ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

TOWARDS ROBUSTNESS IN 3D POINT CLOUD ANALYSIS: NOVEL APPROACHES TO ADVERSARIAL ATTACKS AND DEFENCES

M.Sc. THESIS

Batuhan Cengiz

Department of Computer Engineering

Computer Engineering Programme

JANUARY 2025

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

TOWARDS ROBUSTNESS IN 3D POINT CLOUD ANALYSIS: NOVEL APPROACHES TO ADVERSARIAL ATTACKS AND DEFENCES

M.Sc. THESIS

Batuhan Cengiz
(504211550)

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Prof. Dr. Gözde Ünal

JANUARY 2025

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

3B NOKTA BULUTU ANALİZİNDE GÜRBÜZLÜĞE DOĞRU: ÇEKİŞMELİ SALDIRILAR VE SAVUNMALAR İÇİN YENİ YAKLAŞIMLAR

YÜKSEK LİSANS TEZİ

Batuhan Cengiz
(504211550)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Gözde Ünal

OCAK 2025

Batuhan Cengiz, a M.Sc. student of ITU Graduate School student ID 504211550 successfully defended the thesis entitled "TOWARDS ROBUSTNESS IN 3D POINT CLOUD ANALYSIS: NOVEL APPROACHES TO ADVERSARIAL ATTACKS AND DEFENCES", which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Gözde Ünal** Istanbul Technical University	.....
Jury Members :	**Assoc. Prof. Dr. Yusuf YASLAN** Istanbul Technical University	.....
	Prof. Dr. Yücel YEMEZ Koç University	.....
		.....

Date of Submission : **1 January 2025**
Date of Defense : **21 January 2025**

To my dear family,

FOREWORD

I would like to express my deepest gratitude to my advisor, Prof. Dr. Gözde ÜNAL for guidance, expertise, and inspiration. Her invaluable insights and feedback have been instrumental throughout my journey.

I am deeply thankful to my collaborators, Mert GÜLŞEN and Asst. Prof. Dr. Yusuf H. ŞAHİN. Working alongside them was both enriching and rewarding. I also extend my heartfelt thanks to many other members of ITU Vision Laboratory; Altay ÜNAL, Fırat ÖNCEL, Abdullah AKGÜL, Gülçin BAYKAL CAN, Halil F. KARAGÖZ, and İsmail ÇETİN for their friendship and encouragement during this period. Their support was a source of strength for me.

I would also like to express my gratitude to my colleagues from Control Engineering; Ertuğrul KEÇECİ, Ecem SÜMER KURUCU, and Mert Can KURUCU, for their enjoyable discussions and camaraderie throughout this journey. I am also thankful to Prof. Dr. Tufan KUMBASAR for his valuable insights and collaboration.

Additionally, I would like to extend my gratitude to my colleagues from our faculty: Kadir ÖZLEM, Erhan BİÇER, Şeyma TAKIR, Fatih BEKTAŞ, Yusuf KIZILKAYA, Erdi SARITAŞ, and Ali AZMODUEH, among others. Working alongside them has been both inspiring and enjoyable.

Most of all, I am especially grateful to my dear family: my mother Neslihan, my father Nuri, and my sister Meliha, for their unwavering support, belief in me, and encouragement throughout my life. I could not have accomplished this without their love and care.

Finally, this thesis would not have been completed without the contributions of many others who remain unmentioned above. I thank everyone for being a part of this journey.

I am financially supported by the 2210A-National Scholarship Programme for MSc Students 2022/1 from the Scientific and Technological Research Council of Turkey (TÜBİTAK), and I am sincerely grateful for this support.

January 2025

Batuhan Cengiz
Research Assistant

TABLE OF CONTENTS

ABBREVIATIONS

ANN : Artificial Neural Network
AI : Artificial Intelligence
BP4D+ : Binghamton-Pittsburgh 4D+
BU-3DFE : Binghamton University 3D Facial Expression
BU4DFE : Binghamton University 4D Facial Expression
DCT : Discrete Cosine Transform
CCN : Context-Consistency dynamic graph Network
CoMA : Convolutional Mesh Autoencoder
DDPM : Denoising Diffusion Probabilistic Models
DGIN : Dynamic Geometrical Image Network
DGCNN : Dynamic Graph Convolutional Neural Network
DUP-Net : Denoising and UPsampler Network
FGSM : Fast Gradient Sign Method
IF-Defense : Implicit Function Defense
kNN : K-Nearest Neighbor
LiDAR : Light Detection and Ranging
LBP : Local Binary Patterns
MLP : Multi-layer Perceptron
PGD : Projected Gradient Descent
PCLD : Point Cloud Layerwise Diffusion
PCT : Point Cloud Transformer
PointDP : Point Diffusion Process
SOR : Statistical Outlier Removal
SoTA : State of The Art
SVM : Support Vector Machine

SYMBOLS

$\mathbf{t}$	: Time
ε	: Perturbation boundary parameter
∇	: Gradient operator
t^*	: Truncated diffusion time
$z^{(i)}$	: Latent representation at layer i
α	: Scaling parameter in diffusion
β	: Noise variance parameter
μ	: Mean of the diffusion model
$\hat{y}$	: Predicted class label
y	: True class label
$\mathscr{L}$	: Loss function
f_θ	: Classification function
x	: Input data
x_{adv}	: Adversarial input
$\mathscr{V}_\varepsilon(x)$	: ε neighborhood of x
$\mathscr{N}$	: Gaussian distribution
Σ	: Covariance matrix
$p(x)$	: Probability density function
$q(x)$	: Data distribution
P	: Point cloud
M	: Mesh
$\triangle A_iB_iC_i$	: A triangle in the mesh
$\vec{n}_i$	: Normal vector of triangle $\triangle A_iB_iC_i$
G_i	: Barycenter of triangle $\triangle A_iB_iC_i$
L	: Loss value
k	: Number of gradient steps
n	: Number of points in the point cloud
$\alpha(t)$	: Diffusion parameter at time t
$\hat{x}(0)$	: Purified input
s_θ	: Noise model in diffusion
$f_{\mathbf{rev}}$	: Reverse diffusion process function
$g_{\mathbf{rev}}$	: Reverse diffusion noise function
ε_θ	: Noise prediction model
x_n	: Noisy point cloud at step n
$\mathscr{V}_\varepsilon$	: Perturbation region
Δ	: Change or Laplacian operator
$f_\theta^{(i)}$	: Layer i transformation in classifier
sdeint	: Stochastic differential equation integrator

LIST OF TABLES

LIST OF FIGURES

Page

TOWARDS ROBUSTNESS IN 3D POINT CLOUD ANALYSIS: NOVEL APPROACHES TO ADVERSARIAL ATTACKS AND DEFENCES

SUMMARY

This thesis explores the domain of adversarial robustness in 3D point cloud data, addressing both the offensive and the defensive aspects of adversarial interactions. The subject focuses on designing methods for adversarial attacks and defence mechanisms, particularly for applications in safety-critical domains like autonomous driving, robotics, and facial recognition.

The first part of the study introduces a novel adversarial attack method, named the ε-Mesh Attack. This method confines perturbations to the surface of 3D meshes, preserving the structural integrity of facial data. Unlike traditional approaches that operate within a 3D ε-ball, the ε-Mesh Attack reduces the optimization domain to 2D triangular planes by employing two projection methods: Central projection and Perpendicular projection. These methods ensure that adversarial manipulations remain realistic while misleading classification models. Evaluations were conducted using PointNet and DGCNN models trained on well-known 3D datasets. The results demonstrate that the ε-Mesh Attack effectively compromises model performance while maintaining the original surface integrity.

In the second part, the thesis proposes a novel defence mechanism called Point Cloud Layerwise Diffusion (PCLD). PCLD enhances robustness by employing a diffusion-based purification process that operates layer by layer within the neural network. The method involves training diffusion probabilistic models for each layer of a classifier, enabling hierarchical purification of adversarial perturbations. Suggested Point Cloud Layerwise Diffusion method was tested against state-of-the-art defence techniques and showed superior or comparable performance, particularly in defending against deeper-layer attacks.

The conclusions derived from this research emphasize the importance of preserving structural integrity during adversarial attacks and the effectiveness of layerwise purification in defending against such attacks. The findings contribute to advancing secure and resilient 3D point cloud processing methods, paving the way for their safe deployment in critical applications. Future work aims to extend these methods into the temporal domain and adapt them to handle emerging adversarial strategies effectively.

3B NOKTA BULUTU ANALİZİNDE GÜRBÜZLÜĞE DOĞRU: ÇEKİŞMELİ SALDIRILAR VE SAVUNMALAR İÇİN YENİ YAKLAŞIMLAR

ÖZET

Bu tez, 3B nokta bulutu verilerinin işlenmesinde güvenlik ve dayanıklılığı artırmayı hedefleyen yenilikçi yöntemler geliştirmektedir. Nokta bulutu verileri, sırasız ve düzensiz yapıları nedeniyle analiz edilmesi ve işlenmesi oldukça karmaşık bir veri türüdür. Bu veriler, fiziksel dünyayı üç boyutlu olarak temsil etme kabiliyetleri sayesinde, otonom sürüş, robotik, artırılmış gerçeklik ve yüz tanıma gibi alanlarda geniş bir uygulama yelpazesi bulmuştur. Ancak, bu uygulamalardaki sistemlerin yalnızca yüksek doğrulukta çalışması yeterli değildir; aynı zamanda güvenlik tehditlerine karşı dayanıklı olmaları gerekmektedir. Güvenlik açıklarına maruz kalan sistemler, hem bireysel hem de endüstriyel kullanımda büyük riskler yaratmaktadır. 3B nokta bulutu verileri üzerindeki saldırılar, sistem performansını olumsuz yönde etkileyebilir ve potansiyel olarak ciddi sonuçlara yol açabilir. Bu nedenle, bu tez kapsamında hem saldırı hem de savunma mekanizmalarını içeren kapsamlı bir araştırma yürütülmüştür.

Tezin ilk bölümünde, yeni bir saldırı yöntemi olan ε-Mesh Saldırısı tanıtılmıştır. Geleneksel ε-topu tabanlı saldırılardan farklı olarak, bu yöntem pertürbasyonları yalnızca 3B yüzeylerle sınırlandırmakta ve optimizasyon alanını iki boyutlu üçgen düzlemlere indirgemektedir. Geleneksel saldırılar genellikle yüzey deformasyonlarına neden olarak görsel olarak belirgin manipülasyonlar yaratmaktadır. Bu durum, saldırının gerçekçilikten uzaklaşmasına ve savunma sistemleri tarafından daha kolay tespit edilmesine yol açmaktadır. Buna karşılık, ε-Mesh Saldırısı, iki boyutlu üçgen düzlemler üzerinde çalışan bir optimizasyon süreci ile yüzey bütünlüğünü korurken, sınıflandırma modellerini etkili bir şekilde yanıltmayı başarmaktadır.

ε-Mesh Saldırısı, Merkezi Projeksiyon ve Dik Projeksiyon olmak üzere iki farklı teknikle uygulanmıştır. Merkezi Projeksiyon yöntemi, pertürbasyonları üçgenin kütle merkezine doğru yönlendirirken, Dik Projeksiyon yöntemi, pertürbasyonları üçgene en yakın noktaya taşımaktadır. Her iki projeksiyon yöntemi de yüzey yapısının korunmasını ve saldırının daha gerçekçi olmasını sağlamaktadır.

ε-Mesh Saldırısı, CoMA, Bosphorus ve FaceWarehouse gibi tanınmış veri kümeleri üzerinde test edilmiştir. Bu veri kümeleri, 3B yüzey verilerinin yanı sıra yüz ifadeleri gibi karmaşık geometrik yapıların temsilinde kullanılmaktadır. Deneyler, ε-Mesh Saldırısı'nın, sınıflandırma modellerini yanıltmada etkili olduğunu ve yüzey bütünlüğünü koruduğunu göstermiştir. Örneğin, DGCNN modeli üzerinde CoMA veri kümesi kullanılarak yapılan testlerde, geleneksel saldırılar ile %0.5 oranına kadar çıkan yüzey deformasyonu, ε-Mesh Saldırısı ile %0.15 seviyesine düşürülmüştür. Ayrıca, ε-Mesh Saldırısı'nın sınıflandırma doğruluğunu ortalama %97 oranında

azalttığı gözlemlenmiştir. Bu sonuçlar, özellikle yüz tanıma gibi yüzey bütünlüğünün kritik önem taşıdığı uygulamalarda, ε-Mesh Saldırısı'nın oldukça etkili bir yöntem olduğunu göstermektedir. Yüzey deformasyonlarının düşük seviyede tutulması, saldırının görsel gerçekçiliğini artırmakta ve tespit edilmesini zorlaştırmaktadır.

Tezin ikinci bölümünde, 3B nokta bulutu verilerinin saldırılara karşı dayanıklılığını artırmak amacıyla geliştirilen Katman Tabanlı Nokta Bulutu Difüzyonu (PCLD) yöntemi tanıtılmıştır. Difüzyon yöntemleri, başlangıçta generatif modeller olarak tasarlanmıştır ve bir veri kümesinin temel dağılımını öğrenerek yeni örnekler üretmek için kullanılmıştır. Ancak bu çalışmada, difüzyon modelleri, yüksek boyutlu ara katman dağılımlarını öğrenmek ve adversarial pertürbasyonları temizlemek amacıyla uyarlanmıştır. Bu mekanizma, giriş verisinden başlayarak sinir ağının her katmanındaki ara temsilleri difüzyon yoluyla düzenler ve temizler. Böylece, hem giriş seviyesinde hem de daha derin katmanlarda meydana gelen saldırı etkileri etkili bir şekilde azaltılır. Bu yöntem, sınıflandırıcı modellerin her bir katmanı için difüzyon olasılık modelleri eğitilerek, saldırıya uğramış verilerin etkisinin hiyerarşik bir temizleme süreci ile azaltılmasını sağlamaktadır.

PCLD yöntemi, ModelNet40 veri kümesi üzerinde yapılan kapsamlı deneylerle değerlendirilmiştir. Bu deneylerde, PointNet++, DGCNN ve CurveNet gibi modern sınıflandırıcılar üzerinde, PCLD'nin savunma performansını artırdığı görülmüştür. Örneğin, PointNet++ modeli üzerinde yapılan testlerde, PCLD'nin saldırıya uğrayan modellerde doğruluk oranını %89 seviyesine yükselttiği gözlemlenmiştir. Daha karmaşık senaryolarda ise, DGCNN ve CurveNet modellerinde PCLD kullanıldığında doğruluk oranının %92'den %96'ya çıktığı rapor edilmiştir. PCLD'nin katmanlar arası yüksek boyutlu dağılımları öğrenerek, saldırıların etkilerini temizleme kabiliyeti, yöntemin farklı model mimarileri ve saldırı senaryoları için geniş bir uygulanabilirlik sunduğunu göstermektedir.

Araştırma sonuçları, hem saldırıların hem de savunma mekanizmalarının tasarımında yapısal bütünlüğün korunmasının kritik önem taşıdığını vurgulamaktadır. ε-Mesh Saldırısı, 3B yüzeylerde gerçekçi manipülasyonlar yapma kabiliyetine sahipken, PCLD yöntemi, katman bazında gerçekleştirdiği temizleme süreçleri ile saldırılara karşı dayanıklılığı artırmaktadır. Bu iki yöntem, güvenlik ve dayanıklılık gereksinimlerini karşılayarak 3B nokta bulutu işleme alanında yenilikçi bir çerçeve sunmaktadır.

Bu tez, yalnızca mevcut zorlukları ele almakla kalmayıp, gelecekteki araştırmalara ve uygulamalara da güçlü bir temel sunmaktadır. Gelecekteki çalışmalar, önerilen yöntemlerin zaman boyutuna genişletilmesi ve dinamik 4B verilere uygulanması üzerinde yoğunlaşabilir. Özellikle, zamanla değişen verilerin analizi, bu yöntemlerin etkinliğini artırarak uygulama alanlarını genişletecektir. Bunun yanı sıra, önerilen yöntemlerin daha büyük veri kümelerine ve daha karmaşık modellere ölçeklenebilirliği üzerinde çalışılması gerekmektedir. Ayrıca, yeni ortaya çıkan saldırı stratejilerine uyum sağlamak ve bu tehditlere karşı etkili çözümler geliştirmek için adaptif savunma mekanizmalarının tasarlanması, bu alanda önemli bir araştırma alanı olarak öne çıkmaktadır.

Önerilen yöntemlerin pratikteki uygulanabilirliği, özellikle güvenlik açısından kritik görevlerde büyük bir potansiyele sahiptir. ε-Mesh Saldırısı, yüzey deformasyonlarını

en aza indirerek gerçek dünyadaki uygulamalar için daha gerçekçi ve tespit edilmesi zor saldırılar sunmaktadır. Örneğin, yüz tanıma sistemlerinde güvenlik açıklarını değerlendirmek için bu saldırı yöntemi kullanılabilir. Benzer şekilde, PCLD yöntemi, saldırılara karşı dayanıklı sinir ağları tasarlamak için güçlü bir savunma mekanizması sağlamaktadır. Bu yöntem, yalnızca statik veri senaryolarında değil, aynı zamanda otonom araçlar gibi sürekli değişen ortamlarda çalışan sistemlerde de etkili bir şekilde uygulanabilir. Yüksek boyutlu ara temsillerin temizlenmesi, dinamik sistemlerin güvenliğini artırmada kilit bir rol oynamaktadır. Bu tezde sunulan çözümler, yapay zekanın daha güvenli ve dayanıklı bir şekilde gerçek dünya problemlerine uygulanmasına yönelik önemli bir adım olarak değerlendirilmektedir.

Sonuç olarak, bu tez, 3B nokta bulutu işleme alanında güvenlik ve dayanıklılık açısından yenilikçi çözümler sunmakta ve teknolojik ilerlemelere önemli bir katkı sağlamaktadır. Önerilen yöntemler, 3B verilerin güvenliğinin sağlanmasında hem teorik hem de pratik olarak değerli bir adım niteliğindedir.

1. INTRODUCTION

1.1 Overview

Deep neural networks have demonstrated remarkable efficacy in 3D point cloud processing, building on their success in 2D image tasks. They are widely applied in domains such as classification and segmentation [1]–[4], registration [5,6], and scene reconstruction [7,8]. The adoption of LiDAR and related technologies has further solidified 3D point clouds as foundational elements in safety-critical applications, including autonomous driving [9]–[11], airborne scanning [12], and industrial automation [13]. This growing reliance underscores the pressing need for models that are not only accurate but also robust.

Thus, adversarial attacks [14] have emerged as a critical challenge to test the resilience of deep learning-based methods. These attacks generate subtle yet impactful perturbations that mislead machine learning models, compromising their reliability in critical applications. While adversarial robustness has been extensively studied in the two-dimensional (2D) domain [15], research in the three-dimensional (3D) domain is still in its infancy. The unique characteristics of point cloud data—such as irregularity, sparsity, and unordered structures—pose additional challenges, necessitating tailored solutions for both offensive and defensive strategies.

An equally important aspect of robustness research lies in defending against advancing adversarial attacks. Although the number of defence techniques for point cloud data is currently limited, progress is being made. Unlike in 2D data, developing robust defences for 3D point clouds requires addressing unique constraints and challenges inherent to their structure. This calls for innovative methods that not only detect and mitigate adversarial threats but also adapt to the complex characteristics of 3D data.

1.2 Contributions

This thesis seeks to address these challenges by exploring both adversarial attacks and defence mechanisms tailored for 3D point cloud data. The contributions of this work are organized into two primary components, detailed in the subsequent chapters:

- **ε-Mesh Attack:** A novel surface-based adversarial attack method that introduces perturbations confined to the mesh surface, preserving the structural integrity of 3D facial meshes. Unlike conventional approaches that operate within 3D ε-ball constraints, the ε-Mesh Attack ensures effective and realistic perturbations, particularly in applications like facial expression recognition, where maintaining the natural appearance of the face is crucial.
- **Point Cloud Layerwise Diffusion (PCLD):** A diffusion-based defence mechanism that purifies adversarial perturbations layer by layer within a neural network. PCLD significantly enhances the robustness of point cloud classification systems against various attacks. Its adaptability to diverse architectures and attack scenarios ensures superior performance across multiple experimental setups.

The thesis is structured as follows:

- Chapter 2 introduces the ε-Mesh Attack, detailing its methodology, evaluation, and implications for 3D facial expression recognition.
- Chapter 3 presents the PCLD defence mechanism, discussing its theoretical foundations, experimental results, and comparisons with state-of-the-art techniques.
- Chapter 4 concludes the thesis with a summary of contributions, a discussion of limitations, and directions for future research.

By addressing both the offensive and defensive aspects of adversarial robustness in 3D point cloud data, this thesis provides a comprehensive framework for advancing the field of secure point cloud processing. The proposed methods are anticipated to have significant implications for safety-critical applications, paving the way for more resilient systems in real-world scenarios.

2. ADVERSARIAL ATTACKS: E-MESH ATTACK

2.1 Introduction to Adversarial Attacks

Adversarial attacks aim to generate data examples with imperceptible yet effective small perturbations to mislead vision models. There have been many studies on designing adversarial attacks for 2D [16,17] and 3D [18]–[21] data. The crucial difference between 2D and 3D attacks is that 3D attacks perturb the point positions while 2D attacks change the pixel values, keeping the same positions. By performing an attack on a point cloud, a slightly jittered version of the original point cloud is constituted which fallaciously makes the network predict the wrong class. While yielding great results in terms of accuracy by confusing deep learning models, existing methods fall short on preserving the surface structure, especially in 3D facial expression data since small deformations can cause the overall expression to change.

Preserving the surface structure can be crucial since capturing a facial expression via a 3D sensor could be done by sampling points over the face surface that is represented in the mesh. Thus, for many applications, the point cloud and the related mesh are available together [22]–[26]. However, previous 3D attack methods do not consider mesh data. Following these ideas, we are motivated to develop an adversarial attack method that preserves the face surfaces on 3D point clouds by utilizing available mesh data.

In this thesis, we propose a 3D adversarial attack method for point clouds called ε-*Mesh attack*, which preserves the face surface by strictly keeping adversarial points on the mesh by projecting perturbations onto mesh triangles using two different methods: central and perpendicular projections. We have also parameterized our attack method by ε to scale our attack boundaries into similar triangles as shown in Figure 2.1. We evaluate our attack method on 3D facial expression recognition models and show that compared to other attacks, our attack does not cause as much surface deformation. This creates potential use cases for our method in many cases

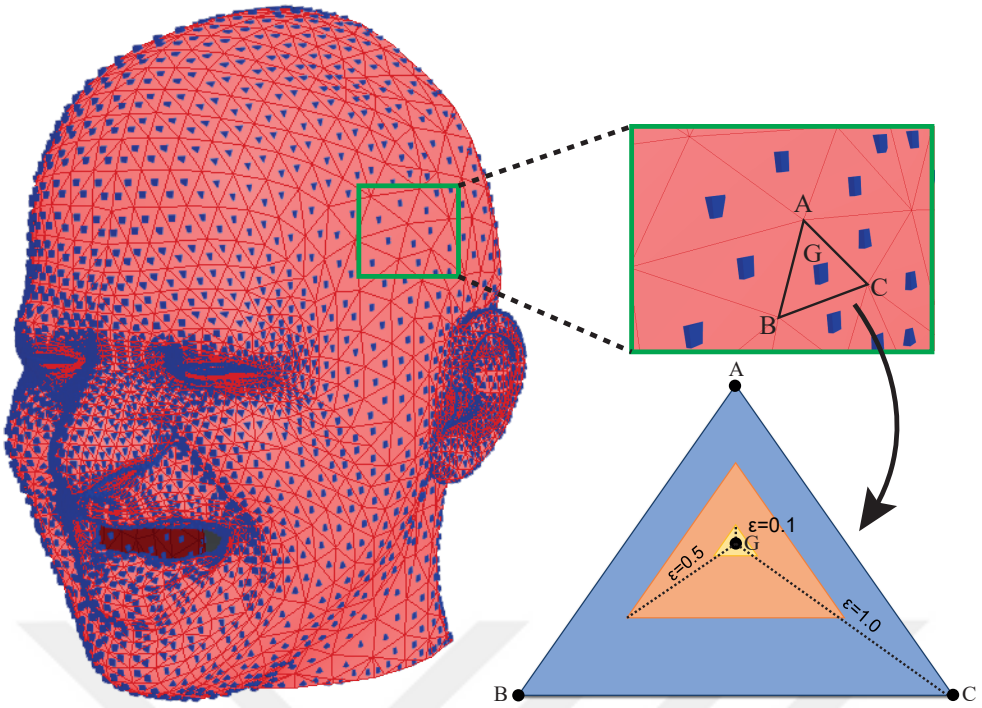

Figure 2.1 : An example Face mesh from CoMA dataset and the suggested triangular bounds for the surface preserving white box attack scaled by parameter ε.

such as safety-critical application of facial expression recognition like human computer interaction or classification in the wild with unsafe data.

2.2 Related Work

Facial Expression Recognition. In the classical facial expression recognition problem, following Ekman's [27] studies, initial methods investigated 2D images or image sequences to map the inputs to basic emotional expressions like anger, disgust and fear [28,29]. In [30], DCT, LBP, and Gabor filters are used to extract facial features, then an SVM is trained to classify the expressions. In [31], a neural network is trained with facial landmark trajectories and geometric features. In [32], a deep metric learning-based training procedure is presented. In [33], a generative network that outputs faces with no expression from expressive face images is trained, and then using the features from this network, a classification is done.

With the emergence of deep learning-based models that directly process point cloud data [34]–[36], there have been significant developments in 3D facial expression

recognition. These permutation invariant point cloud-based models can learn to represent the structure of point cloud face data. In [37], geometrical images for 3D face sequences are fed to Dynamic Geometrical Image Network (DGIN) which combines short-term and long-term information. In [38], histograms of oriented gradients and optical flows are utilized to find correspondence in 3D data and classify facial expressions. In [39], a multi-view transformer architecture is proposed for 3D/4D facial expression recognition. For these tasks, a variety of 3D datasets are accessible, including BU-3DFE [40], FaceScape [23], along with 4D datasets like BU4DFE [41] and BP4D+ [42]. For a further reading on 3D facial expression recognition, detailed surveys on this topic can be investigated [37,43].

Adversarial Attacks. Adversarial attacks are methods that generate adversarial data perceptually similar to the original samples, to deceive deep learning models. Szegedy et al. [44] pioneered to demonstrate the vulnerability of neural networks to adversarial examples and drew attention to the potential security risks in safety-critical applications. In the 2D image domain, many attack methods have been proposed for deep learning models. Goodfellow et al. [14] argued that generation of adversarial examples are possible due to locally linear nature of neural networks and proposed an attack method referred to as Fast Gradient Sign method. This method allows generation of adversarial examples with one step towards the direction of gradient to increase the loss. Madry et al. [16] demonstrated Projected Gradient Descent (PGD) attack by applying multiple gradient steps in a bounded area to find local minimum. Another attack algorithm called C&W attack was proposed by Carlini et al. [45] which optimizes an objective function of distance between original and adversarial examples subject to the constraint of changing the classification of image in order to find the adversarial perturbation.

After their success on 2D images, adversarial attacks are extended to 3D point cloud models. Xiang et al. [18] pioneered the extension of adversarial attack methods to 3D point cloud models by generating 3D adversarial point cloud examples using their proposed methods, adversarial point addition and perturbation. Yang et al. [19] proposed pointwise gradient perturbation, point attachment and detachment methods by leveraging gradient-based adversarial attack algorithms. Zhang et al. [21] proposed

an attack method that minimizes combined loss of mesh edge distances and sampled point cloud Chamfer distances to directly create adversarial meshes. Huang et al. [20] suggested differentiable rotation and translation matrices to create adversarial perturbations that lie on estimated surfaces. Projected Gradient Descent (PGD) attack in 3D [46] iteratively move the points towards gradient directions to maximize loss where the total translation is limited to a spherical ε-ball in L_2 and L_∞ distance metrics. Inspired by PGD attack, we have proposed two different projection mechanisms to limit the adversarial perturbations on the 2D mesh surfaces rather than 3D ε-ball.

2.3 A Novel Method: Epsilon-Mesh Attack

Meshes & Point Clouds. A mesh M is as a set of v-gons in the d-dimensional space. In standard mesh processing, the mesh is defined in the space of triangles ($v = 3$) in 3D space ($d = 3$). Thus, a mesh could be defined as $M = \{t_1, ..., t_i, ..., t_n\}$ where t_i represent the triangles. From each triangle, a sampling process could be performed to create the point cloud $P = \{p_1, ..., p_i, ..., p_n\}$ where p_i represents the point sampled from t_i. We should also note that this sampling process is independent for each triangle.

Moreover, We can define each triangle as $t_i = \triangle A_i B_i C_i$ where vertices are $A_i, B_i, C_i \in R^3$ and the barycenter is $G_i = (A_i + B_i + C_i)/3 \in R^3$. We also denote the normal vector as $\vec{n}_i \perp \triangle A_i B_i C_i$.

Assuming that, each triangle has a uniform probability density function for the sampled point p_i, $E[p_i] = G_i$ is the most representative point of t_i and could be used in sampling. We also followed this method to initialize our point clouds from the meshes.

Adversarial Perturbations. Given a classification function $f_\theta(P)$, its prediction $\hat{K}$, and the ground truth label K we define the adversarial perturbation $\vec{\nabla} \in R^3$ as the gradient ascent step with respect to loss $L = \|K - \hat{K}\|$ as,

$$\vec{\nabla} = \frac{dL}{df_\theta} \frac{df_\theta}{dP}. \tag{2.1}$$

Projection Methods. To keep the adversarial perturbation of p_i on the triangle $\triangle A_i B_i C_i$, we have to project $\vec{\nabla}$ back into the plane using the formula:

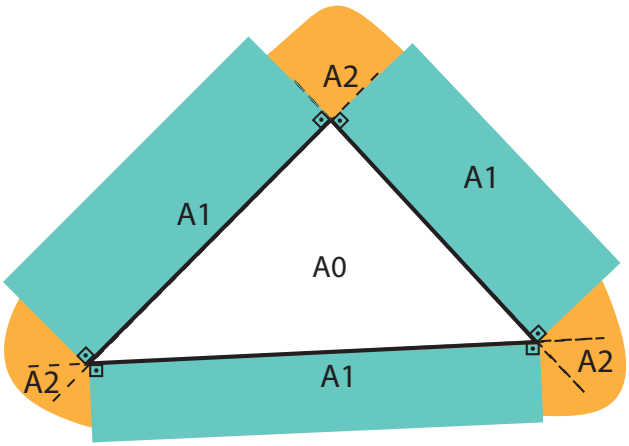

Figure 2.2 : Division of the area near a selected triangle to calculate perpendicular projection.

$$\vec{\nabla}_s = \vec{\nabla} - \frac{\vec{\nabla} \cdot \vec{n}_i}{\vec{n}_i \cdot \vec{n}_i} \vec{n}_i \tag{2.2}$$

where $\hat{p}_i = p_i + \alpha \vec{\nabla}_s$ is the projected position of point p_i and α is the attack learning rate. It is possible for $\hat{p}_i$ to lie outside of the region bounded by triangle $\triangle A_i B_i C_i$. Thus, we propose the following two methods to project $\hat{p}_i$ back into the boundary of $\triangle A_i B_i C_i$ if it is not already in the triangle.

Central Projection. Firstly, we suggest projecting point $\hat{p}_i$ that is outside of the triangular region $\triangle A_i B_i C_i$ by the line that directs into barycenter G_i. Thus, the projection $p_{i,cent}$ is defined as the intersection point of line segment $\overline{\mathrm{G_i\hat{p}_i}}$ and intersecting edge of the triangle $\triangle A_i B_i C_i$ as given in equation 2.3.

$$p_{i,cent} = \begin{cases} \hat{p}_i, & \hat{p}_i \in \triangle A_i B_i C_i \\ \\ \triangle A_i B_i C_i \cap \overline{\mathrm{G_i\hat{p}_i}}, & \text{otherwise} \end{cases} \tag{2.3}$$

Perpendicular Projection. Secondly, we suggest the perpendicular projection method where the area near the triangle is divided into 7 parts as given in Figure 2.2. If $\hat{p}_i$ is in the $A0$ region, it is not projected. If $\hat{p}_i$ is inside an $A1$ region, it is projected to

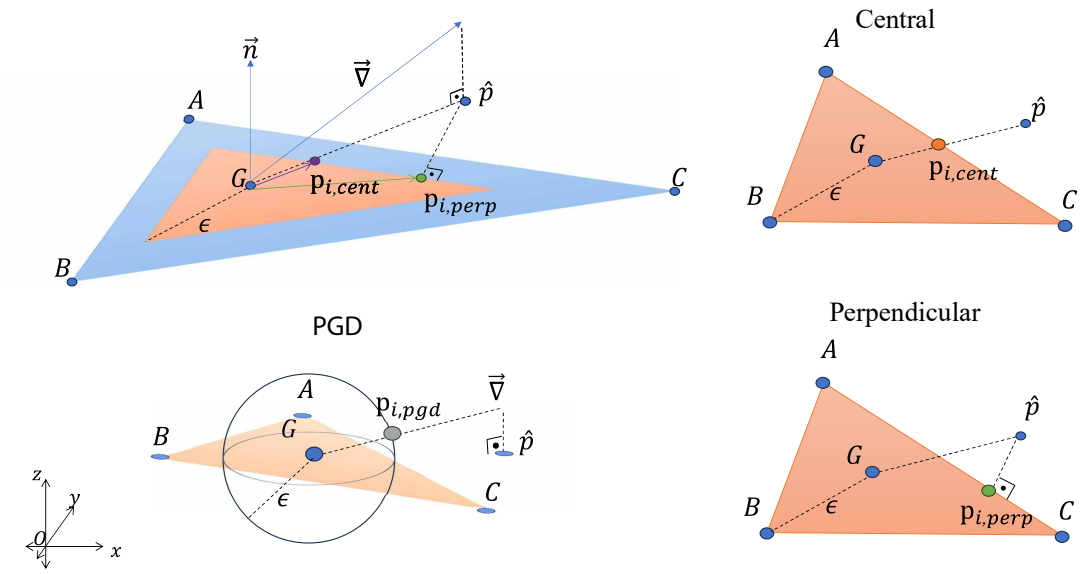

Figure 2.3 : Projection example for adversarial perturbation ∇ (left). Different projection methods in right: PGD, Central and Perpendicular from top to bottom.

the nearest edge of the triangle. For the $A2$ regions, the nearest vertex of the triangle is selected. To formally define, if the projected point is outside of the triangle, it is projected to the closest point of the triangular area as shown in equation 2.4.

$$p_{i,perp} = \begin{cases} \hat{p}_i, & \hat{p}_i \in \triangle A_i B_i C_i \\ \underset{x \in \triangle A_i B_i C_i}{\arg\min} \|x - \hat{p}_i\|, & \text{otherwise} \end{cases} \tag{2.4}$$

A comparative demonstration of PGD and ε-Mesh projections is given in Figure 2.3. The proposed projection methods are applied on each step of gradient ascent optimization process. The triangle $\triangle A_i B_i C_i$ can be scaled by an epsilon paramater $\varepsilon \in [0,1]$ around barycenter G_i to scale down the projection area.

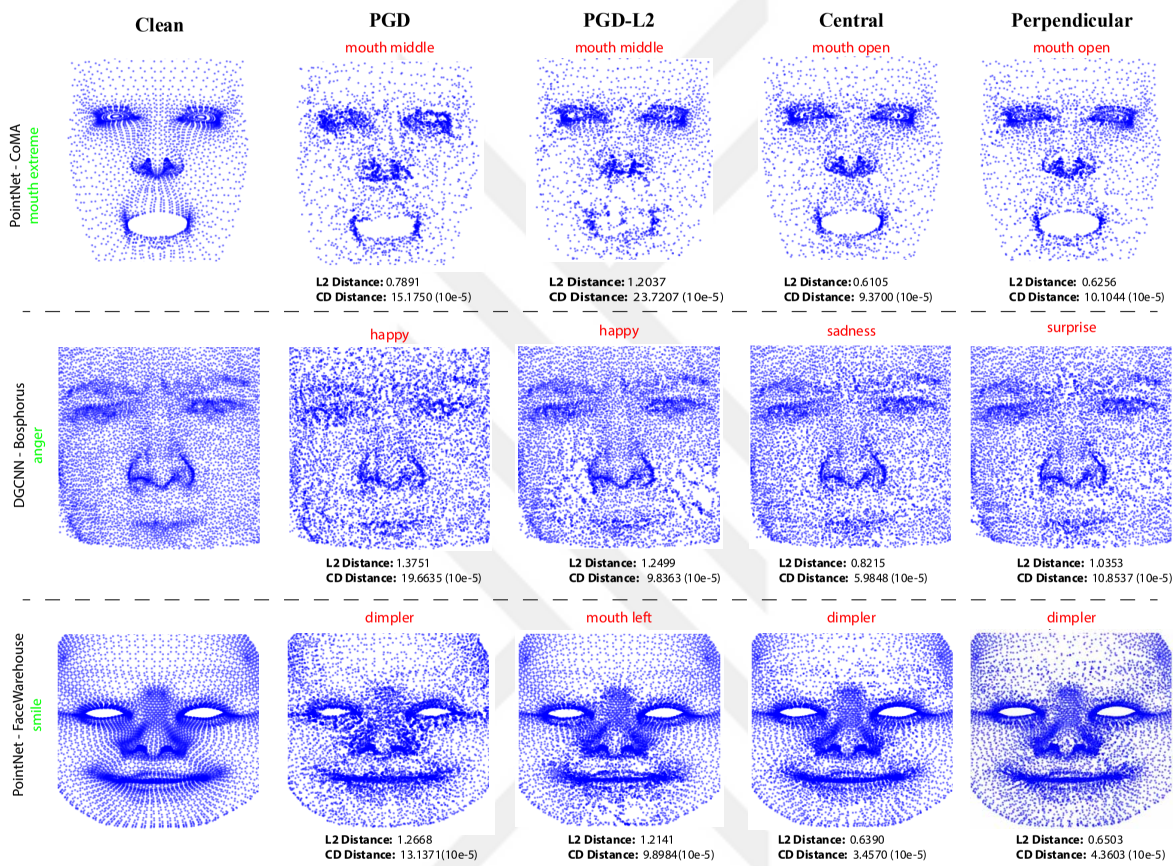

Figure 2.4 : Example front-viewed point cloud images from different datasets are given at each row. Predictions obtained from model denoted at left are written at the top of each image with green and red colors for correct and incorrect predictions respectively. The columns represent a clean face, and its attacked versions by PGD, PGD-L2, ε-mesh central projection, and ε-mesh perpendicular projection from left to right. Also, distances between the attacked and clean point clouds are denoted below with each sample for L_2 and Chamfer distances.

2.4 Experimental Results

Datasets. We have conducted our experiments on three well-known facial expression datasets: CoMA [26], Bosphorus [47] and FaceWarehouse [24]. We have focused on 3D datasets where the mesh data is available and for the datasets like Bosphorus where the underlying mesh is missing, we have used Poisson surface reconstruction [48] as a simple triangular mesh estimator.

CoMA [26] is a publicly available 4D facial expression mesh dataset. CoMA contains 12 facial expression (bare teeth, cheeks in, eyebrow, high smile, lips back, lips up, mouth down, mouth extreme, mouth middle, mouth open, mouth side, mouth up) sequences of 12 different subjects. Each subject performs facial expressions over a series of frames. For each sequence, we selected the peak frame and 4 more adjacent frames that are successive to the denoted frame. We randomly selected 10 subjects (600 meshes) for training and 2 subjects (120 meshes) for testing our models.

Bosphorus 3D Database [47] consists of various action units and emotions from 105 people. It contains both the RGB images and facial 3D coordinates for each pixel. The dataset has 6 emotion classes: anger, disgust, fear, happy, sad and surprise. We have split the dataset as 91 subjects for training and 14 subjects for the test set.

FaceWarehouse dataset [24] contains reconstructed 3D mesh data of 150 individuals with 20 different facial poses (neutral, mouth stretch, smile, brow lower, brow raiser, anger, jaw left, jaw right, jaw forward, mouth left, mouth right, dimpler, chin raiser, lip puckerer, lip funneler, sadness, lip roll, grin, cheek blowing, eyes closed). We selected data from 126 individuals for training while preserving the other 24 for the test.

Preprocessing. We initially selected center of gravity for each triangle to form our point clouds from meshes. Then, a unit sphere scaling is applied. Subsequently, for the datasets containing the full head meshes, namely CoMA and FaceWarehouse, the back

halves of the subjects' heads are deleted. After completing the preprocessing steps, it was observed that each sample from the Bosphorus, CoMA, and FaceWarehouse datasets consisted of approximately 12000, 4000, and 12000 point/mesh pairs, respectively.

Experimental Results. We have compared our two suggested white box point cloud attack methods against Projected Gradient Descent method [46] with L_∞ and L_2 metrics, namely PGD and PGD-L2 respectively.

Table 2.1 : Adversarial attack results on CoMA, Bosphorus, and FaceWarehouse datasets.

					CoMA		Bosphorus		FaceWarehouse	
Model	Attack	Eps	Alpha	Steps	Clean Acc (%)	Attacked Acc (%)	Clean Acc (%)	Attacked Acc (%)	Clean Acc (%)	Attacked Acc (%)
DGCNN [36]	PGD	0.01	0.0004	250	79.17	0.0	69.04	0.0	98.96	0.0
	PGD-L2	1.25	0.05			0.0		0.0		0.0
	(**Ours**) ε-mesh Central	1.00	0.10			5.83		3.57		0.21
	(**Ours**) ε-mesh Perpendicular	1.00	0.10			0.83		0.0		0.0
PointNet [34]	PGD	0.01	0.0004	250	71.67	0.0	60.71	0.0	88.96	0.0
	PGD-L2	1.25	0.05			0.0		0.0		0.0
	(**Ours**) ε-mesh Central	1.00	0.10			0.83		19.04		14.38
	(**Ours**) ε-mesh Perpendicular	1.00	0.10			0.0		7.14		1.67

Table 2.1 illustrates the classification performances of DGCNN [36] and PointNet [34] against PGD, PGD-L2, and our suggested attack methods. For all three aforementioned datasets, both PGD and PGD-L2 attacks were able to effectively degrade model performance to zero by disrupting the underlying structure, as their primary aim is not to preserve facial structural integrity. For our ε-mesh attacks, perpendicular attack achieves less than 2% accuracy in all cases except for prediction with PointNet in Bosphorus dataset. We assume that being due to errors in mesh estimation algorithm. On the other hand, ε-mesh central attack achieves less than 20% accuracy in all cases. Overall, our perpendicular attack performs better according to numerical results.

In the first row of Figure 2.4, we show that PGD attack heavily corrupts the data while PGD-L2 moves points over the empty regions such as the mouth. However, both of our methods keep the mouth region clear since it preserves the surface. For the same figure,

as seen in the examples of second and third rows, PGD-L2 attack creates empty regions on the face surface during adversarial attack. For PGD examples on the same rows, outputs are rather noisy which is shown by high L2 and Chamfer distances to clean point clouds. Partial side views around the nose region is given in Figure 2.6. As seen in figure, the PGD attack pushes most of the points above surface while PGD-L2 shifts a few points to out of the surface furthermore. However, central and perpendicular ε-mesh attacks are the only attacks that keep every point on the surface even after the adversarial perturbations.

Time complexity. We have reported execution time for each model-dataset pair in Table 2.2. Our experiments show that the suggested two attack methods cost almost the same in terms of time, compared to other gradient based attacks like PGD. We apply a projection to each calculated gradient vector in each step. Thus, if we denote number of steps with k and number of points with n, our time complexity would be $O(nk)$. Since CoMA dataset has less points than the other two, it costs less time to attack this dataset. Even though both PGD and ε-mesh attack methods take equal time cost per step, the convergence rates differ as seen in Figure 2.7.

Table 2.2 : Adversarial attack execution times for 250 steps.

Attack Method	PointNet Execution Times (seconds)		
	FaceWarehouse	CoMA	Bosphorus
Perpendicular	1.80	1.37	1.75
Central	1.95	1.47	1.90
PGD	1.71	1.13	1.63
PGDL2	1.70	1.12	1.67
Attack Method	DGCNN Execution Times (seconds)		
	FaceWarehouse	CoMA	Bosphorus
Perpendicular	25.17	5.85	24.35
Central	25.52	5.98	24.59
PGD	24.96	5.74	24.31
PGDL2	25.15	5.73	24.24

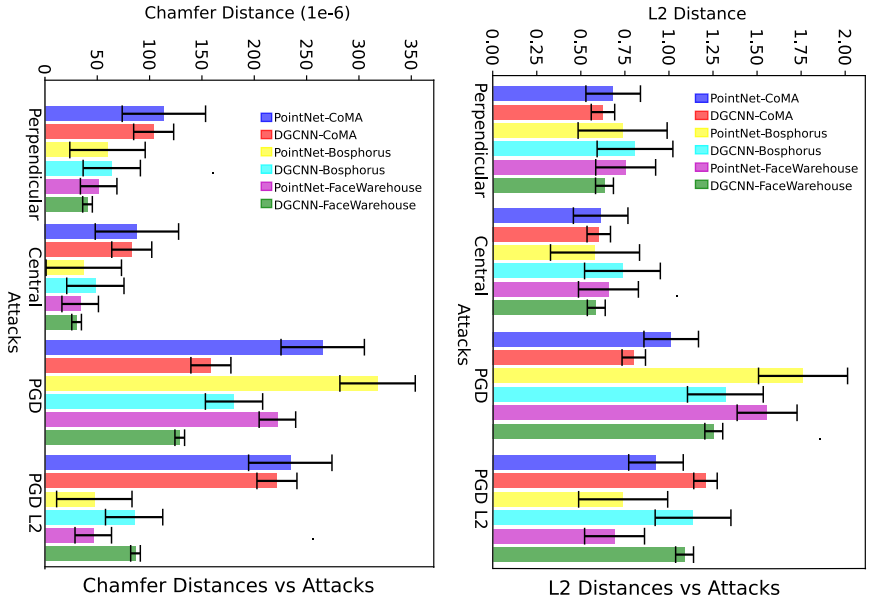

Figure 2.5 : L2 and Chamfer distances between the original data and attacked data.

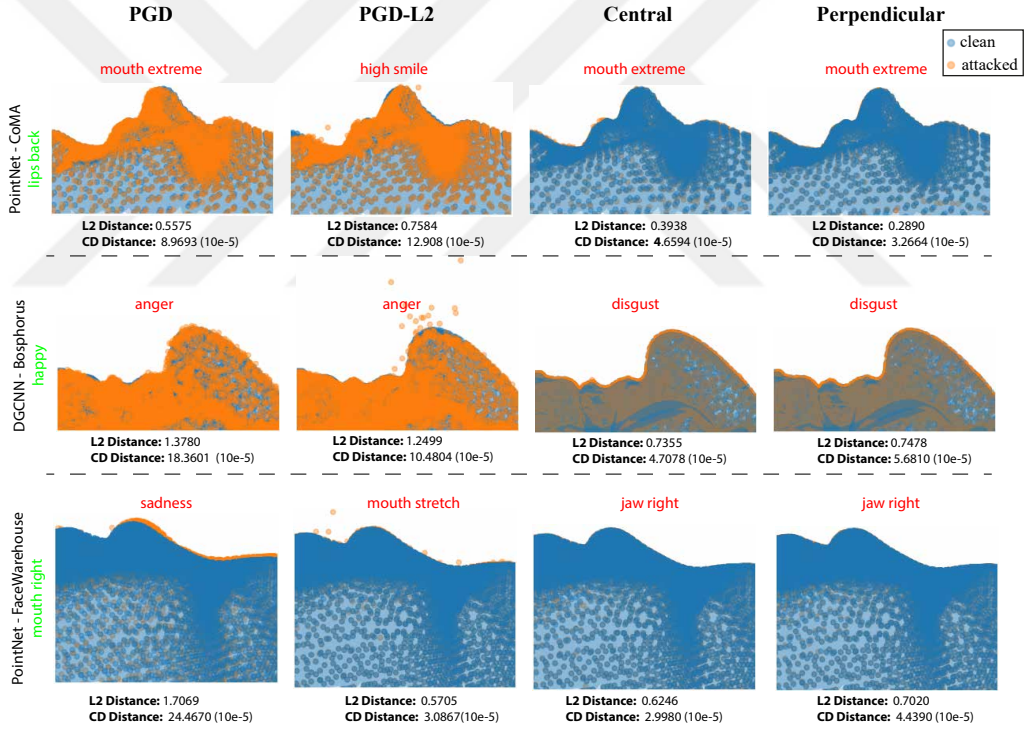

Figure 2.6 : A side view of an example face mesh surface (blue triangles), clean point clouds (blue points), and attacked point clouds (orange points) for some samples from Coma, Bosphorus and FaceWarehouse datasets. Original network predictions (green texts) and attacked predictions (red texts) are also given.

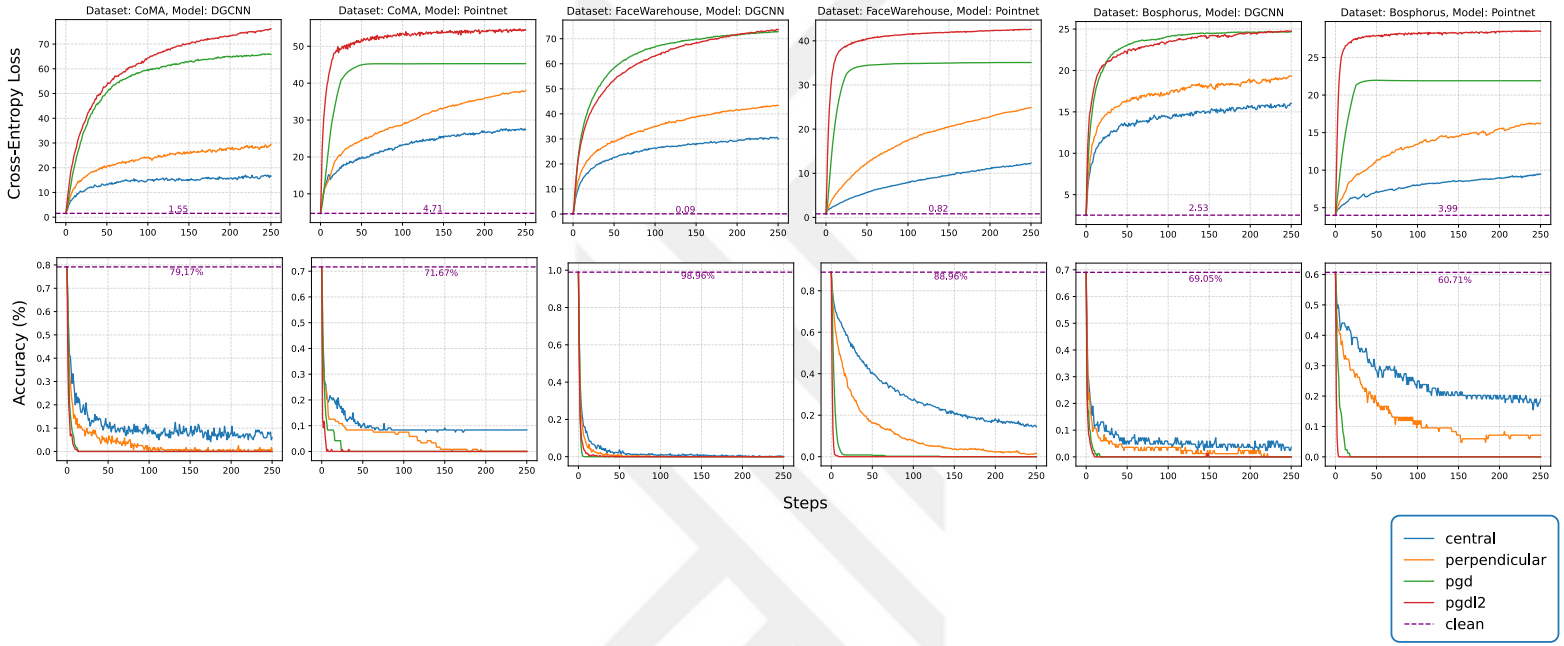

Figure 2.7 : Adversarial attack loss results for varying number of steps. The left and right columns show results on DGCNN and PointNet respectively.

Perturbation distance. We evaluated distances between the attacked point clouds and the clean point clouds using L_2 and Chamfer distances. Normally, it is not possible to calculate the L_2 distance between two point sets due to the unordered nature of the point clouds. However, we already have the correct point correspondes and perturbation vectors to calculate the L_2 distance. In Figure 2.5, we have reported the distances for 6 different model-dataset pairs using bar notation to show means and standard deviations. For L_2 metric, our suggested perpendicular and central ε-mesh attacks have a distance of 0.71 and 0.63 respectively, while PGD and PGD-L2 attacks have 1.28 and 0.97. For Chamfer distance, results are as following: 71.53 for perpendicular, 53.36 for central, 212.22 for PGD, 120.21 for PGD-L2. Allover, it is clear that our suggested methods perturbs the points at least 1.5 times less than the PGD based methods.

Ablation Study. We test out our ε parameter which scales down the triangles to limit the perturbation into the center. In Figure 2.8, we reported results for $\varepsilon \in \{0.1, 0.25, 0.5, 1.0\}$ while keeping the number of steps as 250 to show that scaling up the mesh boundaries increases the performance of attack exponentially. It is also clear that perpendicular projection performs well even under low values of ε such as 0.1.

Also in Figure 2.7, we showed our experimental results for various number of steps in all attack methods. As iteration count increases, all methods perform better since we have an iterative optimization setup. PGD and PGD-L2 methods converge to 0% accuracy just after 5 steps as they have softer perturbation boundaries as it can be deduced from the corruptions in Figure 2.6.

Our experimental results showed that the ε-mesh Attack is capable of significantly reducing the performance of sophisticated deep learning models like DGCNN and PointNet. While our method does not achieve the aggressive performance degradation seen in methods like PGD, it offers a unique advantage in its subtlety and surface preservation. This finding is significant in highlighting the trade-off between

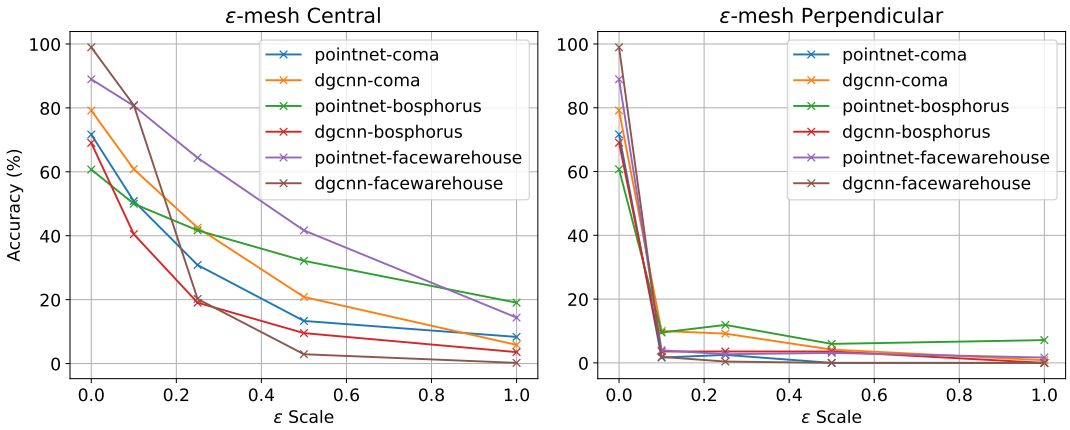

Figure 2.8 : Adversarial attack accuracy results for different epsilon scales. The left and right graphs show results on ε-mesh Central and Perpendicular attacks respectively.

aggressive attack strategies and the need for realistic, undetectable alterations in certain applications.

3. ADVERSARIAL DEFENCES: POINT CLOUD LAYERWISE DIFFUSION

3.1 Introduction to Adversarial Defences

Despite the vulnerabilities of 2D deep learning methods being broadly investigated [15] and many defence mechanisms are suggested [16,49], the study of defences for 3D point clouds is considerably less extensive. Similar to their 2D counterparts, adversarial attacks are a widespread application used to evaluate the robustness of 3D networks. These attacks, which can be in the form of black box [50] or white box attacks [18,46], generate adversarial examples based on the network. Adversarial examples generally look similar to the human eye but are misclassified by the network. Instead of changing pixel values as in 2D, adversarial attacks in 3D focus on changing positions and counts of the points in a given sample.

Many defence mechanisms like DUP-Net [51], IF-Defense [52] and PointDP aimed to purify the point cloud by cleaning the adversarial noise at the input level. On the other hand, the only solution to deal with the adversarial samples in feature domain is CCN [53] which is a novel neural network having denoiser blocks after each convolutional layer. However, CCN cannot be extended for other classifier architectures as a defence method. Inspired by the aforementioned studies, we suggest **PCLD**, a diffusion-based layerwise purifier algorithm that can operate on high dimensional data without the hassle of retraining the neural network. We have demonstrated the overview of PCLD in Figure 3.1 and explained it in detail later in Method section. Experiment results on the defences of widely used point cloud classification networks like PointNet [1],

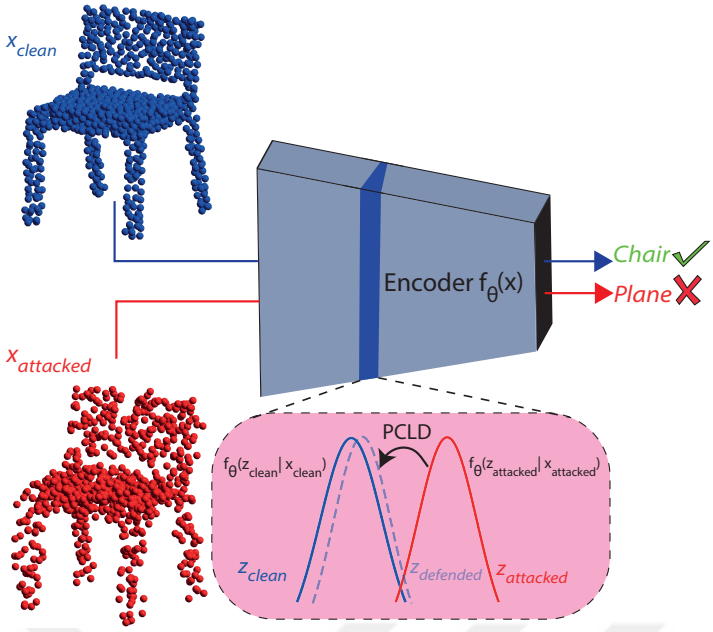

Figure 3.1 : Overview of PCLD. In PCLD, the main focus is denoising the adversarial layer features back into the clean layer features with a diffusion-based purification.

DGCNN [36], and PCT [9] showed that the proposed defence method achieved results that are comparable to or surpass those of existing methodologies.

3.2 Related Work

Adversarial attacks. Point cloud based deep learning models have shown great success and continue to improve [1,35,36,54]–[56]. Despite their success, they have been discovered to be vulnerable against carefully crafted adversarial examples [44]. These adversarial data examples are indistinguishable from clean data examples to human perception, yet they are proven to completely degrade the accuracy of deep learning models. There exist numerous adversarial attack methods for creating adversarial examples. Goodfellow et al. [14] proposed the Fast Gradient Sign Method (FGSM), which perturbs the input image in a single step toward the gradient that maximizes the loss, thereby creating an adversarial image. Kurakin et al. [17]

improved this idea by using multiple steps of FGSM to perform an adversarial attack. Madry et al. [16] demonstrated Projected Gradient Descent (PGD) attack which limits the iterative gradient steps with a box constraint. Carlini et al. [45] proposed optimization-based C&W attacks.

The success of adversarial attacks on 2D models leads to studies adapting adversarial attacks to 3D deep learning models. Xiang et al. [18] were the first to demonstrate that 3D point cloud models are also vulnerable to adversarial attacks. They presented point perturbation and generation attacks that utilize the C&W attack method to perturb points. Many of the following studies proposed the adaptation of different 2D adversarial attacks to 3D point cloud models: Liu et al. [57] extended FGSM and Iterative FGSM (IFGSM) methods to 3D point cloud models. Zheng et al. [58] used point cloud saliency maps for applying a point dropping attack. Sun et al. [46] applied PGD attack to 3D point cloud models. Tsai et al. [59] proposed K-Nearest-Neighbor (kNN) attack, which adds a loss term to C&W attack method using kNN distance for limiting distances between adjacent points.

Adversarial Defences. Success of adversarial attacks on deep learning models raise the importance of developing defence methods due to many safety-critical applications [11,60]. Adversarial training [14,16] is one of the possible countermeasures against adversarial attacks. This method uses both clean and adversarial examples to train a more robust model. Liu et al. [57] extended adversarial training procedure to 3D point cloud models. Zhang et al. [61] proposed PointCutMix, which is a data augmentation method to improve robustness during training process.

Beyond adversarial training and data augmentation, there exists adversarial purification [49,62] as another defence method, which aims to increase model robustness by readjusting adversarial inputs to align more closely with the true distribution, employing a range of transformation methods. Nie et al. [49] proposed DiffPure,

which uses Denoising Diffusion Probabilistic Models (DDPM) [63] for adversarial purification as a preprocessing step on 2D inputs for classification. For 3D point clouds, Zhou et al. [51] proposed Denoising and UPsampler Network (DUP-Net) that uses statistical outlier removal (SOR) [64] and a point upsampler network [65] for removing outliers and increasing surface smoothness on adversarial input point clouds. Furthermore, Wu et al. [52] proposed IF-Defense, which also uses SOR as first step and an implicit function network [66] for recovering surface of point clouds in the next steps to defend against 3D adversarial attacks. PointDP [67] extended 2D adversarial purification diffusion models to 3D, where a conditional 3D point cloud diffusion model is leveraged.

3.3 A Novel Method: Point Cloud Layerwise Diffusion

In the preliminaries section, we will first define the basics of a classifier neural network with multiple layers and diffusion processes. After that, we will explain diffusion purification and our suggested method, **PCLD**, which suggests multi-layer diffusion purification for point cloud classification.

3.3.1 Preliminaries

Classifiers. For any input x from class y, we can define a classifier $f_\theta(x)$ with N layers as:

$$z^{(0)} = x \tag{3.1}$$

$$z^{(i+1)} = f_\theta^{(i)}(z^{(i)}), 0 \leq i < N \tag{3.2}$$

$$\hat{y} = f_\theta^{(N)}(z^{(N)}) \tag{3.3}$$

We first define the layer-wise latent representation $z^{(i)}$ for values $i \in \{0,..,N\}$ where the initial $z^{(0)}$ is equal to the input, as shown in the equation 3.1. In equation 3.2, latent representation for the subsequent layer is calculated using the non-linear intermediate

layer functions, $f_\theta^{(i)}$, which are parameterized over θ. After that, the prediction $\hat{y}$ is calculated by the classification head $f^{(N)}$ in equation 3.3. We can define a loss function (e.g. cross-entropy loss) to calculate the error between true label y and the prediction $\hat{y}$ as,

$$L(y,\hat{y}) = -\sum_k y_k \log(\hat{y}_k). \tag{3.4}$$

Adversarial attacks. An adversarial example is a sample that is within the ε neighbourhood $\mathscr{V}_\varepsilon(x)$ of the input, which maximizes the loss function [14],

$$x_{adv} = \underset{x \in \mathscr{V}_\varepsilon(z^{(0)})}{\operatorname{argmax}} L(y, f_\theta(x)). \tag{3.5}$$

The imperceptible small perturbation between x_{adv} and x in the input space, increases the output loss and causes incorrect predictions in the classifier network.

Assumption 3.1. *An adversarial sample x_{adv} that is within the ε distance to x in the input layer $z^{(0)}$, gets farther from its unattacked location in every layer $z^{(i)}$ to maximize classification loss.*

Diffusion Probabilistic Models. Consistent with the work by Luo and Hu [68], the diffusion process begins with a point cloud x_0 sampled from an unknown data distribution $q(x)$. The forward phase of the diffusion model gradually introduces Gaussian noise to this initial point cloud. This progression is mathematically represented as:

$$q(x_{1:N}|x_0) := \prod_{n=1}^{N} q(x_n|x_{n-1})_{\sim \mathcal{N}(x_n;(1-\beta_n)x_{n-1},\beta_n I)} \tag{3.6}$$

where x_n represents the n^{th} step result generated within the process and β_n is the noise scheduler which increases the noise variance progressively.

The reverse process aims to reconstruct the point cloud by progressively removing the added Gaussian noise using the equation,

$$p_\theta(x_{0:N}|z_x) := p(x_N) \prod_{n=1}^{N} p_\theta(x_{n-1}|x_n, z_x)_{\sim \mathcal{N}(x_{n-1}|\mu_\theta(x_n,n,z_x),\beta_n I)} \tag{3.7}$$

where z_x is an encoded embedding for the diffusion guidance and μ_θ is the mean value of the underlying distribution approximated by the diffusion network sampled as,

$$\alpha_n := \prod_{i=1}^{n}(1-\beta_i) \tag{3.8}$$

$$\mu_\theta(x_n, n, z_x) = \frac{1}{\sqrt{1-\beta_n}} \left(x_n - \beta_n \sqrt{1-\alpha_n} \varepsilon_\theta(x_n, n, z_x) \right) \tag{3.9}$$

following the original DDPM setup [63] and the reparametrization trick. Consequently, considering both forward and backward processes, we end up with the following assumption.

Assumption 3.2. *A diffusion probabilistic model learns underlying data distrubition and it generates new samples by pushing noisy samples back into true distrubition in the reverse process with high precision.*

3.3.2 Point cloud layerwise diffusion (PCLD)

Diffusion Purification. As suggested in DiffPure for 2D [49] and PointDP for 3D [69], it is possible to purify adversarial noise from point clouds with forward and backward diffusion steps. Starting from $x_0 = x_{adv}$, the forward diffusion process for $t = (0, .., t^*)$ and $t^* \in (0, 1)$ can be computed as:

$$x(\frac{n}{N}) := x_n, \quad \beta(\frac{n}{N}) := \beta_n, \quad \alpha(\frac{n}{N}) := \alpha_n \tag{3.10}$$

$$x(t^*) = \sqrt{\alpha(t^*)}x_{adv} + \sqrt{1-\alpha(t^*)}\varepsilon, \quad \varepsilon \sim \mathcal{N}(0,I) \tag{3.11}$$

The purification result $\hat{x}(0)$ could be obtained using stochastic differential equation (SDE) solver $sdeint(\cdot)$ [49,69] defined by the following equations:

$$\hat{x}(0) = \text{sdeint}(x(t^*), f_{\text{rev}}, g_{\text{rev}}, w, t^*, 0) \tag{3.12}$$

$$f_{\text{rev}}(x,t,z_x) = -\frac{1}{2}\beta(t)[x + 2s_\theta(x,t,z_x)], \quad g_{\text{rev}}(t) = \sqrt{\beta(t)} \tag{3.13}$$

$$s_\theta(x,t,z_x) = -\frac{1}{\sqrt{1-\alpha(t)}}\varepsilon_\theta(x(t), tN, z_x) \tag{3.14}$$

Investigating the difference between $x(0)$ and $\hat{x}(0)$, both PointDP and DiffPure pointed out that the adversarial noise in the given examples are eliminated during the diffusion process.

Assumption 3.3. *Using a pretrained diffusion probabilistic model, adversarial noises can be purified with truncated number ($t^* < 1$) of forward and backward steps.*

PCLD: Point Cloud Layerwise Diffusion. Inspired by PointDP, we expand the purification process to latent space where we replace the initial point cloud with the high dimensional layer data $x_0 = z^{(i)}$.

Based on our previous assumptions, we claim that, it is possible to train multiple layer-wise diffusion models to learn underlying distribution in every layer and hierarchically/recursively purify each layer to remove adversarial perturbations. We call our method **Point Cloud Layerwise Diffussion** (**PCLD**) and suggested the following training and inference methods.

In the training phase, we train a diffusion probabilistic model for the layer features $z^{(i)}$ of a pretrained classifier neural network $f_\theta(\cdot)$. We minimize the Fisher Divergence using the following loss:

$$\mathscr{L} = E_{z^{(i)},t,\varepsilon}[\|\varepsilon - \varepsilon_\theta(\sqrt{\alpha(t)}z^{(i)} + \sqrt{1-\alpha(t)}\varepsilon, t, e(z^{(i)}))\|^2] \tag{3.15}$$

In inference however, we apply the truncated forward and backward steps following the equations 3.16 & 3.17:

$$z^{(i)}(t^*) = \sqrt{\alpha(t^*)}z^{(i)}_{adv} + \sqrt{1-\alpha(t^*)}\varepsilon, \quad \varepsilon \sim \mathscr{N}(0,I) \tag{3.16}$$

$$\hat{z}^{(i)}(0) = \text{sdeint}(z^{(i)}(t^*), f_{\text{rev}}, g_{\text{rev}}, w, t^*, 0) \tag{3.17}$$

We have demonstrated the layerwise application processes of PCLD and the internal inference process in the Figure 3.2 part (a) & (b) respectively. Our suggested method can be applied to any trained classifier $f_\theta(\cdot)$ in a plug-and-play manner.

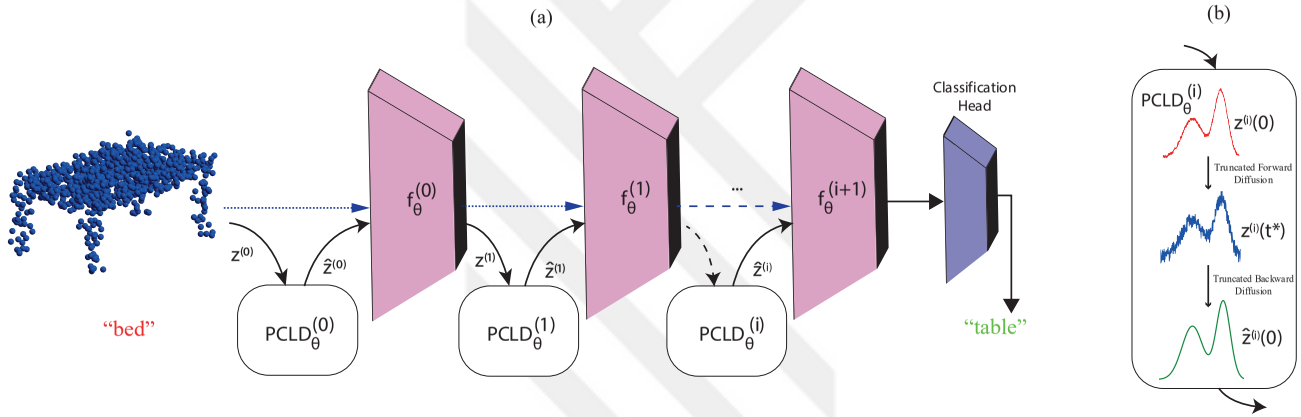

Figure 3.2 : Overview of the **Point Cloud Layerwise Diffusion** method. The application of PCLD blocks in intermediate layers is illustrated in part (a), while the truncated diffusion process and details of the PCLD block are given in part (b).

3.4 Experimental Results

We have evaluated our proposed method with 5 different models and 6 different attacks on ModelNet40 [70] dataset. Our setup consists of various pretrained classification neural networks: DGCNN [36], PCT [54], PointNet [1], PointNet++ [35] and CurveNet [56]. PointNet and PointNet++ are the pioneering point cloud classification neural networks while DGCNN is a following work based on dynamic graphs. PCT and CurveNet on the other hand are more recent architectures with higher classification performance, based on attention and random walks.

For every classifier, we tested out the following white-box adversarial attacks: Add [18], C&W [18,45], Drop [58], kNN [59], PGD ($\|L_\infty\|$) and PGD-L2 ($\|L_2\|$) [16,46] attacks.

We compared our suggested method PCLD against multiple state-of-the-art adversarial purification methods. The compared methods are as follows: SRS [51], SOR [64], DUPNet [51], IF-Defense [52] and PointDP [69]. We selected the number of total steps in diffusion model as $N = 200$. For finding the optimal truncated number of diffusion steps t^*, a grid search algorithm is used for both PointDP and our method PCLD.

We have showed our test outcomes in Table 3.1. The experimental findings indicate that our approach excels in 5 out of 6 attacks in the CurveNet Model and 4 out of 6 attacks in DGCNN, ranking second in the remaining cases. Regarding PCT and PointNet++, our model ranks within the top 2 defence methods in 5 out of 6 attacks. In the case of PointNet, our results are comparable to other defence methods. We propose that the reduced performance observed in PointNet and PointNet++ could be related to the translation network in the initial layer. Additionally, for the other three models, we rank second best on clean point clouds following the SOR. This is naturally expected,

Table 3.1 : Experiment results reported in terms of Accuracy (%) percent. Each row correspond to a Model-attack pair while defense methods are given in the columns. The Clean attack correspondences to no attack being applied while defense is applied. Same applies for 'None' defend method. Best and second scores are shown by * and † respectively.

		Defense						
Model	Attacks	None	SRS	SOR [64]	DUPNet [51]	IF-Defense [52]	PointDP [67]	PCLD (ours)
DGCNN [36]	Clean	92.87	84.81	*91.65	67.18	86.06	88.19 ±2.26	†89.10 ±1.66
	Add	63.29	77.43	77.39	58.67	†84.56	83.14	*84.60
	CW	0.28	61.83	66.65	36.75	82.86	†85.78	*86.47
	Drop	72.20	43.07	*75.61	46.84	73.78	74.59	†75.04
	KNN	0.61	57.86	61.18	35.01	83.67	†85.29	*85.70
	PGD	0.77	32.66	20.62	25.16	71.39	†75.08	*76.05
	PGDL2	2.19	49.76	46.03	48.78	*81.93	77.39	†79.38
PCT [9]	Clean	92.95	†91.41	*92.42	85.53	89.47	89.62 ±3.34	89.59 ±1.15
	Add	63.09	78.36	80.15	72.41	*86.14	82.66	†85.94
	CW	0.00	80.79	*89.71	77.35	†88.86	84.56	87.52
	Drop	73.95	73.91	†76.05	60.45	75.16	75.16	*77.88
	KNN	0.65	76.42	65.76	65.80	†86.02	83.27	*86.10
	PGD	3.16	45.42	32.70	36.51	*72.12	55.55	†69.33
	PGDL2	9.81	44.85	58.63	64.67	*81.00	62.03	†73.70
PointNet [1]	Clean	89.66	†88.90	*89.34	88.37	86.35	88.16 ±0.48	84.78 ±3.11
	Add	51.34	65.52	77.59	79.09	*84.68	†82.50	82.37
	CW	0.00	75.16	86.35	*86.51	†86.47	85.25	84.44
	Drop	45.30	52.27	51.82	55.27	62.84	*71.03	†70.99
	KNN	0.32	65.52	70.95	79.46	†84.20	*85.41	84.00
	PGD	4.09	24.07	47.20	60.29	*79.38	70.42	†72.08
	PGDL2	0.00	6.20	48.30	†63.41	*78.32	33.67	51.94
PointNet++ [35]	Clean	91.05	†90.48	*90.96	87.40	87.76	86.92 ±2.53	86.99 ±1.79
	Add	71.60	78.69	78.77	77.03	*85.17	82.25	†82.98
	CW	0.00	76.50	†85.01	82.50	*87.93	80.83	81.48
	Drop	†81.60	80.31	80.63	75.00	79.25	81.60	*83.14
	KNN	0.61	74.23	62.40	73.18	*86.35	83.51	†83.83
	PGD	0.08	15.03	7.21	28.89	*72.73	67.83	†67.99
	PGDL2	1.22	34.81	41.37	64.83	*83.71	72.61	†73.14
CurveNet [56]	Clean	93.84	88.53	*91.13	89.22	88.13	89.59 ±2.42	†90.93 ±2.10
	Add	66.21	74.84	82.86	82.33	85.49	†85.98	*87.16
	CW	0.00	72.33	87.76	86.14	†88.05	85.45	*88.45
	Drop	80.15	55.19	79.09	76.13	76.66	†80.15	*82.29
	KNN	0.69	75.36	71.68	82.50	†86.47	85.94	*88.57
	PGD	4.09	30.83	21.96	54.70	66.37	†77.67	*78.73
	PGDL2	9.40	40.15	53.00	71.76	*80.43	74.88	†76.86

Table 3.2 : List of truncated diffusion steps for PointDP and PCLD methods. Each row correspond to a Model-attack pair while corresponding layers are given in the columns.

		PointDP [69]	PCLD (ours)				
		Input	Input	Layer 1	Layer 2	Layer 3	Layer 4
DGCNN	Add	5	5	15	20	5	25
	CW	10	5	0	30	10	40
	Drop	5	5	0	30	10	40
	kNN	10	5	5	35	10	50
	PGD	15	10	5	45	0	70
	PGDL2	25	15	5	50	0	25
PCT	Add	5	5	30	35	15	85
	CW	5	5	10	60	35	10
	Drop	5	5	15	90	15	55
	kNN	5	5	25	50	0	100
	PGD	10	5	35	85	70	30
	PGDL2	10	5	35	100	60	0
PointNet	Add	10	15	25	10	0	-
	CW	30	15	15	15	5	-
	Drop	80	70	30	15	0	-
	kNN	30	15	15	15	5	-
	PGD	30	25	25	0	0	-
	PGDL2	30	15	70	35	0	-
PointNet++	Add	5	5	15	20	0	-
	CW	15	10	35	30	0	-
	Drop	0	0	45	80	0	-
	kNN	15	10	15	10	0	-
	PGD	30	20	30	25	0	-
	PGDL2	45	30	5	0	0	-
CurveNet	Add	10	5	90	20	50	20
	CW	20	5	100	55	10	0
	Drop	0	0	60	30	25	100
	kNN	20	5	95	55	40	45
	PGD	95	100	85	25	15	0
	PGDL2	30	20	75	40	0	55

as the SOR method is specifically designed to eliminate outliers, which have minimal impact on clean samples.

In Table 3.2, we present the selected truncated number of diffusion steps t^* for both PointDP and our method PCLD. Among the models, DGCNN, PCT, and CurveNet feature four purified layers, while PointNet and PointNet++ have three layers based on their architectures. Notably, for the PCT model, the number of diffusion steps taken at Input-Layer 1 is relatively low compared to Layers 2-4. When considering that our

PCLD model consistently outperforms PointDP by at least 3% accuracy in every attack scenario for PCT, the significance of purification in deeper layers becomes evident.

Overall, we surpass our pioneering diffusion based purification method PointDP and recieve comparable results against other defence methods.

4. CONCLUSION

4.1 Summary of Findings

4.1.1 Adversarial attacks

In Chapter 2, we proposed a novel 3D adversarial attack, the ε-mesh Attack, designed to preserve the structural integrity of 3D faces while effectively misleading facial expression recognition models. By confining adversarial perturbations to the surface of 3D meshes, our method reduces the 3D optimization domain into 2D triangular planes. This is achieved through two projection methods: Central projection, which projects perturbations toward the triangle's center of mass, and Perpendicular projection, which projects perturbations to the closest point on the triangle. Using PointNet and DGCNN models trained on the CoMA, Bosphorus, and FaceWarehouse datasets, we demonstrated that these methods maintain surface structure and point density while approaching the performance of traditional 3D ε-ball attacks. These results highlight the utility of our approach in applications requiring surface integrity preservation, such as facial expression recognition and safety-critical domains.

4.1.2 Adversarial defences

In Chapter 3, we introduced Point Cloud Layerwise Diffusion (PCLD), a novel defense mechanism aimed at enhancing the robustness of 3D point cloud classification models against adversarial attacks. Extending diffusion-based purification methods like PointDP, PCLD employs a layerwise purification strategy, training diffusion

probabilistic models for each network layer. This enables hierarchical purification of adversarial perturbations across multiple levels of the network.

Comparative analysis with state-of-the-art defense mechanisms revealed that PCLD consistently matches or outperforms existing methods, especially in mitigating attacks on deeper network layers. This establishes PCLD as a promising method for robust 3D point cloud classification, with adaptability to diverse architectures and attack scenarios.

4.2 Discussion

Significance of the ε-Mesh Attack. The ε-mesh Attack introduces a pioneering method for adversarial attacks on 3D facial expression recognition models by ensuring perturbations conform to mesh surfaces. Unlike existing approaches, this method emphasizes maintaining the structural integrity of 3D meshes, a critical aspect for realistic and undetectable adversarial manipulations. This feature is particularly significant for applications in authentication and human-robot interaction, where any visible distortion could compromise functionality.

Potential countermeasures include adversarial training incorporating our methods to improve model robustness, and input preprocessing-based defences such as those proposed in [64] and [51], aimed at sanitizing adversarial examples before classification.

Limitations of the ε-Mesh Attack. The ε-mesh Attack has certain limitations. One key challenge is convergence, as the additional constraint of mesh boundaries increases the steps required to reach convergence, which limits its real-time applicability. Furthermore, the performance of the attack relies heavily on the availability of accurate mesh data. In scenarios where true meshes are unavailable, surface estimation methods

may be employed; however, such estimations can introduce inaccuracies, as observed in experiments with the Bosphorus dataset.

Impact of PCLD. Layerwise approach of PCLD demonstrates significant potential for defending against adversarial attacks, particularly in feature-level purification. While adaptable to diverse domains such as autonomous driving and robotics, challenges such as computational overhead and adaptation to new attack strategies remain and warrant further investigation.

4.3 Future Directions

Future research can focus on enhancing adversarial training by integrating the proposed 3D adversarial sampling methods into adversarial training frameworks to enable classifiers to better withstand attacks while preserving structural integrity. Another direction involves extending current methods to the time domain, which could enable a 4D setup and align with datasets like CoMA [26], facilitating robust defences for dynamic applications involving temporal sequence analysis, such as robotics and video-based authentication. In the defence aspect, optimizing the scalability of PCLD to address computational demands is critical for its application to larger datasets and complex models, ensuring practicality in industrial applications. As adversarial attack strategies continue to evolve, continuous adaptation of PCLD to counter novel and sophisticated threats will be essential to maintain its relevance and effectiveness in real-world scenarios. By pursuing these directions, advancements in both adversarial attack and defence mechanisms for 3D point clouds will support the safe and reliable integration of these technologies into critical applications.

REFERENCES

[1] **Qi, C.R.**, **Su, H.**, **Mo, K. and Guibas, L.J.** (2017). PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, *IEEE/CVF CVPR*.

[2] **Klokov, R. and Lempitsky, V.** (2017). Escape from cells: Deep kd-networks for the recognition of 3d point cloud models, *ICCV*, pp.863–872.

[3] **Zhou, J.**, **Xiong, Y.**, **Chiu, C.**, **Liu, F. and Gong, X.** (2023). Fat: Field-Aware Transformer for 3D Point Cloud Semantic Segmentation, *IEEE ICIP*, IEEE, pp.660–664.

[4] **Li, A.**, **Lv, C.**, **Fang, Y. and Zuo, Y.** (2023). Laptran: Transformer Embedding Graph Laplacian for Point Cloud Part Segmentation, *IEEE ICIP*, IEEE, pp.3070–3074.

[5] **Deng, H.**, **Birdal, T. and Ilic, S.** (2018). Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors, *ECCV*, pp.602–618.

[6] **Mei, G.**, **Huang, X.**, **Zhang, J. and Wu, Q.** (2022). Partial point cloud registration via soft segmentation, *IEEE ICIP*, IEEE, pp.681–685.

[7] **Campagnolo, D.**, **Camuffo, E.**, **Michieli, U.**, **Borin, P.**, **Milani, S. and Giordano, A.** (2023). Fully Automated Scan-to-BIM Via Point Cloud Instance Segmentation, *IEEE ICIP*, IEEE, pp.291–295.

[8] **Chen, J.**, **Kira, Z. and Cho, Y.K.** (2019). Deep learning approach to point cloud scene understanding for automated scan to 3D reconstruction, *Journal of Computing in Civil Engineering*, *33*(4), 04019027.

[9] **Chen, S.**, **Niu, S.**, **Lan, T. and Liu, B.** (2019). PCT: Large-scale 3D point cloud representations via graph inception networks with applications to autonomous driving, *IEEE ICIP*, IEEE, pp.4395–4399.

[10] **Aygun, M.**, **Osep, A.**, **Weber, M.**, **Maximov, M.**, **Stachniss, C.**, **Behley, J. and Leal-Taixé, L.** (2021). 4d panoptic lidar segmentation, *IEEE/CVF CVPR*, pp.5527–5537.

[11] **Cao, Y.**, **Xiao, C.**, **Cyr, B.**, **Zhou, Y.**, **Park, W.**, **Rampazzi, S.**, **Chen, Q.A.**, **Fu, K. and Mao, Z.M.** (2019). Adversarial sensor attack on lidar-based perception in autonomous driving, *ACM SIGSAC conference on computer and communications security*, pp.2267–2281.

[12] **Lin, Y.**, **Zhao, C.**, **Li, D.**, **Xu, J. and Zhang, B.** (2020). Exploring Model Transfer Potential for Airborne LiDAR Point Cloud Classification, *Pattern Recognition and Artificial Intelligence: Third Mediterranean Conference, MedPRAI 2019, Istanbul, Turkey, December 22–23, 2019, Proceedings 3*, Springer, pp.39–51.

[13] **Wang, X.**, **Chen, X.**, **Zhao, Z.**, **Zhang, Y.**, **Zheng, D. and Han, J.** (2023). High-precision point cloud registration system of multi-view industrial self-similar workpiece based on super-point space guidance, *Journal of Intelligent Manufacturing*, 1–15.

[14] **Goodfellow, I.J.**, **Shlens, J. and Szegedy, C.** (2014). Explaining and harnessing adversarial examples, *arXiv preprint arXiv:1412.6572*.

[15] **Qiu, S.**, **Liu, Q.**, **Zhou, S. and Wu, C.** (2019). Review of artificial intelligence adversarial attack and defense technologies, *Applied Sciences*, *9*(5), 909.

[16] **Madry, A.**, **Makelov, A.**, **Schmidt, L.**, **Tsipras, D. and Vladu, A.** (2017). Towards deep learning models resistant to adversarial attacks, *arXiv preprint arXiv:1706.06083*.

[17] **Kurakin, A.**, **Goodfellow, I.J. and Bengio, S.**, (2018). Adversarial examples in the physical world, Artificial intelligence safety and security, Chapman and Hall/CRC, pp.99–112.

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

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

[20] **Huang, Q.**, **Dong, X.**, **Chen, D.**, **Zhou, H.**, **Zhang, W. and Yu, N.** (2022). Shape-invariant 3d adversarial point clouds, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.15335–15344.

[21] **Zhang, J.**, **Chen, L.**, **Liu, B.**, **Ouyang, B.**, **Xie, Q.**, **Zhu, J.**, **Li, W. and Meng, Y.** (2023). 3d adversarial attacks beyond point cloud, *Information Sciences*, *633*, 491–503.

[22] **Colombo, A.**, **Cusano, C. and Schettini, R.** (2011). UMB-DB: A database of partially occluded 3D faces, *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, IEEE, pp.2113–2119.

[23] **Yang, H.**, **Zhu, H.**, **Wang, Y.**, **Huang, M.**, **Shen, Q.**, **Yang, R. and Cao, X.** (2020). Facescape: a large-scale high quality 3d face dataset and detailed riggable 3d face prediction, *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pp.601–610.

[24] **Cao, C.**, **Weng, Y.**, **Zhou, S.**, **Tong, Y. and Zhou, K.** (2013). Facewarehouse: A 3d facial expression database for visual computing, *IEEE Transactions on Visualization and Computer Graphics*, *20*(3), 413–425.

[25] **Bolkart, T.**, **Li, T. and Black, M.J.** (2023). Instant Multi-View Head Capture through Learnable Registration, *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.768–779.

[26] **Ranjan, A.**, **Bolkart, T.**, **Sanyal, S. and Black, M.J.** (2018). Generating 3D faces using Convolutional Mesh Autoencoders, *European Conference on Computer Vision (ECCV)*, pp.725–741, `http://coma.is.tue.mpg.de/`.

[27] **Ekman, P.** (1999). Basic Emotions, chapter 3, John Wiley Sons, Ltd, pp.45–60, `https://onlinelibrary.wiley.com/doi/abs/10.1002/0470013494.ch3`, `https://onlinelibrary.wiley.com/doi/pdf/10.1002/0470013494.ch3`.

[28] **Dhall, A.**, **Ramana Murthy, O.**, **Goecke, R.**, **Joshi, J. and Gedeon, T.** (2015). Video and image based emotion recognition challenges in the wild: Emotiw 2015, *Proceedings of the 2015 ACM on international conference on multimodal interaction*, pp.423–426.

[29] **Lucey, P.**, **Cohn, J.F.**, **Kanade, T.**, **Saragih, J.**, **Ambadar, Z. and Matthews, I.** (2010). The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression, *2010 ieee computer society conference on computer vision and pattern recognition-workshops*, IEEE, pp.94–101.

[30] **Richter, M.**, **Gehrig, T. and Ekenel, H.K.** (2012). Facial expression classification on web images, *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, IEEE, pp.3517–3520.

[31] **Afshar, S. and Ali Salah, A.** (2016). Facial expression recognition in the wild using improved dense trajectories and fisher vector encoding, *Proceedings*

of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp.66–74.

[32] **Liu, X.**, **Vijaya Kumar, B.**, **You, J. and Jia, P.** (2017). Adaptive deep metric learning for identity-aware facial expression recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp.20–29.

[33] **Yang, H.**, **Ciftci, U. and Yin, L.** (2018). Facial expression recognition by de-expression residue learning, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp.2168–2177.

[34] **Qi, C.R.**, **Su, H.**, **Mo, K. and Guibas, L.J.** (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.652–660.

[35] **Qi, C.R.**, **Yi, L.**, **Su, H. and Guibas, L.J.** (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space, *Advances in neural information processing systems*, pp.5099–5108.

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

[37] **Liu, Y.J.**, **Wang, B.**, **Gao, L.**, **Zhao, J.**, **Yi, R.**, **Yu, M.**, **Pan, Z. and Gu, X.** (2023). 4D facial analysis: A survey of datasets, algorithms and applications, *Computers & Graphics*, *115*, 423–445.

[38] **Duh, D.J.**, **Huang, J.C.**, **Chen, S.Y.**, **Su, S.**, **Zhang, H. and Li, S.** (2016). Facial expression recognition based on spatio-temporal interest points for depth sequences, *The Imaging Science Journal*, *64*(7), 396–407.

[39] **Behzad, M.**, **Li, X. and Zhao, G.** (2021). Disentangling 3D/4D facial affect recognition with faster multi-view transformer, *IEEE Signal Processing Letters*, *28*, 1913–1917.

[40] **Yin, L.**, **Wei, X.**, **Sun, Y.**, **Wang, J. and Rosato, M.J.** (2006). A 3D facial expression database for facial behavior research, *7th international conference on automatic face and gesture recognition (FGR06)*, IEEE, pp.211–216.

[41] **Yin, L.**, **Chen, X.**, **Sun, Y.**, **Worm, T. and Reale, M.** (2008). A high-resolution 3D dynamic facial expression database, *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*, pp.1–6.

[42] **Zhang, X.**, **Yin, L.**, **Cohn, J.F.**, **Canavan, S.**, **Reale, M.**, **Horowitz, A.**, **Liu, P. and Girard, J.M.** (2014). Bp4d-spontaneous: a high-resolution spontaneous 3d dynamic facial expression database, *Image and Vision Computing*, *32*(10), 692–706.

[43] **Li, S. and Deng, W.** (2020). Deep facial expression recognition: A survey, *IEEE transactions on affective computing*, *13*(3), 1195–1215.

[44] **Szegedy, C.**, **Zaremba, W.**, **Sutskever, I.**, **Bruna, J.**, **Erhan, D.**, **Goodfellow, I. and Fergus, R.** (2013). Intriguing properties of neural networks, *arXiv preprint arXiv:1312.6199*.

[45] **Carlini, N. and Wagner, D.** (2017). Towards evaluating the robustness of neural networks, *2017 ieee symposium on security and privacy (sp)*, Ieee, pp.39–57.

[46] **Sun, J.**, **Cao, Y.**, **Choy, C.B.**, **Yu, Z.**, **Anandkumar, A.**, **Mao, Z.M. and Xiao, C.** (2021). Adversarially robust 3d point cloud recognition using self-supervisions, *NeurIPS*, *34*, 15498–15512.

[47] **Savran, A.**, **Alyüz, N.**, **Dibeklioğlu, H.**, **Çeliktutan, O.**, **Gökberk, B.**, **Sankur, B. and Akarun, L.** (2008). Bosphorus database for 3D face analysis, *Biometrics and Identity Management: First European Workshop, BIOID 2008, Roskilde, Denmark, May 7-9, 2008. Revised Selected Papers 1*, Springer, pp.47–56.

[48] **Kazhdan, M.**, **Bolitho, M. and Hoppe, H.** (2006). Poisson surface reconstruction, *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, p. 0.

[49] **Nie, W.**, **Guo, B.**, **Huang, Y.**, **Xiao, C.**, **Vahdat, A. and Anandkumar, A.** (2022). Diffusion models for adversarial purification, *arXiv:2205.07460*.

[50] **Wicker, M. and Kwiatkowska, M.** (2019). Robustness of 3d deep learning in an adversarial setting, *IEEE/CVF CVPR*, pp.11767–11775.

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

[52] **Wu, Z.**, **Duan, Y.**, **Wang, H.**, **Fan, Q. and Guibas, L.J.** (2020). If-defense: 3d adversarial point cloud defense via implicit function based restoration, *arXiv preprint arXiv:2010.05272*.

[53] **Li, G.**, **Xu, G.**, **Qiu, H.**, **He, R.**, **Li, J. and Zhang, T.** (2022). Improving adversarial robustness of 3d point cloud classification models, *ECCV*, Springer, pp.672–689.

[54] **Guo, M.H.**, **Cai, J.X.**, **Liu, Z.N.**, **Mu, T.J.**, **Martin, R.R. and Hu, S.M.** (2021). Pct: Point cloud transformer, *Computational Visual Media*, *7*, 187–199.

[55] **Ma, X.**, **Qin, C.**, **You, H.**, **Ran, H. and Fu, Y.** (2022). Rethinking network design and local geometry in point cloud: A simple residual MLP framework, *arXiv:2202.07123*.

[56] **Muzahid, A.A.M.**, **Wan, W.**, **Sohel, F.**, **Wu, L. and Hou, L.** (2021). CurveNet: Curvature-Based Multitask Learning Deep Networks for 3D Object Recognition, *IEEE/CAA Journal of Automatica Sinica*, *8*(6), 1177–1187.

[57] **Liu, D.**, **Yu, R. and Su, H.** (2019). Extending adversarial attacks and defenses to deep 3d point cloud classifiers, *IEEE ICIP*, IEEE, pp.2279–2283.

[58] **Zheng, T.**, **Chen, C.**, **Yuan, J.**, **Li, B. and Ren, K.** (2019). Pointcloud saliency maps, *IEEE/CVF CVPR*, pp.1598–1606.

[59] **Tsai, T.**, **Yang, K.**, **Ho, T.Y. and Jin, Y.** (2020). Robust adversarial objects against deep learning models, *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp.954–962.

[60] **Tu, J.**, **Ren, M.**, **Manivasagam, S.**, **Liang, M.**, **Yang, B.**, **Du, R.**, **Cheng, F. and Urtasun, R.** (2020). Physically Realizable Adversarial Examples for LiDAR Object Detection, *IEEE/CVF CVPR*.

[61] **Zhang, J.**, **Chen, L.**, **Ouyang, B.**, **Liu, B.**, **Zhu, J.**, **Chen, Y.**, **Meng, Y. and Wu, D.** (2022). Pointcutmix: Regularization strategy for point cloud classification, *Neurocomputing*, *505*, 58–67.

[62] **Shi, C.**, **Holtz, C. and Mishne, G.** (2021). Online adversarial purification based on self-supervision, *arXiv:2101.09387*.

[63] **Ho, J.**, **Jain, A. and Abbeel, P.** (2020). Denoising diffusion probabilistic models, *NeurIPS*, *33*, 6840–6851.

[64] **Rusu, R.B.**, **Marton, Z.C.**, **Blodow, N.**, **Dolha, M. and Beetz, M.** (2008). Towards 3D point cloud based object maps for household environments, *Robotics and Autonomous Systems*, *56*(11), 927–941.

[65] **Yu, L.**, **Li, X.**, **Fu, C.W.**, **Cohen-Or, D. and Heng, P.A.** (2018). Pu-net: Point cloud upsampling network, *IEEE/CVF CVPR*, pp.2790–2799.

[66] **Mescheder, L.**, **Oechsle, M.**, **Niemeyer, M.**, **Nowozin, S. and Geiger, A.** (2019). Occupancy networks: Learning 3d reconstruction in function space, *IEEE/CVF CVPR*, pp.4460–4470.

[67] **Sun, J.**, **Wang, J.**, **Nie, W.**, **Yu, Z.**, **Mao, Z. and Xiao, C.** (2023). A critical revisit of adversarial robustness in 3D point cloud recognition with diffusion-driven purification, *ICML*, PMLR, pp.33100–33114.

[68] **Luo, S. and Hu, W.** (2021). Diffusion probabilistic models for 3d point cloud generation, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp.2837–2845.

[69] **Sun, J.**, **Wang, J.**, **Nie, W.**, **Yu, Z.**, **Mao, Z. and Xiao, C.** (2023). A critical revisit of adversarial robustness in 3D point cloud recognition with diffusion-driven purification, *ICML*, PMLR, pp.33100–33114.

[70] **Wu, Z.**, **Song, S.**, **Khosla, A.**, **Yu, F.**, **Zhang, L.**, **Tang, X. and Xiao, J.** (2015). 3d shapenets: A deep representation for volumetric shapes, *IEEE/CVF CVPR*, pp.1912–1920.

CURRICULUM VITAE

Name SURNAME: Batuhan CENGİZ

EDUCATION:

- **B.Sc.:** 2021, Istanbul Technical University, Faculty of Electrical and Electronics Engineering, Department of Electronics and Communication Engineering
- **M.Sc.:** 2025, Istanbul Technical University, Faculty of Computer and Informatics, Department of Computer Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2021 RF Intern, P.I. Works
- 2022-*ongoing* Research and Teaching Assistant at Department of AI and Data Engineering, Istanbul Technical University

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Cengiz, B.**, Gulsen, M., Sahin, Y. H., Unal, G. (2024). ε-Mesh Attack: A Surface-Based Adversarial Point Cloud Attack for Facial Expression Recognition. *2024 IEEE 18th International Conference on Automatic Face and Gesture Recognition (FG)*, 1-9.
- Gulsen, M., **Cengiz, B.**, Sahin, Y. H., Unal, G. (2024). PCLD: Point Cloud Layerwise Diffusion for Adversarial Purification. *2024 6th Mediterranean Conference on Pattern Recognition and Artificial Intelligence (MedPRAI)*, 1-12.

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- **Cengiz, B.**, Karagoz, H. F., Kumbasar, T. (2024). Conformalized High-Density Quantile Regression via Dynamic Prototypes-based Probability Density Estimation. *arXiv preprint* arXiv:2411.01266.
- Gunduzalp, D., **Cengiz, B.**, Unal, M. O., Yildirim, I. (2021). 3d u-netr: Low dose computed tomography reconstruction via deep learning and 3 dimensional convolutions. *arXiv preprint* arXiv:2105.14130.