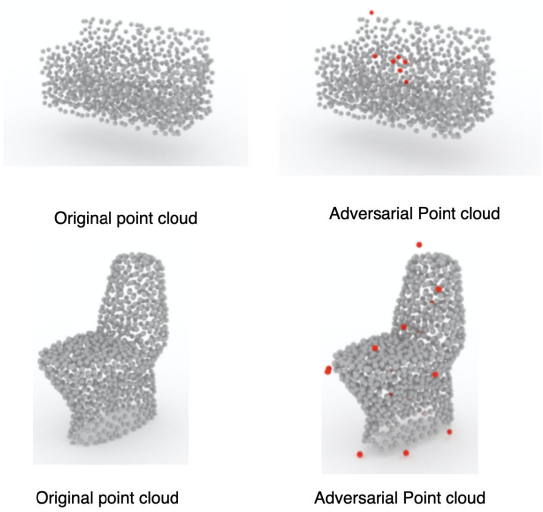

FIGURE 4. Two examples of the original point cloud and the corresponding minimal adversarial attack, where the altered points are shown in red [63] (©2021 IEEE. Reprinted, with permission, from [63]).

logits to create attacks. These methods usually aim to keep perturbations minimal, to reduce the chance of detecting the attack, while deceiving the model into making a wrong decision. Hence, these are often formulated as constrained optimization problems or multi-objective problems. The Carlini and Wagner (C&W) attack is founded on these ideas, as explained below.

a: 3D CARLINI AND WAGNER ATTACK (3D C&W)

The 3D version the C&W attack was developed by Xiang et al. [53] as an extension of the original work by Carlini and Wagner [72]. The method can be described as an optimization problem of finding the minimum perturbation η such that the output of the deep model to the adversarial input $\mathcal{P}^{adv} = \mathcal{P} + \eta$ is changed to the target label T. The problem can be formulated as

$$\min_{\eta} \quad D(\mathcal{P}, \mathcal{P} + \eta) + c \cdot g(\mathcal{P} + \eta) \tag{8}$$

where $D(\cdot, \cdot)$ is the distance measure, c is a Lagrange multiplier and $g(\cdot)$ is a penalty function such that $g(\mathcal{P}^{adv}) \leq 0$ if and only if the output of the deep model is $f(\mathcal{P}^{adv}) = T$. Seven choices for g were listed in [72]. One of the functions evaluated in their experiments, which became popular in the subsequent literature, is as follows:

$$g(\mathcal{P}^{adv}) = \max\left[\max_{i=t}(Z(\mathcal{P}^{adv})_i) - Z(\mathcal{P}^{adv})_t, -\kappa\right] \tag{9}$$

where Z denotes the Softmax function, and κ represents a constant that controls confidence. Compared to the FGSM attack, the C&W attack does not constrain the perturbation; instead, it searches for the minimal perturbation that would produce the target label.

Xiang et al. [53] developed four versions of the 3D C&W attack, featuring various distance measures:

1) **Adversarial perturbation** to shift the points toward the point cloud's surface, using the ℓ_2-norm between all points of $\mathcal{P}$ and $\mathcal{P}^{adv}$ as the distance measure.
2) **Adding adversarial independent points** by using two distance measures – Hausdorff distance and Chamfer distance – between $\mathcal{P}$ and $\mathcal{P}^{adv}$, to push independent points toward the point cloud's surface.
3) **Adding adversarial clusters** based on three principles. (1) Chamfer distance between the original point cloud and the adversarial cluster is used to push clusters toward the point cloud's surface. (2) Only a small number of clusters is added, specifically one to three. (3) 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** based on three principles. (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) Only a small number of objects is added, specifically one to three. (3) The ℓ_2-norm between a real-world object and an adversarial object is used to generate shapes similar to those in the real world.

Wen et al. [58] considered a new distance measure named *consistency of local curvatures* to guide perturbed points towards object surfaces. Adopting the C&W attack framework, the authors use a combination of the Chamfer distance, Hausdorff distance, and local curvature consistency as the distance measure to create a geometry-aware adversarial attack (**GeoA^3**). The GeoA^3 attack enforces the smoothness of the adversarial point cloud to make the difference between it and the original point cloud imperceptible to the human eye. Finally, Zhang et al. [73] introduced a **Mesh Attack** designed to perturb 3D object meshes while minimizing perceptible changes. The Mesh Attack employs two key components in its loss function: a C&W loss, encouraging misclassification of adversarial point clouds, and a set of mesh losses, including Chamfer, Laplacian, and Edge Length losses, to maintain the smoothness and geometric fidelity of the adversarial meshes relative to the original input mesh.

3) TRANSFORM ATTACKS

Transform attacks are crafted in the transform domain rather than the input domain. Usually, the 3D point cloud is transformed into another domain (e.g., 3D frequency domain), then modified, and then transferred back to the original input domain to be fed to the classifier. Liu et al. [74] have suggested an adversarial attack based on the frequency domain, which aims to improve the transferability of generated adversarial examples to other classifiers. They transformed the point cloud into the frequency domain using the graph Fourier transform (GFT) [75], then divided it into low- and high-frequency components, and applied perturbations to the low-frequency components to create an adversarial point cloud. Liu et al. [76] investigated the geometric structure of point clouds by perturbing, in turn, low-, mid-, and high-frequency components. They found that

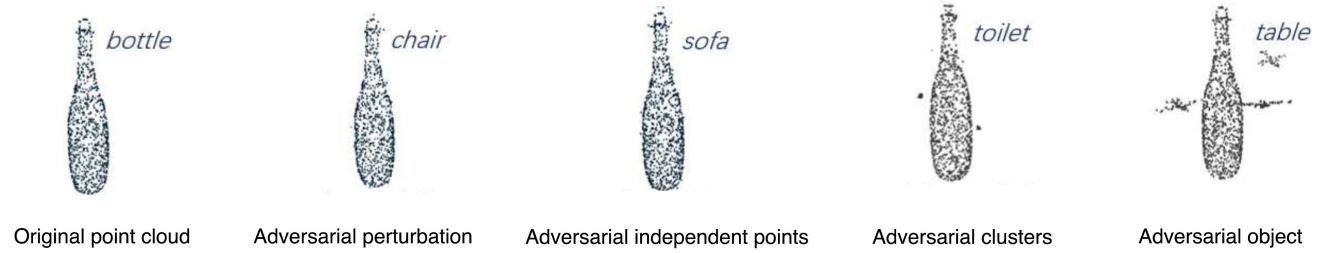

FIGURE 5. Left to right: original point cloud and the adversarial examples produced by the attacks proposed in [53] (©2019 IEEE. Reprinted, with permission, from [53]).

perturbing low-frequency components significantly changed their shape. To preserve the shape, they created an adversarial point cloud with constraints on the low-frequency perturbations and instead guided perturbations to the high-frequency components. Hu et al. [77] suggest that by analyzing the eigenvalues and eigenvectors of the graph Laplacian matrix [75] of a point cloud, one can determine which areas of the cloud are particularly sensitive to perturbations. By focusing on these areas, the attack can be crafted more effectively.

A related attack, though not exactly in the frequency domain, was proposed by Huang et al. [78]. This attack is based on applying reversible coordinate transformations to points in the original point cloud, which reduces one degree of freedom and limits their movement to 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 moved to generate the adversarial point cloud.

In another attack called Variable Step-size Attack (**VSA**) [61], a hard constraint on the number of modified points is incorporated into the optimization function of a PGD attack (8) to try to preserve the point cloud's appearance. Specifically, points with the highest gradient norms, which are thought to have the greatest impact on classification, are selected initially. The selected points are then subject to adversarial perturbations. The goal is to shift these points in a way that maintains their original appearance while maximizing the loss function, thus causing the model to misclassify the input. By controlling the step size, VSA adjusts the magnitude of perturbations applied to the selected points. It starts with a larger step size to allow for rapid exploration of the optimization landscape. As the process advances, the step size is progressively reduced to guide the optimization toward more precise modifications.

4) POINT SHIFT ATTACKS

Point shift attacks involve shifting the points of the original 3D point cloud to fool the deep model, while the number of points remains the same. Tsai et al. [59] developed a shifting point attack called K-Nearest Neighbor (**KNN**) attack that limits distances between adjacent points by adding another loss term to (8). This additional loss term is based on the K-Nearest Neighbor distance for each point, while their main distance term in (8) is the Chamfer distance. Miao et al. [79] proposed an adversarial point cloud based on rotation by applying an isometry matrix to the original point cloud. To find an appropriate isometry matrix, the authors used the Thompson Sampling method [80], which can quickly find a suitable isometry matrix with a high attack rate.

Liu et al. [65] 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. Although some of the attacks described earlier also involve shifting points, the main difference here is that ITA aims to make the attack imperceptible, whereas earlier attacks may cause noticeable changes to the shape. Along the same lines, Tang et al. [81] presented a method called **NormalAttack** for generating imperceptible point cloud attacks. Their method deforms objects along their normals by considering the object's curvature to make the modification less noticeable.

Zhao et al. [82] proposed a class of point cloud perturbation attacks called Nudge attacks that try to minimize point perturbation while changing the classifier's decision. They generated adversarial point clouds using gradient-based and genetic algorithms with perturbations of up to 150 points to deceive the classifier. In some cases, the attack can fool the classifier by changing a single point when the point has a large distance from the surface of the objects. Analogously to the one-pixel attack for images [83], Tan et al. [84] proposed an attack called **One point attack** in which only a single point in the point cloud needs to be shifted in order to fool the deep model. The authors also present a method to identify the most important points in the point cloud based on a saliency map, which could be used as candidates for the attack.

5) POINT ADD ATTACKS

Point add attacks involve the addition of points to the point cloud with the aim of misleading deep models, while remaining plausible. Obviously, the number of points in the point cloud increases after this attack. Yang et al. [66] provided a point-attachment attack by attaching a few points to the point cloud. Chamfer distance is used to keep the distance between the newly added points and the original point cloud small. Hard constraints limit the number of points added in the point cloud, making the adversarial point cloud preserve the appearance of the original one.

Shape Prior Guided Attack [85] is a method that adds points by using a shape prior, or prior knowledge of the structure of the object, to guide the generation of the perturbations. This method introduces Spatial Feature Aggregation (SPGA), which divides a point cloud into sub-groups and introduces structure sparsity to generate adversarial point sets. It employs a distortion function comprising Target Loss and Logical Structure Loss to guide the attack. The Shape Prior Guided Attack is optimized using the Fast Optimization for Attacking (FOFA) algorithm, which efficiently finds spatially sparse adversarial points. The goal of this method is to create adversarial point clouds that have minimal perturbations while still being able to fool the target classification model.

Note that some of the attack approaches described earlier also involve addition of points and can be considered to be point add attacks. For example, Liu et al. [62] present several attacks such as **Perturbation resampling**, **Adding adversarial sticks** and **Adding adversarial sinks**, which can be considered point add attacks. These attacks were explained in more detail in Section III-A1b.

6) POINT DROP ATTACKS

Attacks described in the previous sections involved adding or shifting points in the point domain or transform (latent) domain. This section reviews attacks that instead remove (drop) points from the point cloud to generate the adversarial point cloud. Obviously, the number of points in a point cloud reduces after a point drop attack. The points that are selected to be dropped are often referred to as ''critical'' points, in the sense that they are expected to be critical for a classifier to make the correct decision. Various methods have been developed to identify critical points in a point cloud.

For example, Zheng et al. [54] developed a method that uses a saliency map [86] to find critical points and drop them. As an illustration, critical points identified by high saliency values [86] are illustrated in red in Figure 6. The figure also shows what happens when these points are dropped. A version of this attack exists where, instead of dropping high-saliency points, they are shifted towards the point cloud center, thereby altering the shape of the point cloud in a manner similar to dropping points. Two versions of this attack have become popular in the literature, **Drop100** and **Drop200**, which drop 100 and 200 points, respectively.

An attack described in [87] identifies ''adversarial drop points'' in a 3D point cloud that, when dropped, significantly reduce a classifier's accuracy. These points are specified by analyzing and combining fourteen-point cloud features, independently of a target classifier that is to be fooled. In this way, the attack becomes more transferable to different classifier models.

In [67], the critical points are randomly selected and checked for dropping one by one. If dropping a point increases the probability of changing the ground-truth label $f(\mathcal{P}) = Y$, such point is considered a critical point and will be dropped. Otherwise, it will not be dropped. This procedure continues iteratively until the classifier's decision is changed. The process can be described as the following optimization problem

$$\min_{\mathcal{P}^{adv} \subseteq \mathcal{P}} \quad (|\mathcal{P}| - |\mathcal{P}^{adv}|) \quad \text{such that} \quad f(\mathcal{P}^{adv}) \neq f(\mathcal{P}) \tag{10}$$

where $|\mathcal{P}|$ and $|\mathcal{P}^{adv}|$ are the number of points in the original and adversarial point cloud, respectively.

In order to determine the level of influence of a given point in PointNet decision-making, Yang et al. [66] introduced a **Point-detachment attack** that assigned a *class-dependent importance* to each point. A greedy strategy was employed to generate an adversarial point cloud, in which the most important points dependent on the ground-truth label are dropped iteratively. The class-dependent importance associated with a given point was determined by multiplying two terms. The first term used the PointNet feature matrix before max-pooling aggregation and the second term used the gradients relative to the ground-truth label output. The combination of these terms helped determine which points had the highest impact on the PointNet decision.

7) GENERATIVE STRATEGIES

Generative approaches utilize models such as Generative Adversarial Networks (GANs) and variational autoencoder models to create adversarial point clouds. Most of these attacks [56], [57], [88], [89] attempt to change the shape of the point cloud in order to fool the deep model. The concept of these attacks can be related to what is called unrestricted attacks in 2D images [90], [91], [92]. When such attacks occur, the input data might change significantly while remaining plausible. These attacks can fool the classifier without making humans confused. In this regard, Lee et al. [56] proposed shape-aware adversarial attacks called **ShapeAdv** that are based on injecting an adversarial perturbation η into the latent space z of a point cloud autoencoder. Specifically, 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. Lee et al. [56] proposed three attacks with varying distance measures, which are used as a term in the C&W loss to maintain similarity between the original and adversarial point clouds. All three attacks calculate the gradient of the C&W loss with respect to the adversarial perturbation of the latent representation z. The three attacks are as follows:

1) **Shape-aware attack in the latent space.** Here, the goal is to minimize the ℓ_2-distance between the latent representation z and the perturbed representation $z + \eta$. Using this approach, the original and adversarial point clouds are close in the latent space, but they could be highly dissimilar in the point space.
2) **Shape-aware attack in the point space.** In this case, Chamfer distance is used to encourage the similarity of the original and adversarial point cloud in the point space. This is an attempt to resolve the issues with the previous attack, where the original and adversarial point cloud could be very different in the point space.

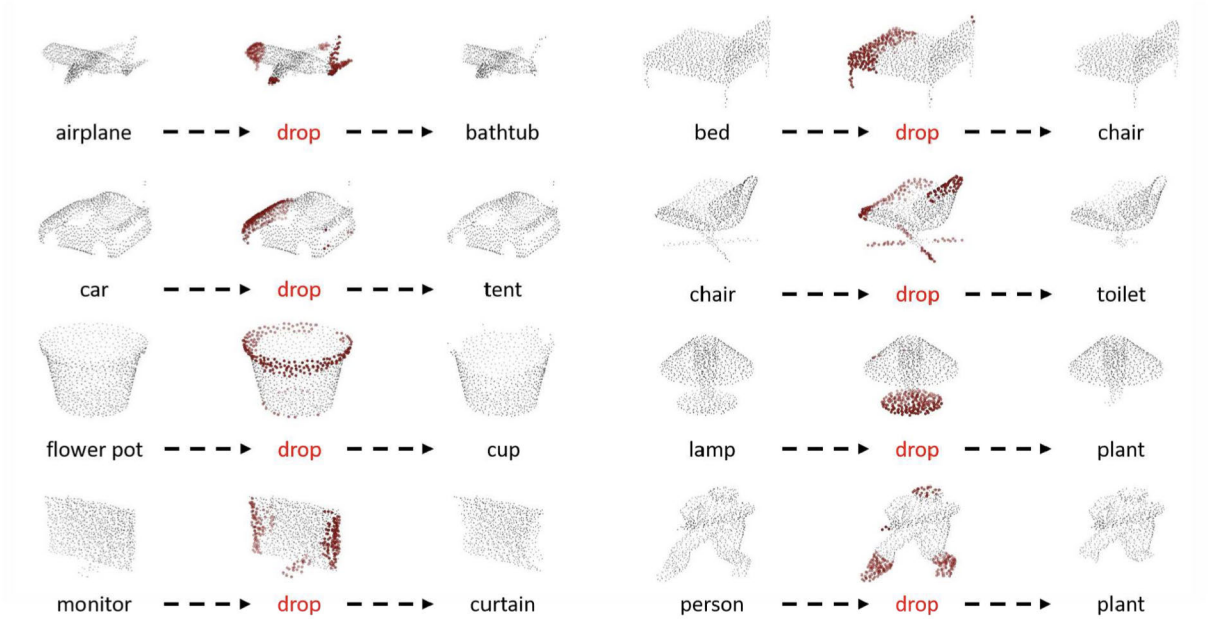

FIGURE 6. Original point clouds with labels (left), dropped points in red associated with highest scores (middle), and adversarial point clouds with new labels (right) [54] (©2019 IEEE. Reprinted, with permission, from [54]).

3) **Shape-aware attack with auxiliary point clouds.** This attack minimizes the Chamfer distance between the adversarial point cloud and an auxiliary point cloud, which is created as the average of k nearest neighbors sampled from the same class as the original point cloud. The goal is to avoid large adversarial perturbations in any direction in the latent space. To guide this process, point clouds sampled from the class of the original point cloud are used.

Hamdi et al. [55] proposed an attack called **Advpc** by using an autoencoder that could be transferred between classification networks. The autoencoder was trained using a combination of two loss functions: the C&W loss when the adversarial point cloud is fed directly to the classifier, and the C&W loss when the point cloud is first fed to the autoencoder to project a perturbed point cloud onto the natural input manifold, then reconstructed, and then fed to the classifier. This strategy improved the transferability of the attack to different classification networks.

Tang et al. [88] proposed a deep **manifold attack** that deforms the intrinsic 2-manifold structures of 3D point clouds. The attack strategy comprises two steps. In the first step, an autoencoder is used to establish a representation of the mapping between a 2D parameter plane and the underlying 2-manifold surface of the point cloud. This representation serves as a basis for subsequent transformations. The second step involves learning stretching operations within the 2D parameter plane. This stretching produces a 3D point cloud that can fool a pretrained classifier, all while keeping geometric distortion minimal.

LG-GAN attack [57] generates an adversarial point cloud based on a Generative Adversarial Network (GAN).

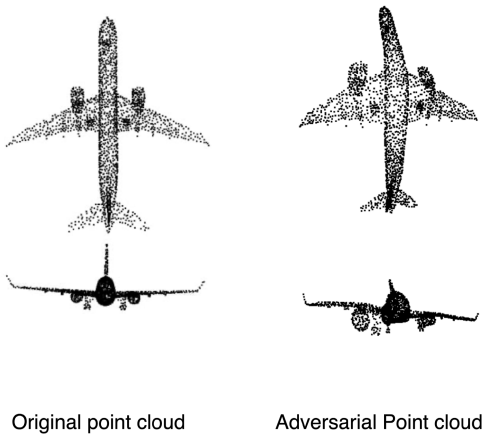

FIGURE 7. An example of original point cloud and LG-GAN attack were proposed in [57] (©2020 IEEE. Reprinted, with permission, from [57]).

The GAN is trained using the original point clouds and target labels to learn how to generate adversarial point clouds to fool a classifier. It extracts hierarchical features from original point clouds, then integrates the specified label information into multiple intermediate features using the label encoder. The encoded features are fed into a reconstruction decoder to generate the adversarial point cloud. Once the GAN is trained, the attack is very fast because it only takes one forward pass to generate the adversarial point cloud. Figure 7 shows an instance of the LG-GAN attack.

Dai et al. [89] proposed another GAN-based attack, where the input to the GAN is noise, rather than the original point cloud. The noise vector and the target label are fed into a graph convolutional GAN, which outputs the

generated adversarial point cloud. The GAN is trained using a four-part loss function including the objective loss, the discriminative loss, the outlier loss, and the uniform loss. The objective loss encourages the victim network to assign the (incorrect) target label to the adversarial point cloud while the discriminative loss encourages an auxiliary network to classify the adversarial point cloud correctly. The outlier loss and the uniform loss encourage the generator to preserve the point cloud shape. Besides these GAN-based attacks, Lang et al. [93] proposed an attack that alters the reconstructed geometry of a 3D point cloud using an autoencoder trained on semantic shape classes, while Mariani et al. [94] proposed a method for creating adversarial attacks on surfaces embedded in 3D space, under weak smoothness assumptions on the perceptibility of the attack.

B. ATTACK LOCATION

The location of perturbations plays a crucial role in changing the shape and distribution of points within a point cloud. This can result in points being shifted either off the object's surface, introducing noise, or along the surface, thereby altering the distribution of points. Hence, in terms of the location of the perturbations, attacks can be categorized into two groups: on-surface and off-surface.

On-surface perturbation attacks are those attacks in which the points of the adversarial cloud $\mathcal{P}^{adv}$ are located along the object's original surface. Notably, drop attacks [54], [66], [67] are an example of such attacks, since drop attacks involve solely the removal of points from the point cloud, so the remaining points stay on the original surface. While other attack methods like point shift or transform would normally tend to move the points off the object's surface, various approaches can be employed to keep the points at or near the original surface. For example, Hamdi et al. [55] employ an autoencoder that projects off-surface perturbations onto the natural input manifold, thereby minimizing the movements of points off the surface. Another example is provided by Tsai et al. [59], who developed the KNN attack. This approach introduces constraints on the distances between adjacent points by adding an extra loss term based on the K-Nearest Neighbor distances for each point, with the goal of keeping the perturbations on the original surface. In the VSA attack [61], the magnitude of perturbations applied to adversarial points is adjusted by controlling the step size, which again could be used to keep points on the surface. The ''distributional attack'' [62] employs the Hausdorff distance between the adversarial point cloud and the triangular mesh fitted over the original point cloud. In this way, perturbed points can be guided toward the triangular mesh, effectively keeping the perturbed points at or near the object's original surface.

Off-surface perturbation attacks produce adversarial point clouds $\mathcal{P}^{adv}$ that include points off the original object's surface. As noted earlier, many strategies for generating adversarial point clouds include a distance term $D(\mathcal{P}, \mathcal{P}^{adv})$ between the original and adversarial point cloud. This distance term is either involved in a hard constraint (upper bounded by an explicit value) or included as a loss term in the overall loss function. However, if its hard constraint is too high or if its scaling factor in the loss function is too low compared to other terms, this can result in off-surface points. For example, Yang et al. [66] set the upper bound on the Chamfer distance to 0.2, which is somewhat high and could result in off-surface perturbations. In other cases, off-surface perturbations are intentionally created. For example, Liu et al. [62], in their adversarial sticks attack, add four sticks to the point cloud. These sticks are attached at one end to the point cloud and extend a small distance away, thereby creating an off-surface attack. Similarly, Xiang et al. [53] introduce clusters, individual points, or small objects off the surface of the object to craft their attacks. It should be noted that off-surface attacks might be easier to detect since the perturbed cloud's appearance starts deviating more obviously from the original one.

C. ADVERSARIAL KNOWLEDGE

In the context of adversarial knowledge, attacks can be categorized into three classes: white-box, black-box, and gray-box attacks. This classification is based on the extent of the attacker's knowledge about the target model. White-box and black-box scenarios represent extremes, whereas the gray-box scenario covers a wide range of possibilities between these extremes.

White-box attacks are those in which the attacker has complete information about the DL model under attack. This includes knowledge of the model's architecture, parameters, loss function, training details, and input/output training data. In the literature on adversarial attacks on 3D point cloud models, white-box attacks are quite common. Examples in this category include various gradient-based attack methods, such as those by Zhang et al. [54] and Liu et al. [62]. These approaches make use of the gradients of the loss function, propagated back through the model, to construct a variety of attacks such as point shifting, addition, and dropping. Other examples include attacks developed by Xiang et al. [53] and Liu et al. [60], [62], among others.

Black-box attacks are those in which the attacker has limited information – sometimes none – about the target model being attacked. In this case, at most, the attacker has access to the target model as a ''black box,'' meaning that it can generate the output of the model for a given input but lacks knowledge of the model's internal structure, training details, etc. Black-box attacks align more closely with real-world attack scenarios, but they are more difficult to construct.

Black-box attacks are less common in the literature on adversarial attacks on 3D point cloud models. One example of a black-box attack is the ''model-free'' approach of Naderi et al. [87], which does not require any knowledge of the target model and instead focuses on identifying critical points within point clouds. This method takes advantage of the inherent properties of point clouds, bypassing the need for knowledge about the target model, and is therefore applicable to any model. Huang et al. [78] proposed two versions of their

attack, one white-box and the other black-box. The black-box attack relies on queries and saliency maps generated from a separate white-box surrogate model to craft adversarial perturbations that fool the target model.

Wicker and Kwiatkowska in [67] randomly select and test critical points, dropping them if it increases the likelihood of changing the label. This iterative process continues until the model's decision changes. Therefore, the approach requires the ability to input a point cloud into the target model and access the output, but not the internal details of the target model, making it a black-box attack. ITA, by Liu et al. [65], is another approach that could be classified as a black-box attack. It is based on subtly shifting each point in the point cloud along its normal vector, for which the knowledge of the internal architecture of the target model is not needed.

The term **gray-box attack** has appeared in the literature more recently [95]. It is intended to capture various scenarios between the extremes of white-box and black-box attacks. It should be noted that the boundaries of what are considered white- or black-box attacks are not crisp, and some variations in their interpretations do exist. For example, in a white-box scenario, the attacker may have access to all the internal parameters of the target model, but might not use all of them in constructing the attack. Therefore, such an attack can also be classified as a gray-box attack. That being said, so far very few approaches in the literature on 3D point cloud adversarial attacks have been declared gray-box, with notable exceptions in [93], and [96].

D. TARGET TYPE

Some adversarial attacks attempt to guide the DL model towards a specific wrong label, while others simply want the model to produce any wrong label. The choice between these depends on the objectives of the attacker. Depending on the type of the label target, attacks can be classified as targeted or non-targeted.

Targeted attacks are those in which the goal is to make the DL model's output be a specific target label. There are two common approaches for choosing the target label in a targeted attack:

1) Most likely wrong label: Here, the target label is selected to be the one with the highest probability confidence) other than the ground-truth label. The underlying idea is that this may be the easiest wrong label to lead the model towards.
2) Random label: Here, the target label is chosen randomly among the wrong labels. Although it might be harder to lead the model towards a randomly chosen label, such attack may be more impactful, especially in cases where the most likely wrong label is semantically close to the ground truth label (e.g., motorcycle vs. bicycle).

The approach for target selection will depend on the specific objectives in a given scenario. For example, Wu et al. [97] and Naderi et al. [98] have utilized the latter approach, while Ma et al. [64] have explored both strategies in their work.

Non-targeted attacks are those in which the goal is simply to make the DL model misclassify the input, regardless of which wrong label it eventually predicts. Examples of such attacks are those in [54] and [66], which operate by dropping points from the original point cloud until a label change occurs. Further examples of non-targeted attacks include [55], [56], [57], [58]. Interestingly, some studies present attacks that encompass both targeted and non-targeted types, for example [67]. Here, a flexible attack framework is presented where, by encoding appropriate conditions and objectives through a Boolean function, both targeted and non-targeted attacks can be produced.

IV. DEFENSES AGAINST ADVERSARIAL ATTACKS

Adversarial defense methods for 3D point clouds can be data-focused or model-focused, as indicated in Fig. 1. Data-focused strategies involve modifying the data on which the model is trained, or at inference time, in order to defend against attacks. Model-focused strategies may involve changing the model's architecture and/or retraining it to increase its robustness against attacks. Of course, combinations of these strategies are also possible. The following sections discuss defense methods under each of these categories. Moreover, we have provided an overview of the most prevalent defense strategies in Table 4 to simplify navigation and provide quick reference.

A. DATA-FOCUSED STRATEGIES

1) INPUT TRANSFORMATION

An input transformation is a preprocessing step that involves applying some transformation(s) to the input point cloud before it is fed into the deep model. These transformations could be designed to reduce the sensitivity of the deep 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.

a: SIMPLE RANDOM SAMPLING (SRS)

Simple random sampling (**SRS**) [53] is a statistical technique that randomly drops a certain number of points (usually 500) from an input point cloud, with equal probability. It is crude but very fast. Many attacks involve shifting or adding points to a point cloud to cause a deep model to make an error. Random removal of points may remove some of these deliberately altered/inserted points and thereby make it less likely for the model to make an error.

b: STATISTICAL OUTLIER REMOVAL (SOR)

Adversarial attacks that involve adding or shifting points usually result in outliers. Based on this observation, Zhou et al. [99] proposed a defense based on statistical outlier removal (**SOR**). Specifically, their method removed a point in an adversarial point cloud if the average distance of the point to its k nearest neighbors was larger than $\mu + \sigma \cdot \alpha$, where μ is the mean and σ is the standard deviation of the distance of the k nearest neighbors to other points in the point cloud.

TABLE 4. Categorization of defenses against adversarial attacks.

Reference	Defense Name	Data- / Model-focused	Type
Yang *et al.* [66]	SRS	Data	Input transformation
Zhou *et al.* [99]	SOR	Data	Input transformation
Liu *et al.* [60]	SRP	Data	Input transformation
Zhou *et al.* [99]	DUP-Net	Data	Input transformation
Wu *et al.* [97]	If-Defense	Data	Input transformation
Liu *et al.* [60]	FGSM	Data	Adversarial training
Liu *et al.* [65]	ITA	Data	Adversarial training
Liang *et al.* [100]	PAGN	Data	Adversarial training
Sun *et al.* [101]	—	Data	Adversarial training
Zhang *et al.* [102]	—	Data	Data augmentation
Yang *et al.* [66]	—	Data	Data augmentation
Zhang *et al.* [103]	PointCutMix	Data	Data augmentation
Naderi *et al.* [98]	LPF-Defense	Data	Data augmentation
Zhang *et al.* [104]	Defense-PointNet	Model	Deep model modification
Zhang *et al.* [104]	CCN	Model	Deep model modification
Li *et al.* [105]	LPC	Model	Deep model modification
Sun *et al.* [106]	DeepSym	Combined	Deep model modification & adversarial training

Scaling factor α depends on k and in [99], the authors used $\alpha = 1.1$ and $k = 2$. A similar defense method was proposed in [107]. The Euclidean distance between each point and its k-nearest neighbors was used to detect outliers, and points with high average distances were discarded as outliers.

c: SALIENT POINT REMOVAL (SPR)

Conceptually, salient point removal (**SPR**) is related to SOR, except that the outliers here are identified differently. For example, Liu et al. [60] assumed that the adversarial points have fairly large gradient values. Based on this assumption, this method calculates the saliency of each point using the gradient of the output class of the model with respect to each point, and then removes the points with high saliency scores.

d: DENOISER AND UPSAMPLER NETWORK (DUP-NET)

The **DUP-Net** defense approach consists of two steps. The first is a ''denoising'' step using SOR to remove outliers. This results in a point cloud with fewer points than the input cloud. The second step is upsampling using an upsampler network [108] to produce a denser point cloud. These two steps are meant to undo typical attacks that generate outliers (either by shifting or adding points) in order to fool the deep model. By removing outliers and then bringing back the density, DUP-Net is meant to approximate the original point cloud.

e: IF-DEFENSE

IF-Defense [97] is a preprocessing technique whose first step is SOR to remove outliers from the input point cloud. In the next step, two losses are used to optimize the coordinates of the remaining points under geometry and distribution constraints. The geometry-aware loss tries to push points towards the surface to improve smoothness. To estimate the surfaces of objects, the authors train a separate implicit function network [109], [110]. Because the outputs of implicit functions are continuous, the predicted surface is locally smooth. This reduces the impact of the remaining outliers. The second, distribution-aware loss, encourages the points to have a uniform distribution by maximizing the distance between each point and its k-nearest neighbors. Accordingly, IF-Defense produces smooth, uniformly sampled point clouds.

Figure 8 shows the results of three defense methods – SOR, DUP-Net, and If-Defense – against a Drop200 attack. As seen in the figure, SOR results in a relatively sparse point cloud, while DUP-Net produces a much denser cloud. IF-Defense produces a smooth, approximately uniformly sampled point cloud.

f: MISCELLANEOUS DEFENSES

Besides the above defenses, a few other approaches have been proposed to counter adversarial attacks through input transformation. Dong et al. [96] proposed Gather-vector Guidance (**GvG**), which is sensitive to the change of local features. In case the adversarial perturbation changes the local features, the gather-vector will also change, thereby providing a way to detect the attack. Zhang et al. [111] proposed **Ada3Diff**, which uses adaptive diffusion to smooth out perturbations in the point cloud. In doing so, it acts similarly to outlier removal, since the points perturbed during the attack often reduce local smoothness in order to fool the classifier.

Liu et al. [112] developed an ensembling method called **PointGuard**. Here, a number of random subsets of the point cloud are taken and each is separately classified. Then the majority vote among the labels of these random subsets is taken as the final prediction. Similarly to SRS, the idea is that a random subset has fewer adversarially-perturbed points than the input point cloud, which may make it more likely to be classified correctly. An ensemble of such decisions makes the final prediction more robust.

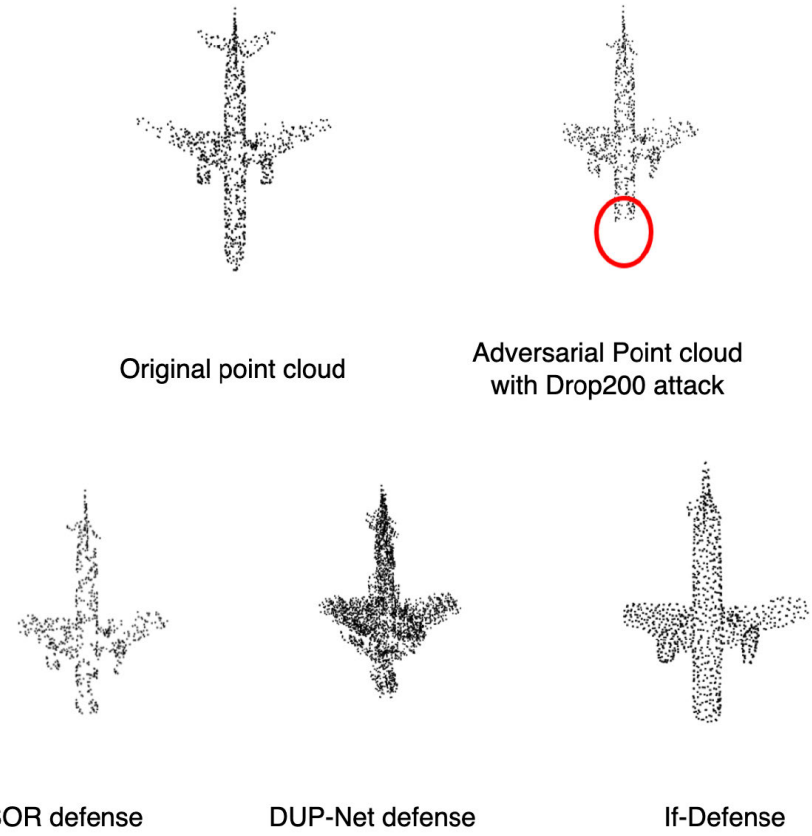

FIGURE 8. Results of three different defense methods applied on the Drop200 attack. Figure taken from [97] (Image source: [97]; use permitted under the Creative Commons Attribution License CC BY 4.0).

2) TRAINING DATA OPTIMIZATION

Another group of defenses involves optimizing training data in order to make the trained model more robust against adversarial attacks. Various modifications to the training data have been proposed, as described below.

a: ADVERSARIAL TRAINING

One way to make the model more robust against adversarial attacks is to expose it to adversarial examples during training, which is termed **adversarial training** [69]. In adversarial training, both the original and adversarial point clouds are used. The use of adversarial training as a defense for point cloud models was first described in [60]. The authors of [60] and [65] trained a deep model by augmenting the training data using adversarial examples generated by FGSM and ITA attacks. As a way to improve adversarial training, the authors of [100] employed 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. [101] applied self-supervised learning to adversarial training on point clouds.

b: POINTCUTMIX

Zhang et al. [103] proposed the **PointCutMix** technique to generate a new training set by swapping points between two optimally aligned original point clouds and training a model on this new training set. PointCutMix provides two strategies for point swapping: randomly replacing all points or replacing the k nearest neighbors of a randomly chosen point. Additionally, the method uses a saliency map to guide point selection, enhancing its effectiveness. Augmented sample labels in PointCutMix are formed by blending the labels of the source point clouds. The augmented point clouds, along with their associated labels, are integrated into the training set, thereby creating a novel collection of training samples that capture variations from both original point clouds. Overall, PointCutMix proves valuable for augmenting point cloud data in tasks such as classification and defense against adversarial attacks.

c: LOW PASS FREQUENCY-DEFENSE (LPF-DEFENSE)

In LPF-Defense [98], deep models are trained with the low-frequency version of the original point clouds. More specifically, using the Spherical Harmonic Transform (SHT) [113], original point clouds are transformed from the spatial to the frequency domain. Then the high-frequency components are removed and the low-frequency version of the point cloud is recovered in the spatial domain. The idea is that adversarial attacks, through point shifting, insertion, or deletion, often introduce high frequencies into the point cloud. When a deep model is trained on the low-frequency

versions of the point clouds, it learns to associate the label with low frequencies and thereby implicitly ignores high frequencies which may have been introduced during an attack.

B. MODEL-FOCUSED STRATEGIES

1) DEEP MODEL MODIFICATION

Another class of defenses involves modifying the architecture of the deep model itself and may involve retraining in order to improve its robustness to adversarial attacks. Examples of this type of defense are given below.

a: DEFENSE-POINTNET

Zhang et al. [104] proposed a defense method that involves splitting the PointNet model into two parts. The first part is the feature extractor, with a discriminator attached to its last layer. The second part is the remainder of the PointNet model, which acts as a classifier. The feature extractor is fed with a mini-batch consisting of the original point clouds and adversarial examples generated by the FGSM attack. The discriminator attempts to classify whether the features come from the original or adversarial point cloud. Model parameters are optimized using three different loss functions: one for the classifier, one for the discriminator, and one for the feature extractor. While discriminator loss encourages the model to distinguish the original point cloud from the adversarial one, the feature extractor loss tries to mislead the discriminator to label every feature vector as the original. Therefore, the feature extractor acts as an adversary to the discriminator. Finally, the classifier loss encourages the classifier to give correct predictions for each input.

b: CONTEXT-CONSISTENCY DYNAMIC GRAPH NETWORK (CCN)

Li et al. [114] proposed two methodologies to improve the adversarial robustness of 3D point cloud classification models. The first one involves a novel point cloud architecture named the Context-Consistency dynamic graph Network (CCN). This architecture is predominantly constructed upon the Dynamic Graph CNN (DGCNN) model [115], but it incorporates a lightweight Context-Consistency Module (CCM) into various layers of DGCNN. This module aims to reduce feature gaps between clean and noisy samples. The second one is a new data augmentation technique. In each training epoch, the method generates three types of batches from each sample: adversarial examples created by dropping points, adversarial examples created by shifting points, and clean samples. Subsequently, it dynamically identifies the most appropriate samples based on their accuracy to train the model, thereby adaptively balancing the model's accuracy and robustness to attacks. To provide a more robust model against adversarial point clouds, the authors integrate the two methodologies.

c: LATTICE POINT CLASSIFIER (LPC)

Li et al. [105] proposed embedding a declarative node into the networks to transform adversarial point clouds such that they may be classified more easily. Specifically, structured sparse coding in the permutohedral lattice [116] is used to construct a Lattice Point Classifier (LPC). The LPC projects each point cloud onto a lattice and generates a 2D image, which is then input to a 2D CNN for classification. Projection onto a lattice may remove some of the noise and/or outliers introduced during an adversarial attack.

2) MODEL RETRAINING

Model (re)training strategies include adversarial (re)training, discussed in Section IV-A2a, but also other strategies intended to make the model more robust without explicitly using adversarial examples. Such strategies may involve various data augmentation methods, additional regularization terms to encourage robustness and generalization, as well as contrastive learning to robustify class boundaries. The authors of [66], [102] augmented the training data by noise to make the resulting model more robust against attacks. One noise model employed was additive Gaussian noise, which was meant to improve robustness against point shifts in an attack. Another type of noise used was quantization noise, which involved converting point cloud coordinates to low precision during training. Quantization noise is often modeled as uniform noise [117], so this augmentation was meant to improve robustness against small point movements in a limited range.

3) COMBINED STRATEGIES

Some of the adversarial defense methods combine various strategies described above. For example, Sun et al. [106] studied the role of pooling operations in enhancing model robustness during adversarial training. They found that fixed operations like max-pooling weaken the effectiveness of adversarial training, while sorting-based parametric pooling operations improve the model's robustness. As a result, they proposed **DeepSym**, a symmetric pooling operation that increases model's robustness to attacks.

V. DATASETS AND VICTIM MODELS

A variety of 3D point cloud datasets have been collected to train and evaluate deep models on point cloud classification. These include ModelNet [118], ShapeNet [119], ScanObjectNN [120], McGill Benchmark [121], ScanNet [122], Sydney Urban Objects [123]. A summary of these datasets and their unique characteristics is presented in Table 5.

These 3D point cloud datasets can be broadly categorized into two groups: synthetic and real. ShapeNet and ModelNet are well-known datasets that contain synthetic data. These datasets are often used for model training and evaluation in controlled settings, because objects in synthetic datasets are typically complete, without occlusions, "holes," and free of noise. For instance, ModelNet10 and ModelNet40 consist

of 3D models of various objects, categorized into 10 and 40 classes, respectively, and are widely used in point cloud research. ShapeNet is a larger dataset with a larger number of classes, making it suitable for more challenging classification tasks. Virtual KITTI [124] is an example of a synthetic dataset built for autonomous driving.

In contrast, datasets such as ScanNet and ScanObjectNN contain real data collected from real-world measurements, reflecting the complexity and variability of actual environments. ScanObjectNN is a real-world dataset suitable for evaluating 3D object classification in real-world scenarios. ScanNet is another real-world dataset that includes 3D scans of indoor environments. KITTI [125] is a real-world dataset featuring 3D scenes related to autonomous driving. Real 3D point-cloud scans are often subject to occlusion and may contain noise, which may necessitate "hole filling" [126] and/or denoising [127] before further use. Among the datasets discussed above, ModelNet10 [118], ModelNet40 [118], ShapeNet [119] and ScanObjectNN [120] have been very popular in the literature on point cloud adversarial attacks and defenses.

Table 6 presents an overview of prominent victim models that researchers commonly employ to assess adversarial attacks and defense strategies in the context of point cloud classification. PointNet, PointNet++, and DGCNN are the models that are the most frequently targeted for adversarial assessment. Each of these models employs distinct mechanisms for processing point clouds. PointNet employs multi-layer perceptrons (MLPs) to extract pointwise features and aggregate them using max-pooling. PointNet++ builds upon PointNet, incorporating three key layers: the sampling layer, the grouping layer, and the PointNet-based learning layer. This architecture is repeated to capture fine geometric structures in point clouds. DGCNN, another widely used model, leverages local geometric structures by constructing a local neighborhood graph and applying convolution-like operations on the edges connecting neighboring points.

Beyond these popular models, there are other notable architectures like PointConv, which extends the Monte Carlo approximation of 3D continuous convolution operators. It employs MLPs to approximate weight functions for each convolutional filter and applies density scaling to re-weight these learned functions. The Relation-Shape Convolutional Neural Network (RS-CNN) extends regular grid CNNs to handle irregular point-cloud configurations. It achieves this by emphasizing the importance of learning geometric relations among points, forcing the convolutional weights to capture these relations based on predefined geometric priors. VoxNet, on the other hand, is an architecture that combines a volumetric grid and a 3D CNN to improve object recognition using point cloud data from sensors like LiDAR and RGBD cameras. VoxNet predicts object class labels directly from the volumetric occupancy information.

SpiderCNN is specifically designed for extracting geometric features from point clouds. It achieves this by using a family of convolutional filters parametrized as a combination of a step function, capturing local geodesic information, and a Taylor polynomial to enhance expressiveness. PointASNL is capable of handling noisy point clouds effectively. Its core feature is the adaptive sampling module, which re-weights and adjusts sampled points to improve feature learning and mitigate the impact of outliers. It also includes a local-nonlocal module to capture local and global dependencies. CurveNet addresses the limitations of existing local feature aggregation approaches by grouping sequences of connected points (curves) through guided walks in point clouds and then integrating these curve features with point-wise features. Lastly, AtlasNet introduces a novel approach to 3D shape generation that does not rely on voxelized or point-cloud representations. Instead, it directly learns surface representations by deforming a set of learnable parameterizations.

VI. CHALLENGES AND FUTURE DIRECTIONS

In this section, we explore the current challenges within the domain of adversarial attacks and defenses on 3D point clouds. We also present several promising directions for future research in this area.

A. CURRENT CHALLENGES

1) CRAFTING REAL-WORLD ATTACKS

As mentioned earlier, majority of attacks on 3D point clouds reported in the literature are white-box attacks. However, in practice, the white-box scenario is much less likely compared to the black-box and gray-box scenarios. Existing results suggest that black-box attacks are much less effective than white-box attacks. Hence, one of the current challenges is developing attack strategies that do not rely on complete knowledge of the target model and whose effectiveness could approach that of white-box attacks.

2) UNDERSTANDING THE ROLE OF FREQUENCY

Points in a point cloud are irregularly placed in the 3D space. This makes understanding the frequency content of point clouds more challenging than that in the case of images or other regularly-sampled signals. Tools from graph signal processing [75] or spherical harmonic analysis [113] are useful in this context, but the fact remains that even the basic notion of frequency and its role in attacks and defenses is harder to analyze in the case of point clouds. Many attacks introduce high frequencies into the point cloud through methods like point shifting or adding. But if the original point cloud already contains high frequencies, they may mask the attack and therefore make defenses less effective.

A better understanding of the role of frequency may help explain the reasons behind the vulnerability of 3D deep models to adversarial attacks. For example, [98] has tackled this problem, suggesting that 3D deep models may rely too heavily on high-frequency details within 3D point clouds, and removing these details could potentially lead to models that are more robust against attacks. Such deeper understanding may be useful in the context of adversarial attacks and defenses in other areas, not just 3D point clouds.

TABLE 5. Summary of the datasets commonly used 3D point cloud classification.

Dataset	Year	Type	Classes	Samples (Training / Test)
ModelNet10 [118]	2015	Synthetic	10	4899 (3991 / 605)
ModelNet40 [118]	2015	Synthetic	40	12311 (9843 / 2468)
ShapeNet [119]	2015	Synthetic	55	51190 (/)
ScanObjectNN [120]	2019	Real	15	2902 (2321 / 581)
KITTI [125]	2012	Real	8	7058 (6347 / 711)
Virtual KITTI [124]	2016	Synthetic	8	21260 (/)
ScanNet [122]	2017	Real	17	12283 (9677 / 2606)
3DMNIST [128]	2019	Synthetic	10	12000 (10000 / 2000)
McGill Benchmark [121]	2008	Synthetic	19	456 (304 / 152)
Sydney Urban Objects [123]	2013	Real	14	588 (/)

TABLE 6. Summary of datasets and victim models used in attacks and defenses on 3D point clouds.

Datasets	ModelNet10 [118]	[67], [106], [129], [82]
	ModelNet40 [118]	[60], [67], [65], [63], [62], [61], [59], [57], [58], [54] [53], [97], [78], [130], [100], [131], [106], [132], [129], [73] [56], [96], [82], [79], [64], [99], [133], [105], [134], [76], [85], [135]
	ShapeNet [119]	[61], [57], [55], [97], [130], [79], [93], [104]
	ScanObjectNN [120]	[63], [131], [112], [106], [129]
	KITTI [125]	[67], [136]
	ScanNet [19]	[112], [105]
	3D-MNIST [128]	[54], [85]
Victim models	PointNet [19]	[60], [67], [66], [65], [63], [62], [61], [57], [58], [55] [54], [53], [97], [100], [74], [131], [112], [106], [132], [129] [89], [73], [56], [96], [82], [79], [93], [104], [64], [99] [136], [133], [105], [134], [76], [85], [135]
	PointNet++ [137]	[60], [66], [65], [63], [62], [61], [59], [57], [58], [55] [54], [53], [97], [78], [130], [100], [74], [132], [89], [73] [56], [96], [79], [64], [99], [136], [134], [76], [85], [135]
	DGCNN [138]	[66], [65], [63], [62], [61], [57], [58], [55], [54], [53] [97], [78], [130], [100], [74], [112], [129], [89], [73], [56] [82], [79], [64], [133], [134], [76], [85], [135]
	PointConv [139]	[97], [130], [74]
	RS-CNN [140]	[97], [135]
	VoxNet [141]	[67]
	SpiderCNN [142]	[63]
	PointASNL [143]	[63]
	CurveNet [144]	[78]
	AtlasNet [145]	[93]
	PointTrans [146]	[76]
	PointMLP [147]	[76]

3) TRAINING FOR ROBUSTNESS

As mentioned earlier in Section IV, model training plays a key role in achieving robustness against adversarial attacks. The issue of distinguishing the appearance of point clouds from one class versus another class may present significant challenges. For example, a ''flower pot'' looks similar to a ''cup'' (see Figure 6) due to its conical shape, so it does not take much to make a deep model misclassify one for another. From this point of view, models should be trained to be very strong at distinguishing classes whose appearance is similar. This would help improve not only robustness against attacks but also the overall accuracy and generalizability.

A basic premise in statistical ML [148] is that simpler models generalize better, although they may not be as accurate as more complex models. Since adversarial attacks often involve perturbations of the original point cloud, this would seem to imply that simpler models are less likely to be fooled by them. From this point of view, the choice of a model is a trade-off between accuracy, which generally requires higher complexity, and robustness (to attacks, as well as unseen data), which seems to favor not-too-high complexity. Since accuracy is the predominant factor of usefulness of a model, the trend has been towards more complex models, but with additional regularization [148] and more sophisticated learning strategies to strengthen the robustness.

B. FUTURE DIRECTIONS

1) TRANSFERABILITY

In the context of adversarial attacks, the term **transferability** refers to the ability of an attack against a given target model to

be effective against a different, potentially unknown model. Transferable attacks are not tied the specifics of any one model, but target more fundamental issues, and are therefore also useful in broadening the understanding of adversarial attack and defense principles. Currently, there is a limited amount of research on transferable attacks on 3D point clouds [55], [65], [68], [74], so this is one potentially fruitful direction for future research.

2) NEW TASKS

Presently, most adversarial attack research is focused on the classification task. This was in part influenced by the wide availability of datasets and related classification models. However, in practice, the role of adversarial attacks is to disrupt a complex system, which may involve other tasks such as detection, segmentation, tracking, etc. It is important to study adversarial attacks and defenses in these more general settings, in order to gain a comprehensive understanding of their performance in diverse applications.

3) POINT CLOUD ATTRIBUTES

The vast majority of adversarial attacks and defenses related to point clouds have focused on point-cloud geometry. However, point clouds may also have attributes such as color [149]. Changing the color of points in a point cloud may disrupt classification, segmentation, and other analysis tasks, hence attributes are a potential target for attacks. Since the color attributes of a point cloud play a similar role to the pixel colors in an image, 2D attacks and defenses may provide useful guidelines for initiating the work in this area. Moreover, this would open up possibilities for creating attacks and defenses that simultaneously consider geometry and attributes, a previously unexplored topic.

VII. CONCLUSION

Adversarial attacks on 3D point cloud classification have become a significant concern in recent years. These attacks are able to manipulate 3D point clouds in a way that leads the victim model(s) to make incorrect decisions with potentially harmful consequences. Adversarial attacks on 3D point clouds can be categorized according to the methodologies employed to modify the point cloud, and may have additional attributes in terms of the location of the attack, target type, and adversarial knowledge. We have reviewed a variety of attack methodologies, with examples from the existing literature, highlighting their main characteristics and their relationships.

To defend against these attacks, researchers have proposed two main categories of approaches: data-focused and model-focused. Data-focused techniques attempt to undo adversarial modifications on the point cloud in order to increase the chance of correct decision, while model-focused approaches attempt to make the model(s) more resilient to adversarial attacks. For stronger protection against attacks, data-focused and model-focused techniques can be combined.

In addition to reviewing the main attack and defense approaches related to 3D point cloud classification, we also presented the main datasets used in this field, as well as the most widely used victim models. Finally, we summarized the main challenges and outlined possible directions for future research in this field. We hope the article will be helpful to those entering the field of adversarial attacks on 3D point clouds and serve the current research community as a quick reference.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[2] Y. Li, "Research and application of deep learning in image recognition," in *Proc. IEEE 2nd Int. Conf. Power, Electron. Comput. Appl. (ICPECA)*, Jan. 2022, pp. 994–999.
[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
[4] H. Naderi, L. Goli, and S. Kasaei, "Scale equivariant CNNs with scale steerable filters," in *Proc. Int. Conf. Mach. Vis. Image Process. (MVIP)*, Feb. 2020, pp. 1–5.
[5] N. Silberman and R. Fergus, "Indoor scene segmentation using a structured light sensor," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Nov. 2011, pp. 601–608.
[6] Y. Mo, Y. Wu, X. Yang, F. Liu, and Y. Liao, "Review the state-of-the-art technologies of semantic segmentation based on deep learning," *Neurocomputing*, vol. 493, pp. 626–646, Jul. 2022.
[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. 19143–19165, 2019.
[8] K. I. Taher and A. M. Abdulazeez, "Deep learning convolutional neural network for speech recognition: A review," *Int. J. Sci. Bus.*, vol. 5, no. 3, pp. 1–14, 2021.
[9] K. Chowdhary, "Natural language processing," in *Fundamentals of Artificial Intelligence*. New Delhi, India: Springer, 2020, pp. 603–649.
[10] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proc. ICLR*, 2014, pp. 1–15.
[11] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2574–2582.
[12] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1625–1634.
[13] M. Mozaffari-Kermani, S. Sur-Kolay, A. Raghunathan, and N. K. Jha, "Systematic poisoning attacks on and defenses for machine learning in healthcare," *IEEE J. Biomed. Health Informat.*, vol. 19, no. 6, pp. 1893–1905, Nov. 2015.
[14] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. M. Paixao, and F. Mutz, "Self-driving cars: A survey," *Expert Syst. Appl.*, vol. 165, Mar. 2021, Art. no. 113816.
[15] M. Hassanalian and A. Abdelkefi, "Classifications, applications, and design challenges of drones: A review," *Prog. Aerosp. Sci.*, vol. 91, pp. 99–131, May 2017.
[16] H. A. Pierson and M. S. Gashler, "Deep learning in robotics: A review of recent research," *Adv. Robot.*, vol. 31, no. 16, pp. 821–835, Aug. 2017.
[17] Z. Liu, H. Tang, Y. Lin, and S. Han, "Point-voxel CNN for efficient 3D deep learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.
[18] L. Ladický, O. Saurer, S. Jeong, F. Maninchedda, and M. Pollefeys, "From point clouds to mesh using regression," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3913–3922.
[19] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.
[20] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14410–14430, 2018.
[21] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2805–2824, Sep. 2019.
[22] S. Qiu, Q. Liu, S. Zhou, and C. Wu, "Review of artificial intelligence adversarial attack and defense technologies," *Appl. Sci.*, vol. 9, no. 5, p. 909, Mar. 2019.

[23] H. Wei, H. Tang, X. Jia, Z. Wang, H. Yu, Z. Li, S. Satoh, L. Van Gool, and Z. Wang, "Physical adversarial attack meets computer vision: A decade survey," 2022, *arXiv:2209.15179*.

[24] Z. Zhai, P. Li, and S. Feng, "State of the art on adversarial attacks and defenses in graphs," *Neural Comput. Appl.*, vol. 35, no. 26, pp. 18851–18872, Sep. 2023.

[25] P. Bountakas, A. Zarras, A. Lekidis, and C. Xenakis, "Defense strategies for adversarial machine learning: A survey," *Comput. Sci. Rev.*, vol. 49, Aug. 2023, Art. no. 100573.

[26] S. Pavlitska, N. Lambing, and J. M. Zöllner, "Adversarial attacks on traffic sign recognition: A survey," in *Proc. 3rd Int. Conf. Electr., Comput., Commun. Mechatronics Eng. (ICECCME)*, Jul. 2023, pp. 1–6.

[27] 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. 155161–155196, 2021.

[28] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3D point clouds: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 12, pp. 4338–4364, Dec. 2021.

[29] A. Xiao, J. Huang, D. Guan, X. Zhang, S. Lu, and L. Shao, "Unsupervised point cloud representation learning with deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–20, 2023.

[30] Y. Xie, J. Tian, and X. X. Zhu, "Linking points with labels in 3D: A review of point cloud semantic segmentation," *IEEE Geosci. Remote Sens. Mag.*, vol. 8, no. 4, pp. 38–59, Dec. 2020.

[31] H. Zhang, C. Wang, S. Tian, B. Lu, L. Zhang, X. Ning, and X. Bai, "Deep learning-based 3D point cloud classification: A systematic survey and outlook," *Displays*, vol. 79, Sep. 2023, Art. no. 102456.

[32] D. Fernandes, A. Silva, R. Névoa, C. Simões, D. Gonzalez, M. Guevara, P. Novais, J. Monteiro, and P. Melo-Pinto, "Point-cloud based 3D object detection and classification methods for self-driving applications: A survey and taxonomy," *Inf. Fusion*, vol. 68, pp. 161–191, Apr. 2021.

[33] D. Krawczyk and R. Sitnik, "Segmentation of 3D point cloud data representing full human body geometry: A review," *Pattern Recognit.*, vol. 139, Jul. 2023, Art. no. 109444.

[34] C. Cao, M. Preda, and T. Zaharia, "3D point cloud compression: A survey," in *Proc. The 24th Int. Conf. 3D Web Technol.*, 2019, pp. 1–9.

[35] R. Reza Wiyatno, A. Xu, O. Dia, and A. de Berker, "Adversarial examples in modern machine learning: A review," 2019, *arXiv:1911.05268*.

[36] H. Xu, Y. Ma, H.-C. Liu, D. Deb, H. Liu, J.-L. Tang, and A. K. Jain, "Adversarial attacks and defenses in images, graphs and text: A review," *Int. J. Autom. Comput.*, vol. 17, no. 2, pp. 151–178, Apr. 2020.

[37] N. Martins, J. M. Cruz, T. Cruz, and P. Henriques Abreu, "Adversarial machine learning applied to intrusion and malware scenarios: A systematic review," *IEEE Access*, vol. 8, pp. 35403–35419, 2020.

[38] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "A survey on adversarial attacks and defences," *CAAI Trans. Intell. Technol.*, vol. 6, no. 1, pp. 25–45, Mar. 2021.

[39] I. Rosenberg, A. Shabtai, Y. Elovici, and L. Rokach, "Adversarial machine learning attacks and defense methods in the cyber security domain," *ACM Comput. Surv.*, vol. 54, no. 5, pp. 1–36, Jun. 2022.

[40] A. Michel, S. K. Jha, and R. Ewetz, "A survey on the vulnerability of deep neural networks against adversarial attacks," *Prog. Artif. Intell.*, vol. 11, no. 2, pp. 131–141, Jun. 2022.

[41] 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, Jul. 2022.

[42] 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, Jul. 2022.

[43] H. Liang, E. He, Y. Zhao, Z. Jia, and H. Li, "Adversarial attack and defense: A survey," *Electronics*, vol. 11, no. 8, p. 1283, Apr. 2022.

[44] Y. Li, M. Cheng, C.-J. Hsieh, and T. C. M. Lee, "A review of adversarial attack and defense for classification methods," *Amer. Statistician*, vol. 76, no. 4, pp. 329–345, Oct. 2022.

[45] K. D. Gupta and D. Dasgupta, "Adversarial attacks and defenses for deployed AI models," *IT Prof.*, vol. 24, no. 4, pp. 37–41, Jul. 2022.

[46] X. Wei, B. Pu, J. Lu, and B. Wu, "Visually adversarial attacks and defenses in the physical world: A survey," 2022, *arXiv:2211.01671*.

[47] J.-X. Mi, X.-D. Wang, L.-F. Zhou, and K. Cheng, "Adversarial examples based on object detection tasks: A survey," *Neurocomputing*, vol. 519, pp. 114–126, Jan. 2023.

[48] S. Y. Khamaiseh, D. Bagagem, A. Al-Alaj, M. Mancino, and H. W. Alomari, "Adversarial deep learning: A survey on adversarial attacks and defense mechanisms on image classification," *IEEE Access*, vol. 10, pp. 102266–102291, 2022.

[49] S. Kotyan, "A reading survey on adversarial machine learning: Adversarial attacks and their understanding," 2023, *arXiv:2308.03363*.

[50] H. Baniecki and P. Biecek, "Adversarial attacks and defenses in explainable artificial intelligence: A survey," in *Proc. ICML Workshop*, 2023, pp. 1–13.

[51] S. Han, C. Lin, C. Shen, Q. Wang, and X. Guan, "Interpreting adversarial examples in deep learning: A review," *ACM Comput. Surv.*, vol. 55, no. 14s, pp. 1–38, Jul. 2023.

[52] D. Tian, H. Ochimizu, C. Feng, R. Cohen, and A. Vetro, "Geometric distortion metrics for point cloud compression," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3460–3464.

[53] C. Xiang, C. R. Qi, and B. Li, "Generating 3D adversarial point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9128–9136.

[54] T. Zheng, C. Chen, J. Yuan, B. Li, and K. Ren, "Pointcloud saliency maps," in *Proc. IEEE/CVF ICCV*, Oct. 2019, pp. 1598–1606.

[55] A. Hamdi, S. Rojas, A. Thabet, and B. Ghanem, "ADVPC: Transferable adversarial perturbations on 3D point clouds," in *Computer Vision—ECCV 2020*. Cham, Switzerland: Springer, 2020, pp. 241–257.

[56] K. Lee, Z. Chen, X. Yan, R. Urtasun, and E. Yumer, "ShapeAdv: Generating shape-aware adversarial 3D point clouds," 2020, *arXiv:2005.11626*.

[57] H. Zhou, D. Chen, J. Liao, K. Chen, X. Dong, K. Liu, W. Zhang, G. Hua, and N. Yu, "LG-GAN: Label guided adversarial network for flexible targeted attack of point cloud based deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10353–10362.

[58] Y. Wen, J. Lin, K. Chen, C. L. P. Chen, and K. Jia, "Geometry-aware generation of adversarial point clouds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 2984–2999, Jun. 2022.

[59] T. Tsai, K. Yang, T.-Y. Ho, and Y. Jin, "Robust adversarial objects against deep learning models," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 1, pp. 954–962.

[60] D. Liu, R. Yu, and H. Su, "Extending adversarial attacks and defenses to deep 3D point cloud classifiers," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 2279–2283.

[61] A. Arya, H. Naderi, and S. Kasaei, "Adversarial attack by limited point cloud surface modifications," in *Proc. 6th Int. Conf. Pattern Recognit. Image Anal. (IPRIA)*, Feb. 2023, pp. 1–8.

[62] D. Liu, R. Yu, and H. Su, "Adversarial shape perturbations on 3D point clouds," in *Computer Vision—ECCV 2020*. Cham, Switzerland: Springer, 2020, pp. 88–104.

[63] J. Kim, B.-S. Hua, D. T. Nguyen, and S.-K. Yeung, "Minimal adversarial examples for deep learning on 3D point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 7777–7786.

[64] 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 *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 1819–1827.

[65] D. Liu and W. Hu, "Imperceptible transfer attack and defense on 3D point cloud classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4727–4746, Apr. 2023.

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

[67] M. Wicker and M. Kwiatkowska, "Robustness of 3D deep learning in an adversarial setting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11759–11767.

[68] B. He, J. Liu, Y. Li, S. Liang, J. Li, X. Jia, and X. Cao, "Generating transferable 3D adversarial point cloud via random perturbation factorization," in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, no. 1, pp. 764–772.

[69] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. ICLR*, 2015.

[70] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9185–9193.

[71] A. Mądry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. STAT*, vol. 1050, 2017, p. 9.

[72] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 39–57.

[73] J. Zhang, L. Chen, B. Liu, B. Ouyang, Q. Xie, J. Zhu, W. Li, and Y. Meng, "3D adversarial attacks beyond point cloud," *Inf. Sci.*, vol. 633, pp. 491–503, Jul. 2023.
[74] B. Liu, J. Zhang, and J. Zhu, "Boosting 3D adversarial attacks with attacking on frequency," *IEEE Access*, vol. 10, pp. 50974–50984, 2022.
[75] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.
[76] D. Liu, W. Hu, and X. Li, "Point cloud attacks in graph spectral domain: When 3D geometry meets graph signal processing," 2022, *arXiv:2207.13326*.
[77] Q. Hu, D. Liu, and W. Hu, "Exploring the devil in graph spectral domain for 3D point cloud attacks," in *Computer Vision—ECCV 2022*. Cham, Switzerland: Springer, 2022, pp. 229–248.
[78] Q. Huang, X. Dong, D. Chen, H. Zhou, W. Zhang, and N. Yu, "Shape-invariant 3D adversarial point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 15314–15323.
[79] Y. Zhao, Y. Wu, C. Chen, and A. Lim, "On isometry robustness of deep 3D point cloud models under adversarial attacks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1198–1207.
[80] D. Russo, B. Van Roy, A. K. Benjamin, and A. Osband, "A tutorial on Thompson sampling. Foundations and trends," *Mach. Learn.*, vol. 11, no. 10, 2017, Art. no. 2200000070.
[81] 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," *Secur. Commun. Netw.*, vol. 2022, pp. 1–11, Aug. 2022.
[82] Y. Zhao, I. Shumailov, R. Mullins, and R. Anderson, "Nudge attacks on point-cloud DNNs," 2020, *arXiv:2011.11637*.
[83] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 828–841, Oct. 2019.
[84] H. Tan and H. Kotthaus, "Explainability-aware one point attack for point cloud neural networks," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 4570–4579.
[85] Z. Shi, C. Zhi, X. Zhenbo, Y. Wei, Y. Zhidong, and L. Huang, "Shape prior guided attack: Sparser perturbations on 3D point clouds," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 8277–8285.
[86] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.
[87] H. Naderi, C. Dinesh, I. V. Bajic, and S. Kasaei, "Model-free prediction of adversarial drop points in 3D point clouds," 2022, *arXiv:2210.14164*.
[88] K. Tang, J. Wu, W. Peng, Y. Shi, P. Song, Z. Gu, Z. Tian, and W. Wang, "Deep manifold attack on point clouds via parameter plane stretching," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 2, 2023, pp. 2420–2428.
[89] X. Dai, Y. Li, H. Dai, and B. Xiao, "Generating unrestricted 3D adversarial point clouds," 2021, *arXiv:2111.08973*.
[90] T. B. Brown, N. Carlini, C. Zhang, C. Olsson, P. Christiano, and I. Goodfellow, "Unrestricted adversarial examples," 2018, *arXiv:1809.08352*.
[91] H. Naderi, L. Goli, and S. Kasaei, "Generating unrestricted adversarial examples via three parameteres," *Multimedia Tools Appl.*, vol. 81, no. 15, pp. 21919–21938, Jun. 2022.
[92] Y. Song, R. Shu, N. Kushman, and S. Ermon, "Constructing unrestricted adversarial examples with generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.
[93] I. Lang, U. Kotlicki, and S. Avidan, "Geometric adversarial attacks and defenses on 3D point clouds," in *Proc. Int. Conf. 3D Vis. (3DV)*, Dec. 2021, pp. 1196–1205.
[94] G. Mariani, L. Cosmo, A. M. Bronstein, and E. Rodola, "Generating adversarial surfaces via band-limited perturbations," *Comput. Graph. Forum*, vol. 39, no. 5, pp. 253–264, Aug. 2020.
[95] B. S. Vivek, K. R. Mopuri, and R. V. Babu, "Gray-box adversarial training," in *Proc. ECCV*, Sep. 2018, pp. 203–218.
[96] X. Dong, D. Chen, H. Zhou, G. Hua, W. Zhang, and N. Yu, "Self-robust 3D point recognition via gather-vector guidance," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11513–11521.
[97] Z. Wu, Y. Duan, H. Wang, Q. Fan, and L. J. Guibas, "IF-defense: 3D adversarial point cloud defense via implicit function based restoration," 2020, *arXiv:2010.05272*.
[98] H. Naderi, K. Noorbakhsh, A. Etemadi, and S. Kasaei, "LPF-defense: 3D adversarial defense based on frequency analysis," *PLoS ONE*, vol. 18, no. 2, Feb. 2023, Art. no. e0271388.
[99] H. Zhou, K. Chen, W. Zhang, H. Fang, W. Zhou, and N. Yu, "DUP-Net: Denoiser and upsampler network for 3D adversarial point clouds defense," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1961–1970.
[100] Q. Liang, Q. Li, W. Nie, and A.-A. Liu, "PAGN: Perturbation adaption generation network for point cloud adversarial defense," *Multimedia Syst.*, vol. 28, no. 3, pp. 851–859, Jun. 2022.
[101] 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," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021.
[102] Y. Zhang, J. Hou, and Y. Yuan, "A comprehensive study of the robustness for LiDAR-based 3D object detectors against adversarial attacks," 2022, *arXiv:2212.10230*.
[103] 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, Sep. 2022.
[104] Y. Zhang, G. Liang, T. Salem, and N. Jacobs, "Defense-PointNet: Protecting PointNet against adversarial attacks," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 5654–5660.
[105] K. Li, Z. Zhang, C. Zhong, and G. Wang, "Robust structured declarative classifiers for 3D point clouds: Defending adversarial attacks with implicit gradients," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 15273–15283.
[106] J. Sun, K. Koenig, Y. Cao, Q. A. Chen, and Z. Mao, "On the adversarial robustness of 3D point cloud classification," in *Proc. BMVC*, 2021.
[107] 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," 2018, *arXiv:1812.11017*.
[108] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "PU-Net: Point cloud upsampling network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2790–2799.
[109] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, "Convolutional occupancy networks," in *Computer Vision—ECCV 2020*. Cham, Switzerland: Springer, 2020, pp. 523–540.
[110] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3D reconstruction in function space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4455–4465.
[111] K. Zhang, H. Zhou, J. Zhang, Q. Huang, W. Zhang, and N. Yu, "Ada3Diff: Defending against 3D adversarial point clouds via adaptive diffusion," in *Proc. 31st ACM Int. Conf. Multimedia*, Oct. 2023, pp. 8849–8859.
[112] H. Liu, J. Jia, and N. Z. Gong, "PointGuard: Provably robust 3D point cloud classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6182–6191.
[113] T. S. Cohen, M. Geiger, J. Köhler, and M. Welling, "Spherical CNNs," in *Proc. ICLR*, 2018.
[114] G. Li, G. Xu, H. Qiu, R. He, J. Li, and T. Zhang, "Improving adversarial robustness of 3D point cloud classification models," in *Computer Vision—ECCV 2022*. Cham, Switzerland: Springer, 2022, pp. 672–689.
[115] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, Oct. 2019.
[116] M. Kiefel, V. Jampani, and P. V. Gehler, "Permutohedral lattice CNNs," 2014, *arXiv:1412.6618*.
[117] B. Widrow and I. Kollár, *Quantization Noise*. Cambridge, U.K.: Cambridge Univ. Press, 2008.
[118] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.
[119] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*.
[120] 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 *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1588–1597.
[121] K. Siddiqi, J. Zhang, D. Macrini, A. Shokoufandeh, S. Bouix, and S. Dickinson, "Retrieving articulated 3-D models using medial surfaces," *Mach. Vis. Appl.*, vol. 19, no. 4, pp. 261–275, Jul. 2008.

[122] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2432–2443.

[123] M. De Deuge, A. Quadros, C. Hung, and B. Douillard, "Unsupervised feature learning for classification of outdoor 3D scans," in *Proc. Australas. Conf. Robitics Autom.*, vol. 2. Kensington, NSW, Australia: University of New South Wales, 2013, p. 1.

[124] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "VirtualWorlds as proxy for multi-object tracking analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4340–4349.

[125] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[126] C. Dinesh, I. V. Bajic, and G. Cheung, "Adaptive nonrigid inpainting of three-dimensional point cloud geometry," *IEEE Signal Process. Lett.*, vol. 25, no. 6, pp. 878–882, Jun. 2018.

[127] C. Dinesh, G. Cheung, and I. V. Bajic, "Point cloud denoising via feature graph Laplacian regularization," *IEEE Trans. Image Process.*, vol. 29, pp. 4143–4158, 2020.

[128] *A 3D Version of the MNIST Database of Handwritten Digits*. Accessed: Jan. 30, 2019. [Online]. Available: https://www.kaggle.com/datasets/daavoo/3d-mnist

[129] 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 *Proc. Int. Conf. Mach. Learn. Workshop (ICMLW)*, vol. 1, 2021.

[130] 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 Things J.*, vol. 10, no. 6, pp. 5158–5169, Mar. 2023.

[131] D. D. Denipitiyage, T. Ajanthan, P. Kamalaruban, and A. Weller, "Provable defense against clustering attacks on 3D point clouds," in *Proc. AAAI*, 2021.

[132] Y. Sun, F. Chen, Z. Chen, and M. Wang, "Local aggressive adversarial attacks on 3D point cloud," in *Proc. Asian Conf. Mach. Learn.*, 2021, pp. 65–80.

[133] C. Ma, W. Meng, B. Wu, S. Xu, and X. Zhang, "Towards effective adversarial attack against 3D point cloud classification," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2021, pp. 1–6.

[134] J. Wang, "Adversarial examples in physical world," in *Proc. 30th Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 19716–19731.

[135] 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.

[136] R. Cheng, N. Sang, Y. Zhou, and X. Wang, "Universal adversarial attack against 3D object tracking," in *Proc. IEEE 23rd Int. Conf. High Perform. Comput. Commun., 7th Int. Conf. Data Sci. Syst. 19th Int. Conf. Smart City; 7th Int. Conf. Dependability Sensor, Cloud Big Data Syst. Appl.*, Dec. 2021, pp. 34–40.

[137] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[138] A. V. Phan, M. L. Nguyen, Y. L. H. Nguyen, and L. T. Bui, "DGCNN: A convolutional neural network over large-scale labeled graphs," *Neural Netw.*, vol. 108, pp. 533–543, Dec. 2018.

[139] W. Wu, Z. Qi, and L. Fuxin, "PointConv: Deep convolutional networks on 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9613–9622.

[140] Y. Liu, B. Fan, S. Xiang, and C. Pan, "Relation-shape convolutional neural network for point cloud analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8887–8896.

[141] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 922–928.

[142] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao, "SpiderCNN: Deep learning on point sets with parameterized convolutional filters," in *Proc. ECCV*, 2018, pp. 87–102.

[143] X. Yan, C. Zheng, Z. Li, S. Wang, and S. Cui, "PointASNL: Robust point clouds processing using nonlocal neural networks with adaptive sampling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5588–5597.

[144] T. Xiang, C. Zhang, Y. Song, J. Yu, and W. Cai, "Walk in the cloud: Learning curves for point clouds shape analysis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 895–904.

[145] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "A Papier-Mache approach to learning 3D surface generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 216–224.

[146] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun, "Point transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16239–16248.

[147] 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," in *Proc. ICLR*, 2022.

[148] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning From Data*. Pasadena, CA, USA: AMLBook, 2012.

[149] E. d'Eon, B. Harrison, T. Myers, and P. A. Chou, *8I Voxelized Full Bodies—A Voxelized Point Cloud Dataset*, Standard ISO/IEC JTC1/SC29, Joint WG11/WG1 (MPEG/JPEG) input document WG11M40059/WG1M74006, Jan. 2017. [Online]. Available: http://plenodb.jpeg.org/pc/8ilabs/

HANIEH NADERI (Student Member, IEEE) received the M.Sc. degree in computer engineering from the Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran, and the Ph.D. degree in artificial intelligence from the Department of Computer Engineering, Sharif University of Technology. Her current research interests include machine learning, deep learning, and adversarial attacks and defense.

IVAN V. BAJIĆ (Senior Member, IEEE) is a Professor of engineering science and the Co-Director of the Multimedia Laboratory, Simon Fraser University, Burnaby, BC, Canada. His research interests include signal processing and machine learning, with applications in multimedia signal compression, processing, and collaborative intelligence. His group's work has received the 2023 IEEE TCSVT Best Paper Award; the Conference Paper Awards at ICME 2012, ICIP 2019, MMSP 2022, and ISCAS 2023; and other recognitions (e.g., paper award finalist, top n%) at Asilomar, ICIP, ICME, ISBI, and CVPR. He is currently the Chair of the IEEE Multimedia Signal Processing Technical Committee. He was on the editorial boards of IEEE TRANSACTIONS ON MULTIMEDIA, *IEEE Signal Processing Magazine*, and *Signal Processing: Image Communication*. He is currently a Senior Area Editor of the IEEE SIGNAL PROCESSING LETTERS.