Synthetic Training Data Generation for Point Cloud Defects

Master's Thesis

for the degree of

Master of Science (M.Sc.)

Data Science

at the Faculty of Sciences of
Friedrich-Alexander-Universität Erlangen-Nürnberg

submitted on **22.01.2024**

by **Nishitha Yedalapalli**

First Supervisor: Prof. Dr. Enrique Zuazua
Second Supervisor: Dr. Yongcun Song

Friedrich-Alexander-Universität
Naturwissenschaftliche Fakultät

Contents

1 Abstract

In the field of aviation maintenance, the detection of submillimeter cracks within combustion chambers presents a significant challenge, given the intricacies of sensor data containing both fine surface features and roughness. Conventional data processing techniques struggle to differentiate between benign abnormalities and genuine cracks in this complex scenario. However, the aviation industry, specifically in aircraft maintenance, grapples with a data deficit, making model training and generalization particularly daunting. Obtaining real crack data is challenging due to its expense and unavailability in real-time. To overcome this hurdle, the proposed techniques employ Generative Adversarial Networks (GANs) to generate synthetic data. This industry project, conducted at 3D-aero, aims to establish a robust pipeline dedicated to generating synthetic training samples using GANs. By training a GAN with actual crack data and utilizing its generator to produce synthetic cracks, the synthesized data is enriched with diverse crack patterns. This dataset becomes instrumental in training a deep learning network, aiming to enhance the efficiency of crack detection procedures during airplane maintenance. The synergy between GANs and deep learning not only addresses data constraints but also empowers the model to accurately identify a spectrum of crack configurations encountered in real-world situations. Both the conventional Generative Adversarial Network (GAN) and the Wasserstein Generative Adversarial Network (WGAN) methodologies were employed to generate synthetic point clouds in the implementation. The findings underscore the effectiveness of the WGAN model in producing compelling results, showcasing its advanced capability in faithfully replicating real-world point clouds. WGAN's efficient performance is attributed to its unique wasserstein distance metric, which provides a more stable and meaningful measure of the difference between the generated and real distributions. Unlike traditional GAN, WGAN address issues such as mode collapse and training instability, leading to more realistic and diverse synthetic data. This significant finding underscores the potential of generative models, particularly WGAN, in bridging the gap posed by data scarcity. In conclusion, this research contributes to the broader goal of advancing data-driven methodologies in safety-critical industries, offering a promising avenue for future applications and research endeavors in the field.

2 Introduction

In recent years, deep learning has influenced the way we analyse and interpret complex visual data. The development of deep learning techniques has accelerated advancements across varied applications, such as object detection, image segmentation, and instance semantics. Among the numerous types of data, three-dimensional (3D) point clouds have emerged as a versatile representation that can be applied to a variety of different fields, including robotics [29], autonomous navigation [41], augmented reality [21], and many more. In robotic vision, point clouds are broadly used as 3D representations of real-world scenes. In scenarios where labelled real data is limited, efficient 3D data synthesis is essential [34].

For deep learning models to succeed in analysing point cloud data, they must have qualitative and quantitative characteristics of the training data. Graph Neural Networks (GNNs) and Convolutional Neural Networks (CNNs) are deep learning models that rely on large, diverse, and accurately labelled datasets to achieve effective performance. It can be difficult and resource-intensive to collect such datasets for point clouds in practice, especially in real-world scenarios [16], due to their multidimensional nature of data, requiring specialized sensors such as LiDAR or depth cameras, which may have inherent limitations in terms of their range and resolution. For specific tasks, such as object detection, segmentation, or classification, deep learning models require substantial amounts of labelled data that accurately reflect the complexity and variations experienced in real-world point clouds. Despite these challenges, ongoing advancements in technology and collaboration between researchers and industry may contribute to overcoming these obstacles in the future.

The integration of deep learning with aircraft maintenance is a pioneering development. As it utilizes advanced neural networks to effectively handle complex aircraft data. A key benefit of this technology is that it allows for accurate identification of potential problems, enables predictive maintenance by analysing historical data for forecasting, and enhances overall safety and reliability by continuously learning and adapting. As a result, maintenance procedures can be handled more efficiently and proactively, reducing the risk of unexpected failures and minimizing aircraft downtime. By using deep learning algorithms, such as 3D CNN, to analyse 3D point cloud data within combustion chambers, meticulous crack patterns within volumetric data can be precisely detected. This is especially important when it comes to combustion chambers, where traditional methods may struggle with the complex three-dimensional structures. Deep learning enhances precision by effectively capturing spatial relationships in the point cloud, enabling accurate detection of subtle or evolving cracks.

In addition, deep learning reduces the need for manual inspection of large datasets and facilitates a quicker analysis of them with its automation capabilities. As a result, this approach contributes to improved combustion chamber reliability, safety, and maintenance efficiency. Due to their ability to capture subtle nuances and patterns, these models can detect even the tiniest cracks while minimizing false positives caused by surface roughness.

Moreover, the depth and complexity of deep learning architectures align well with the multifaceted nature of crack detection. Deep neural networks use hierarchical representation learning to abstract features at different scales, enhancing their ability to distinguish cracks from surface irregularities [43]. During point cloud data processing, the network undergoes learning to differentiate structural defects from surface imperfections, thereby resolving the persistent challenge of confusing roughness with cracks. However, deep learning models are only effective when they are trained on extensive, high-quality data [33]. Herein lies a major hurdle for the aviation industry. A lack of quantity and diversity of data from actual aircraft maintenance inspections, particularly concerning combustion chamber cracks, is one of the main challenges. It is crucial to address these constraints to successfully deploy deep learning models.

The proposed methods in this master's thesis address two main challenges. To begin with, synthetic training samples are generated to address the data deficiency issue. Generative Adversarial Networks (GANs) can be used to generate synthetic cracks that resemble the real cracks. This synthetic data solves the shortage issue by giving an abundance of data for deep learning models. Second, the synthetic data, which is created, bridges the gap between real-world data and the training set of the deep learning model. These synthetic data samples are expected to provide the model with a more thorough grasp of fractures and surface roughness. As a result, the deep learning model's capacity to distinguish between various complexities in point cloud data is projected to be greatly improved, leading to more reliable and accurate crack detection.

We will go deeper into the technical complexities of deep learning model development, training, and the assessment in following sections of this thesis, all of which are crucial to the effective application of deep learning in the domain of aviation maintenance. We will investigate the intricacies of data preparation, model architecture selection, hyper parameter tweaking, and model performance evaluation. This study is a pioneering attempt at the interface of cutting-edge deep learning technology and aviation industry demands. This thesis intends to create a novel solution for the aviation sector by solving the severe issue of data shortages and using the potential of deep learning, enhancing aircraft safety and dependability through more precise combustion chamber crack detection.

2.1 Problem Statement

In the field of aviation maintenance, white light interferometry is useful for detecting submillimeter cracks within combustion chambers. This cutting-edge technology generates high-resolution point cloud data that captures fine surface features. However, because the sensor data contains surface roughness, it is difficult for standard data processing techniques to distinguish between benign surface abnormalities and genuine cracks. To overcome this barrier, the proposed solution makes use of deep learning methodologies. The goal is to create models capable of recognizing minute and as well as essential distinctions between surface roughness and real cracks using advanced neural network architectures. Deep learning introduces a paradigm change by allowing the system to understand deep patterns and make nuanced decisions based on the intricacies contained in the data. However, the aviation industry, particularly in aircraft maintenance, faces a data deficit. The lack of data makes training and generalizing models particularly challenging. Therefore, the technique employs Generative Adversarial Networks (GANs) to produce synthetic data.

2.2 Objective

To address the issue of limited data, the proposed technique entails building a robust pipeline dedicated to creating synthetic training samples using Generative Adversarial Networks (GANs). One critical component of this pipeline is the creation of artificial cracks. The method involves training a GAN with actual crack data and then using its generator to generate synthetic cracks. The synthesized data, which has been enhanced with various crack patterns, will subsequently be used to train a deep learning network. The end goal is to increase the efficiency of crack detection procedures during airplane maintenance by combining the power of GANs and deep learning techniques to solve data shortages and improve the model's generalization capabilities. This method not only overcomes data constraints but also enables the model to accurately identify diverse crack configurations observed in real-world situations.

2.3 Structure of Thesis

The following chapters examine specific aspects of the research questions and objectives. A detailed literature review is provided in Chapters 3 and a comprehensive overview of point cloud analysis, GANs, and WGANs along with their benefits and challenges is outlined in Chapter 4 and in Chapter 5, describing the methods and techniques for generating synthetic point clouds, and analysing the evolution of data generation techniques. As the experiment progresses, Chapter 6 presents the results, and the validation of these results is carried out.

A detailed discussion of the findings, their interpretation, and implications is provided in Chapter 7. A comprehensive conclusion is presented in the next chapter and finally ending with suggestions for future research avenues in this field, highlighting what was learned during the research process. The methodical sequence of chapters ensures a thorough exploration of the subject matter, combining theoretical foundations, empirical findings, and thoughtful discussions that contribute to the existing body of knowledge in the field.

This master's thesis seeks to address the challenges of data scarcity and improve the accuracy, robustness, and generalization capabilities of deep learning models in this domain by exploring the potential of GAN-based synthetic data generation to contribute significantly to the field of point cloud analysis. With a comprehensive investigation and empirical evaluation, this research provides useful insights and tools to advance point cloud analysis.

3 Literature Review

In recent years, researchers have embarked on an intriguing exploration of the synergies between Generative Adversarial Networks (GANs) and the challenges inherent in generating realistic images from point cloud data. This literature review meticulously navigates the pivotal contributions of researchers who have shaped the intersection of GANs and point clouds. Commencing with the foundational birth of GANs, we trace the evolutionary journey that seamlessly transitions into the realm of point clouds. Our exploration spans the processing of point clouds, shedding light on innovative methodologies and key advancements. Specifically, we delve into the applications where GANs prove instrumental in synthesizing images from point clouds, unravelling the complexity of this fusion. By providing a thorough grasp of the environment, approaches, and new developments at the intersection of GANs and point cloud data, this section of the thesis seeks to summarize research efforts.

3.1 GANs in image generation

The work of Goodfellow et al. [17] laid the groundwork for GANs. Synthetic data and assessing the authenticity of it is accomplished through adversarial training, which uses a generator network and a discriminator network. It marked the beginning of a new era in generative modelling, enabling a novel method of synthesising images. He not only introduced Generative Adversarial Networks (GANs) but also revolutionized the landscape of generative modelling. The adversarial training paradigm, involving the interplay between a generator and a discriminator network, brought forth a unique approach to the creation of synthetic data. The generator network, employing deep neural architectures, is tasked with transforming random noise into data that ideally mirrors the characteristics of real-world samples. Concurrently, the discriminator network is trained to differentiate between authentic and generated data. This adversarial dynamic has proven to be a robust mechanism for producing high-quality synthetic data, subsequently opening up novel avenues in various domains, from computer vision to natural language processing. The ability of GANs to learn and replicate complex patterns in data marked the beginning of a new era in generative modelling, offering innovative solutions for image synthesis, data augmentation, and beyond. The influence of this seminal work extends far beyond the realms of academia, as GANs continue to be a driving force behind advancements in artificial intelligence and machine learning.

In addition, Radford et al. (2015) [32] further developed this paradigm with Deep Convolutional Generative Adversarial Networks (DCGAN), introducing deep convolutional structures to enhance the generation of high-resolution and realistic images. DCGANs proved to be a significant improvement over traditional GAN architectures, providing stability in training and enabling the synthesis of intricate features in images. The deep convolutional layers not only captured spatial hierarchies more effectively but also facilitated the generation of more complex and detailed visual content. As

GANs gained widespread success in various image generation tasks, their application to point clouds, which represent three-dimensional data, has presented unique challenges. Unlike regular images, point clouds lack a grid-like structure, requiring adaptations to leverage the strengths of GANs in capturing intricate patterns and structures in this non-grid data format. Researchers have explored novel approaches and modifications to extend the capabilities of GANs to point cloud generation, marking an exciting frontier in the application of adversarial networks to diverse data modalities.

Furthermore, the Wasserstein distance is presented in the "Wasserstein GAN" paper by Martin Arjovsky, Soumith Chintala [24], and Leon Bottou as a critical parameter for improving the training stability of Generative Adversarial Networks (GANs). In order to assure computational feasibility and stability, the research suggests a Lipschitz restriction on the critic network, addressing the drawbacks of conventional divergence metrics. The authors hope to address problems with conventional GANs, like mode collapse and training instability, by using Wasserstein distance. The research presents empirical findings that illustrate the effectiveness of Wasserstein GAN (WGAN) in producing superior quality samples and attaining enhanced convergence in contrast to its predecessors. By providing a more dependable framework for GAN training and demonstrating the potential of Wasserstein distance in tackling important issues in generative model optimization, this study represents a substantial contribution to the field.

3.2 GANs in point clouds

In the realm of 3D data generation, the work presented by Chun-Liang Li et al. [22] presents a significant advancement by applying Generative Adversarial Networks to the synthesis of point cloud data. Chun-Liang Li et al. address the unique challenges associated with generating realistic and coherent point cloud representations, offering a pioneering solution in the domain of 3D data synthesis. By extending the principles of GANs to the intricacies of point cloud structures, the proposed framework holds promise for capturing and replicating the complexities inherent in three-dimensional data. This work expands the application of GANs into the realm of point clouds, showcasing their potential impact on diverse applications such as 3D object generation and scene reconstruction. The contributions of Point Cloud GAN provide valuable insights into the synthesis of realistic point cloud data, contributing to the ongoing advancements in generative modelling and 3D data representation.

In the dynamic landscape of 3D data generation, the pioneering paper "Point Cloud GAN" by Chun-Liang Li et al. [22] not only represents a significant stride but also unfolds a rich tapestry of advancements. This innovative work harnesses the power of Generative Adversarial Networks (GANs) in the synthesis of point cloud data, pushing the boundaries of what is achievable in the realm of 3D data representation.

Addressing Specific Challenges, Chun-Liang Li et al meticulously tackle the unique challenges associated with generating realistic and coherent point cloud representations. Unlike traditional 2D images, point clouds demand a nuanced approach to capture spatial intricacies and maintain structural integrity. The paper delves into the intricacies of adapting GANs to address these challenges, demonstrating a deep understanding of the complexities involved in synthesizing three-dimensional data. The implications of Point Cloud GAN extend far beyond the confines of its inception.

The paper highlights the potential impact of this novel framework on diverse applications, including 3D object generation and scene reconstruction. This versatility positions Point Cloud GAN as a valuable tool with applications spanning virtual reality, robotics, and computer-aided design. The work of Chun-Liang Li et al contributes significantly to the ongoing advancements in generative modelling. By successfully applying GANs to point clouds, the paper enriches the toolkit available for researchers and practitioners working on generative models, expanding the horizons of what can be achieved in the synthesis of complex and realistic 3D data.

The paper 'Point Cloud GAN' not only addresses challenges specific to point cloud synthesis but also pioneers solutions with far-reaching implications across diverse applications, making it a cornerstone in the evolution of 3D data representation and generative modelling.

Additional, variation of GANs called tree-GANs was introduced in the paper '3D Point Cloud Generative Adversarial Network Based on Tree Structured Graph Convolutions' [36], the authors introduce a novel generative adversarial network (GAN) called tree-GAN for generating 3D point clouds. To achieve state-of-the-art performance, they propose a tree-structured graph convolution network (TreeGCN) as a generator for tree-GAN. This network performs graph convolutions within a tree, enabling it to utilize ancestor information to enhance the representation power for features. Moreover, they introduce a new evaluation metric called Fréchet Point Cloud Distance (FPD) specifically designed for accurately assessing GANs for 3D point clouds. This metric considers both the local and global structure of the point clouds. Experimental results demonstrate that the proposed tree-GAN outperforms existing GANs in terms of both conventional metrics and FPD. Moreover, the method can produce point clouds for various semantic parts without the need for prior knowledge. To summarize, the paper presents a novel GAN architecture and an effective evaluation metric, achieving state-of-the-art performance in 3D point cloud generation while maintaining flexibility and versatility.

CPCGAN proposed by Ximing Yang1 ,Yuan Wu, Kaiyi Zhang1 and Cheng Jin in the paper 'CPCGAN: A Controllable 3D Point Cloud Generative Adversarial Network with Semantic Label Generating' [42] is a revolutionary approach to 3D point cloud generation that utilizes a two-stage GAN framework to produce realistic, controllable, and semantically labelled point clouds from random latent codes. The first stage, the structure point cloud generator, creates a sparse representation with semantic labels, capturing the essential structural information of the target shape. The second stage, the dense point cloud generator, takes this processed structure point cloud and generates a denser point cloud with semantic labels, incorporating the detailed structures and semantic information from the structure point cloud.

This two-stage approach allows CPCGAN to achieve higher fidelity and more accurate semantic segmentation compared to existing point cloud GANs. The latent code-based control mechanism enables users to adjust the generated point cloud's appearance and attributes by modifying the latent code, providing a powerful tool for shaping and customizing 3D models. Additionally, the semantic segmentation branch ensures that the generated point clouds preserve the semantic relationships between points, making them more meaningful and interpretable for applications in 3D reconstruction, shape generation, and 3D object modelling. The CPCGAN's versatility is further demonstrated by its ability to generate point clouds from a wide range of datasets thanks to a pre-trained encoder network that extracts features from the training data. This adaptability makes CPCGAN a versatile tool for generating realistic 3D point clouds for diverse object categories. The CPCGAN's innovations have the potential to revolutionize the field of 3D point cloud generation, paving the way for a new era of generative models that can effectively capture and manipulate complex 3D structures.

3.3 Point Cloud Representation and Processing

An important element of synthetic image generation is the representation and processing of point clouds. With PointNet, Qi et al. (2017) [5] proposed a groundbreaking approach to directly processing point clouds, without relying on predefined structures like voxel grids. PointNet directly operates on individual data points, eliminating the need for structured representations. This innovative approach allows for the effective examination of unordered point clouds, addressing the challenges posed by their irregular nature. PointNet's neural network architecture enables direct assimilation of raw point cloud data, making it well-suited for capturing complex 3D structures without the constraints of predefined grids. This paradigm shift has significantly influenced point cloud processing, providing a foundation for more versatile and efficient applications, including its integration into the context of synthetic image generation.

This novel approach has aided in the more effective examination of unordered point clouds. With PointNet++, Qi et al. (2017) extended this work by integrating hierarchical processing for enhanced feature extraction from point clouds, tolerating the changing densities and complicated structures that are common in real-world 3D data.

Few of the applications of synthetic data generated from point clouds are:

1. Robotics: Synthetic image generation has profound implications for robotics, partic-

ularly in the development and testing of perception systems. Realistic synthetic images derived from point clouds can facilitate the training of robotic vision models, enabling more robust object recognition and scene understanding. This notion aligns with the work of Wang et al. (2020) [39] in employing GANs for realistic synthetic data generation in robotics applications.

2. Computer-Aided Design (CAD): In the realm of CAD, the ability to generate high-fidelity synthetic images from point clouds holds significant promise. Designers and engineers can benefit from realistic visualizations that aid in the conceptualization and refinement of 3D models. Synthetic images contribute to the validation of design choices and accelerate the prototyping process, by leveraging synthetic data for CAD applications [27].

3. Augmented Reality (AR): Synthetic images derived from point clouds are instrumental in enhancing the visual quality and realism of augmented reality applications. Whether overlaying virtual objects onto real-world scenes or providing immersive experiences in architectural visualization, realistic synthetic images play a crucial role in elevating the visual fidelity of AR environments. This aligns with the findings of Le, Robert and Nguyen, Minh and Yan, Weiqi [21] who explored the integration of GANs in improving visual quality for AR applications.

4 Theoretical Concepts

In this chapter of the thesis, we will discuss the theoretical background for generating synthetic data using deep learning. The chapter is split into three sections, in the first section, we will explore Point Cloud and its positive and negative aspects, next will focus on a well-known image generative approach Generative Adversarial Networks (GAN); lastly will look into a variation of GAN called Wasserstein Generative Adversarial Networks (WGAN) which tackles the challenges of traditional GANs.

4.1 Point Cloud

Point cloud data is a collection of spatial points that are represented as X, Y, and Z coordinates. This data can be used to create a 3D surface that can be used to map terrain, buildings, roads, and objects [1]. These may include further attributes like surface normals and RGB values [3]. Each point on the object's surface represents a single spatial measurement. A point cloud depicts an object's full exterior surface when all of its points are added together. An example of a point cloud is shown in Figure 1. Color information may be added to the point cloud if the RGB value of each point is captured. A 3D scanner, lidar, or photogrammetry software is used to generate a point cloud. Surface reconstruction allows it to be translated into the common forms of mesh models, CAD models, or NURBS surface models [15].

Point cloud data represent a convenient format for representing the 3D world. Point clouds are commonly used as a data format in several disciplines such as geomatics /surveying (mobile mapping); architecture, engineering, and construction; and Building Information Modelling (BIM) [35] [37] [38]. Point clouds have a range of applications in different areas such as robotics [28], autonomous driving [6], augmented and virtual reality [26], manufacturing and building rendering [13], etc.

4.1.1 Benefits and Drawbacks of point clouds

We need to first establish a fundamental understanding of the dynamic nature of point clouds before we dive into the nuanced exploration of deep learning. We will explore the inherent strengths that render point clouds valuable for 3D data representation, along with the challenges that need to be considered carefully in the theoretical landscape of deep learning.

As previously stated, a point cloud is a collection of data points in a three-dimensional coordinate system. These points indicate the surfaces or characteristics of a space-bound entity. Each point in the cloud can be connected with extra information, such as color or intensity. They are often produced using techniques such as LiDAR (Light Detection and Ranging), structured light scanning, or stereo vision. For example,

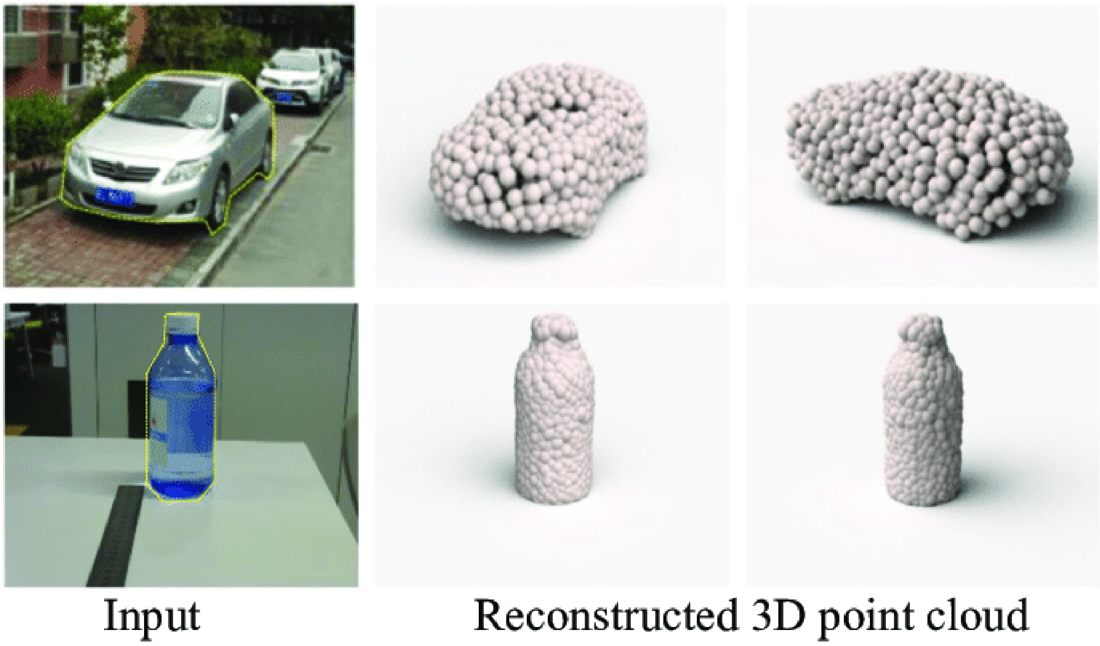


Figure 1: Point cloud [14]

LiDAR sends laser beams and measures the time it takes for the laser to return, resulting in a three-dimensional picture of the surroundings. They are classified as sparse or dense. Sparse point clouds have fewer data points and may depict large-scale landscapes, whereas dense point clouds have a high point density and can capture minute features.

Some of the advantages of point clouds include:

1. 3D Spatial Representation: Point clouds are a natural representation of three-dimensional (3D) spatial data that accurately depicts real-world surroundings [30].

2. Realistic Data Capture: Point clouds succeed at capturing complicated geometry, which makes them ideal for applications that need exact depictions of actual things.

3. Spatial Relationship Understanding: Point clouds inherently maintain spatial connections, allowing deep learning models to grasp and use data's 3D structure [9].

Moving on to weaknesses:

1. Data Sparsity: Point clouds may be sparse, especially in locations with fewer data points, which can put deep learning models to the test [2].

2. Limited Standardization: Models cannot be universally applicable due to a lack of standardized formats for point cloud data.

3. Computational Complexity: Processing large-scale point cloud data can be computationally intensive, requiring substantial resources for training deep learning models.

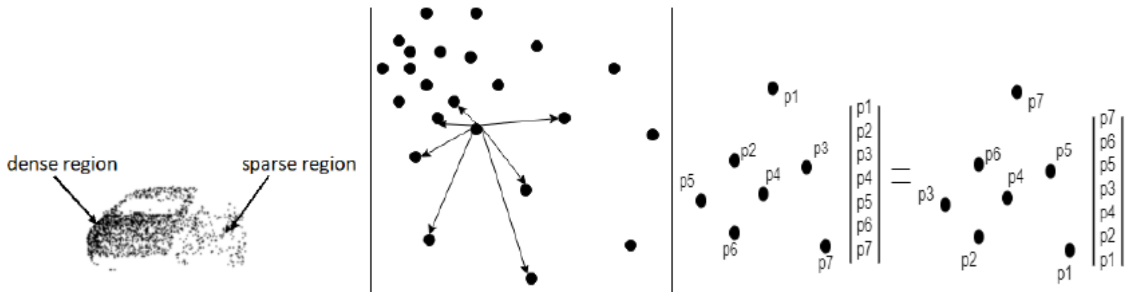
4.1.2 Challenges of Deep Learning with Point Clouds

There are numerous challenges to applying deep learning to 3D point cloud data. These issues include occlusion, which arises when certain parts of the scene or objects are hidden from the sensor, leading to incomplete point cloud data; noise/outliers, which are unintentional and incorrect points introduced during the data collecting process is referred to as noise. Data points that substantially deviate from the predicted trends are called outliers and point misalignment; the absence of connection or inconsistency in the spatial arrangement of points between several scans or frames are a few of the challenges [18] [25]. However, the most significant issues concerning are:

1. Irregularity: Points are not equally sampled over various parts of an object/scene, resulting in dense points in some places and sparse points in others [31]. Figure 2(a) shows examples of them. It can be addressed by using sub-sampling techniques [10].

2. Unstructured: Point cloud data is not organized into a regular grid [23]. Each point is scanned separately, and the distance between them is not always constant. Pixels in photographs, on the other hand, are represented on a two-dimensional grid, and the distance between two neighbouring pixels is always set. Is it depicted in the Figure 2(b).

3. Unorderdness: A point cloud of a scene is the set of points (represented as XYZ) obtained around the objects in the scene, and these are generally stored as a list in a file. As a set, the order in which the points are stored does not change the scene represented; therefore, it is invariant to permutation [5]. For illustration purposes, the unordered nature of point sets is shown in Figure 2(c).



(a) Irregular. Sparse and dense regions. (b) Unstructured. No grid; each point is independent and the distance between neighboring points is not fixed. (c) Unordered. As a set, point clouds are invariant to permutation.

Figure 2: Challenges of point cloud data [20]

4.2 Generative Adversarial Networks (GANs)

GANs, or Generative Adversarial Networks, are a generative modelling approach that uses techniques from deep learning such as convolutional neural networks. It involves discovering and learning regularities or patterns in input data so that the model can then be used to generate or output new examples that are plausible to have been drawn from the original data. It is an unsupervised learning task in machine learning. Generative adversarial networks are based on a game, in the sense of game theory, between two machine learning models, typically implemented using neural networks [8]. The two neural networks that make up a GAN are called the generator and discriminator. The discriminator is a deconvolutional neural network, and the generator is a convolutional neural network. The generator is designed to produce outputs that are easily mistaken for real data. A discriminator identifies artificial outputs by determining which were generated artificially.

The GAN's working is based on three principles, firstly to make the generative model learn, so that the data can be generated employing probabilistic representation. Secondly, the training of the model unfolds in an adversarial setting, characterized by continual competition between the generator and discriminator. Lastly, by using the deep learning neural networks and using the artificial intelligence algorithms for training the complete system [17] to make the generator better at creating realistic data.

4.2.1 Architecture of GAN

As mentioned before, a Generative Adversarial Network (GAN) is made up of two major components: the Generator and the Discriminator. A detailed overview of the architecture is depicted in Figure 3.

4.2.2 Generator Model

The generator model is an essential part of a Generative Adversarial Network (GAN), which is in charge of producing precise data. Using random noise as input, the generator creates complex data samples, such as text or images. Through training, its architecture's layers of learnable parameters capture the training data's underlying distribution. The generator uses backpropagation to fine-tune its parameters as it is trained to produce samples that closely resemble real data, and it modifies its output accordingly. What makes the generator successful is its capacity to produce varied, high-quality samples that can trick the discriminator.

4.2.3 Generator Loss

The generator reduces the log-likelihood that the discriminator is accurate for samples that are generated. The generator is motivated to produce samples that the discriminator is likely to classify as real as a result of this loss. The generator gathers random noise samples during training and produces an output from that noise. After that, the output is run through the discriminator, which classifies it as "Real" or "Fake" depending on how well it can distinguish between the two. The generator loss is then calculated based on the discriminator's categorization; if it is successful in tricking the discriminator, it is rewarded; if not, it is penalized [12].

To train the generator, the following equation is minimized:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} log(1 - D(G(z^{(i)})))$$

where G is the generator, D is discriminator and z is a sample from the latent space, G(z), D(G(z)) denote the generator's output when it receives as input noise z, and the discriminator's probability that a synthetic sample G(z) of data is real and θ_g is a hyperparameter of a multilayer perception that represents a differentiable function $G(z; \theta_g)$ which maps input noise z to data space.

4.2.4 Discriminator Model

A discriminator model, a type of artificial neural network, is used in Generative Adversarial Networks (GANs) to differentiate between generated and real input. The discriminator evaluates incoming samples and assigns a probability of authenticity, thereby functioning as a binary classifier. With time, the discriminator gains the ability to discriminate between real data from the dataset and fake samples produced by the generator. This enables it to progressively adjust its settings and raise its level of expertise. Convolutional layers or pertinent structures for other modalities are usually used in its architecture when handling image data. The goal of the adversarial training method is to increase the discriminator's capacity to correctly classify real samples as authentic and generated samples as fraudulent. The discriminator becomes extremely discriminating as a result of the generator and discriminator's interaction, which enab les the GAN to generate synthetic data that overall looks incredibly realistic.

4.2.5 Discriminator Loss

The discriminator reduces the negative log chance of correctly identifying both synthetic and real samples, much like the generator does. The discriminator loss, as opposed

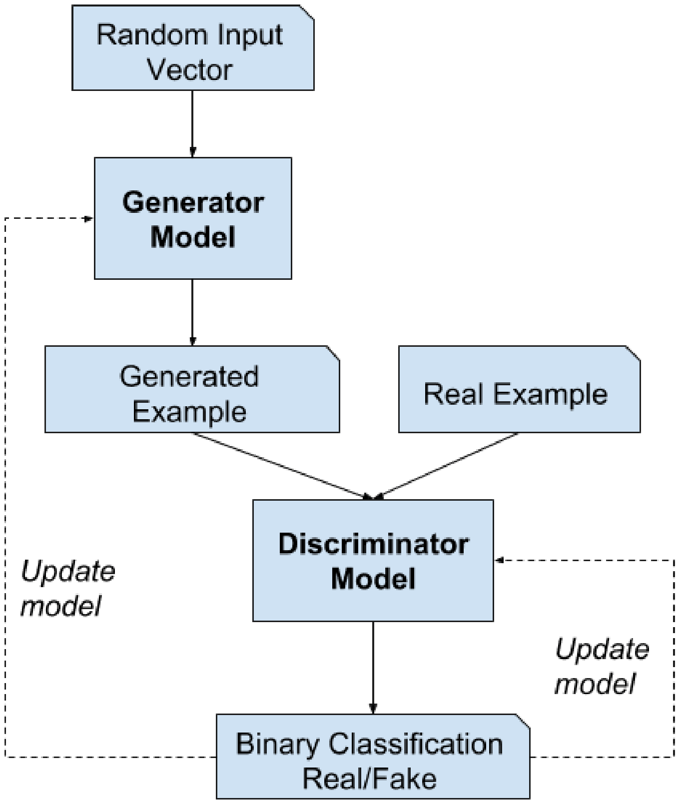


Figure 3: Architecture of GAN [4]

to the generator loss, causes the discriminator to correctly classify generated samples as fake. The discriminator classifies generated and real data while it is being trained.

By maximizing the below function, it penalizes itself for misclassifying a genuine instance as fake or a fake instance (made by the generator) as real [12].

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} [log\ D(x^{(i)}) + log(1 - D(G(z^{(i)})))]$$

where G is the generator, D is discriminator and z is a sample from the latent space, log(D(x)) refers to the probability that the generator is rightly classifying the real image, maximizing log(1-D(G(z))) would help it to correctly label the fake image that comes from the generator.

4.2.6 GAN Loss - Minmax Loss

The standardised GAN loss is the min-max loss [17]. Whereas the discriminator seeks to maximize this function, the generator seeks to minimize it. When viewed as a minmax game, this way of expressing the loss seemed reasonable. In actuality, it saturates for the generator, which means that if it cannot keep up with the discriminator, the generator frequently stops training [12].

$$\min_{G} \max_{D} V(D,G) = \mathbb{E}_x[log\ D(x)] + \mathbb{E}_z[log(1 - D(G(z)))]$$

where G is the generator, D is discriminator and z is a sample from the latent space, G(z), D(x), D(G(z)) denote the generator's output when it receives as input noise z, the discriminator's probability that the original data x is real, and the discriminator's probability that a synthetic sample G(z) of data is real, and E_x, E_z denote the mean likelihood over all original data and synthetic data respectively.

4.2.7 Challenges in Training GANs

In recent years, generative adversarial networks have emerged as powerful tools for generating realistic data, with applications ranging from image synthesis to data augmentation. The training process of GANs, however, is rife with challenges that need to be faced carefully.

1. Mode Collapse: This occurs when the generator is not capable of capturing the entire complexity of the training data distribution, causing mode collapse. A challenge such as this hampers the ability of the GAN to generate a diverse and representative sample of information [7].

2. Vanishing/Exploding Gradients: When trained, GANs are prone to vanishing or exploding gradients, which can lead to slow convergence [7].

3. Training Instability: As hyper-parameter choices impact training, GANs can exhibit oscillations or divergence during training [11].

4. Evaluation Metrics: The quality and diversity of generated samples may not be adequately captured by traditional evaluation metrics, making it difficult to assess the performance of GAN objectively [11].

4.3 Wasserstein Generative Adversarial Networks

The goal of the Wasserstein Generative Adversarial Network (WGAN) is to improve training stability and the calibre of generated samples by introducing notable innovations to address issues with conventional GANs. The main distinction from conventional GANs is the Wasserstein distance, which is used as a metric to measure how different two probability distributions are from one another. In contrast to traditional GAN, the wasserstein distance offers a more continuous and significant measure of distribution dissimilarity. In contrast to traditional GAN algorithms, the WGAN algorithm delivers significant practical benefits. Figure 4 provides a comprehensive depiction of the architectural details of WGAN. The real image represents the input to the discriminator, a neural network tasked with distinguishing between real and fake data. The discriminator aims to output a high score for real data and a low score for fake data. The critic, a type of discriminator specifically employed in WGANs, functions as a single-output neural network. The critic's output, a single real number, corresponds to the Wasserstein distance between the input data and the real data distribution. The generator, another neural network, strives to produce fake data that resembles real data. The generator is trained by minimizing the generator loss function.

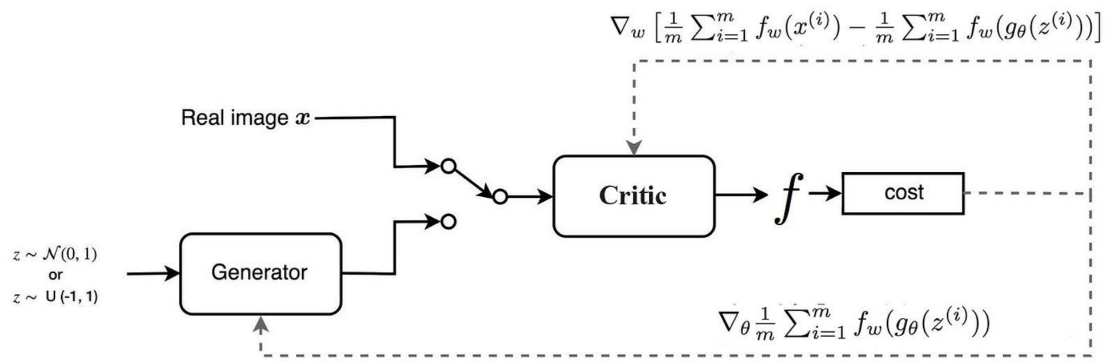


Figure 4: Architecture of WGAN [19]

The novel loss function that forms the basis of WGAN is commonly known as the Wasserstein loss, or W-loss. The goal of the generator is to reduce the negative wasserstein distance between the generated data distribution and the distribution of real data. Conversely, this wasserstein distance is maximized by training the discriminator. The training dynamics are significantly affected by this switch from binary classification to distance optimization, which provides a more reliable and instructive signal for the discriminator and generator. Weight clipping is a mechanism introduced by WGAN to guarantee the existence of the Wasserstein distance and to enforce Lipschitz continuity which is often introduced to ensure the stability of the training process. The Lipschitz continuity constraint helps in controlling the gradients of the neural network, which can be crucial for the convergence of the optimization algorithms used during training.

In WGANs, enforcing Lipschitz continuity is one approach to overcome some of the challenges associated with traditional GAN training. More specifically, during training, the discriminator's weights are limited to a range. By encouraging the discriminator to be Lipschitz continuous, this regularization method helps to overcome certain difficult-

ies gradient control, convergence issues, mode collapse which arise from the existence of the wasserstein distance. The discriminator and generator are updated iteratively as part of the WGAN training process. The wasserstein distance is maximized by updating the discriminator's parameters, and the negative wasserstein distance is minimized by updating the generator's parameters. By combining this dual optimizat-
ion strategy with weight clipping to enforce Lipschitz continuity, the likelihood of mode collapse and other training instabilities is decreased and convergence becomes more stable. Compared to conventional GANs, Wasserstein GAN has shown to have several advantages like it have improved training stability, which reduces its sensitivity to hyperparameter selections and results in more stable learning dynamics.

By addressing mode collapse-related concerns [24] and encouraging a wider diversity in the generated outputs, the wasserstein distance also motivates the generator to produce higher-quality samples. It is also called as Earth Mover's distance, or EM distance, because it can be interpreted as a minimum energy cost of moving dirt in one probability distribution into another probability distribution. Dirt movement and moving distance are used to quantify the cost [40].

5 Implementation

This section delves into the implementation details of approaches to generate synthetic 3D point cloud data, employing traditional Generative Adversarial Network (GAN) and Wasserstein Generative Adversarial Network (WGAN). While these two approaches have demonstrated remarkable success in this domain, the pursuit of methodological diversity remains imperative to ensure robustness and applicability across various domains. In this section, we shall also explore other techniques to generate synthetic data besides GAN-based approaches which is PointNet which was not successful in producing synthetic point clouds. A comprehensive exploration of these three approaches is conducted throughout our implementation for the generation of synthetic 3D point clouds. This includes architectural choices, training strategies, and hyperparameter tuning tailored to each model. Here, we will examine our methodologies in-depth, detailing how the generative models were trained and fine-tuned.

5.1 Groundtruth Data

The White Light Interferometer is used to scan the combustion chambers during aircraft maintenance. The 3D data captured during the scan is then projected onto a 2D space for the purpose of crack detection, and conventional image processing is then used to identify the cracks. The point cloud of the real cracks captured from the surface of an aircraft combustion chamber is shown in the Figure 5.

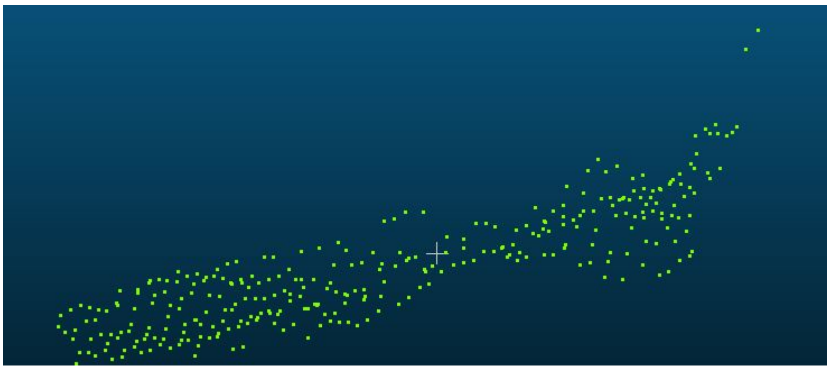


Figure 5: Groundtruth Point Cloud

There are 164 point clouds of cracks and the aim is to generate multiple point clouds of cracks. Each point cloud has four primary properties: X, Y, Z coordinates and a scalar value. These coordinates define the position of each point in a three-dimensional space. The X coordinate represents the horizontal position, the Y coordinate represents the vertical position, and the Z coordinate represents the depth position. The scalar value specifies the column in the input file that contains the scalar field values for the points. Scalar fields are additional properties that can be associated with points, such as intensity, color, or temperature.

5.2 Generative Adversarial Network (GAN) Approach

5.2.1 Architecture Design

Generator Architecture:

A sequential model for the Generative Adversarial Network (GAN) is defined in the generator architecture clearly illustrated in Figure 6. The generator network, responsible for creating new point clouds, is designed with a series of layers that process and transform the input noise into a realistic-looking point cloud. The network comprises of ten layers. The input layer receives the input data, which is a 100-dimensional vector. The dense layer 3 then applies a linear transformation to the input data and outputs 256 latent vectors. The leaky ReLU 2 activation function is then applied to the outputs of the dense layer 3. This activation function allows for a small amount of non-zero output even when the input is negative, which can help to prevent the network from dying during training. The batch normalization layer then normalizes the outputs of the leaky ReLU 2 activation function. This helps to improve the performance of the network by making it less sensitive to the initialization of the weights.

The dense layer 4 then applies a linear transformation to the outputs of the batch normalization layer and outputs 512 latent vectors. The leaky ReLU 3 activation function is then applied to the outputs of the dense layer 4. The batch normalization layer 1 then normalizes the outputs of the leaky ReLU 3 activation function. The dense layer 5 then applies a linear transformation to the outputs of the batch normalization layer 1 and outputs 2000 latent vectors. The reshape layer then reshapes the outputs of the dense layer 5 into a 500x4 matrix. The output layer then applies a linear transformation to the outputs of the reshape layer and outputs 500 latent vectors.

By opting for a 10-layer architecture, the generator demonstrated the capability to learn rich and abstract features while facilitating a smooth convergence during training. It was observed that a 10-layer architecture struck a harmonious balance between model capacity and training efficiency.

Discriminator Architecture:

The discriminator network comprises six primary layers that effectively distinguish between real and generated images. The initial Flatten layer transforms the input data, a 2D matrix representing the image, into a 1D vector. This flattens the data into a more manageable format for subsequent processing. Next, a dense layer with 512 neurons applies linear transformations to the flattened data, generating a 512-dimensional vector. LeakyReLU activation is employed, allowing for a small positive output even for negative inputs, contributing to better training stability.

To further refine the representation, a second dense layer with 256 neurons applies linear transformations and utilizes LeakyReLU activation. This process enhances the network's ability to capture subtle image characteristics. A dense layer with 1 neuron concludes the network, employing a sigmoid activation function. This single output represents the probability that the input image is real or fake. The sigmoid function ensures that the output lies between 0 and 1, reflecting the confidence of the discriminator.

5.2.2 Phases of GAN model

Moving on, three separate phases make up the organization of the GAN model. First, during the data preprocessing phase, the input data for the GAN model is prepared. Reading 3D point cloud data from files, processing each file's contents to create a list of 3D point coordinates, and normalizing the data are all necessary steps in the process. Next is model training, which involves using the prepared dataset to teach the GAN model the underlying structures and patterns of the 3D point clouds. The final step is data generation, which entails creating new, artificial 3D point cloud data using the trained generator.

5.2.3 Data Preprocessing

The content of each 3D point cloud file is processed in the first step of the data preparation process to produce a list of 3D point coordinates and a list of lists with floating-point values, which ensure that the raw data is correctly formatted and parsed for further processing.

The maximum values for the x, y, and z coordinates across all of the dataset's data points are computed after data extraction. It determines the maximum values for each dimension by going through each data point iteratively. The entire dataset is then normalized using these maximum values. By guaranteeing that every data point is scaled appropriately, normalization makes it easier to provide the GAN model with a consistent input during training. Next, NumPy arrays containing the normalized dataset and the normalization vector are saved.

5.2.4 Training Procedure

Training Loop:

The training loop iterates through 1000 epochs. The discriminator and generator are trained alternately within each epoch. Real and generated data batches are processed during each iteration.

Discriminator Training:

Real data batches are chosen at random from the normalized dataset in order to train the discriminator. These real data batches have a value of 1 with a label "real". To accurately classify these real data batches, the discriminator is trained on them. The generator generates synthetic data batches and noise vectors simultaneously. These artificial data batches have a value of 0 with a label "fake". These artificial data batches are then used to train the discriminator. The average of the losses on batches of real and fake data is used to calculate the discriminator's overall loss.

Generator Training:

The generator is taught to generate fake data in order to trick the discriminator. The GAN model receives generated random noise vectors as input. For these produced data batches, the target labels have a value of 1 and are set to "real". The weights of the generator are adjusted in accordance with the discriminator's calculated loss. The goal is to maximize the discriminator's error, encouraging the generator to generate point clouds that are more difficult for the discriminator to distinguish from real ones.

Adversarial Training:

The fundamental idea here is that the generator is regularly trying to enhance its capacity to build realistic point clouds that can mislead the discriminator. Although the discriminator is not yet trainable during the generator training step, it has already learned to be a skilled judge of realism through previous training. This dynamic interaction is iterative. The generator improves its capacity to make realistic samples, and the discriminator adapts to better discriminate between real and generated samples. Finally, the generator model is saved into a H5 file for further process,

5.2.5 Data Generation

Following the training phase, the pre-trained generator model is loaded and 20 synthetic point clouds are generated with random noise vectors are produced as input. Random noise is fed into the pre-trained generator to stimulate diversity and creativity in the generated samples. This input acts as a point in the latent space, allowing the generator to map it to a variety of outputs. The introduction of random noise helps to avoid problems like mode collapse, enhances generalization to previously undiscovered patterns, and facilitates adversarial learning by pushing the discriminator to differentiate between actual and produced samples. Next, using the previously established normalization vector, the generated data is normalized to maintain the consistency and same data distribution as the training data. The artificial 3D point clouds are stored in files with a predetermined format so they can be examined and used later.

5.2.6 Hyper-parameter Tuning

Hyperparameter tweaking is critical in enhancing the performance of both the generator and discriminator networks in the context of the GAN model developed for point cloud data creation. The latent dimension, a defining factor set at 100, determines the size of the input noise vectors injected into the generator. This dimensionality is critical in determining the variety and complexity of the generated point clouds. Furthermore, the Adam optimizer's learning rate of 0.0002 and beta value of 0.5 act as fundamental parameters in guiding the optimization process. Another important hyper-parameter, the batch size, is 32. This parameter affects both training dynamics and computational efficiency because it specifies how many samples are processed in each iteration. Finally, the 1000 number of training iterations across the whole dataset is set by the epochs hyper-parameter. It takes 1000 epochs for the model to learn and converge.

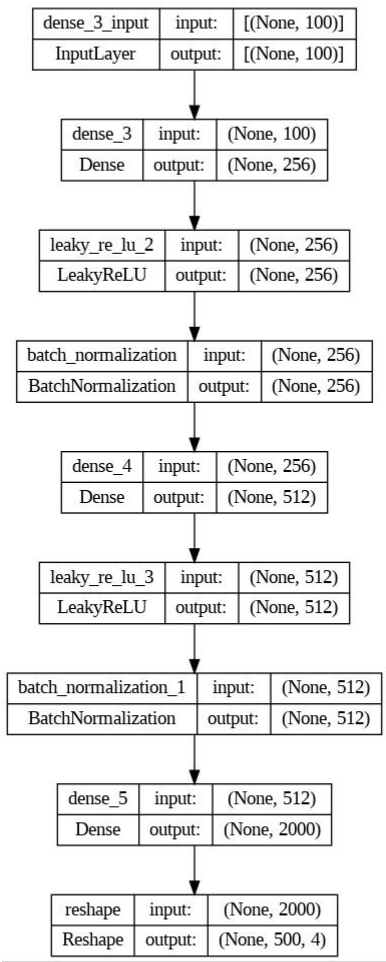


Figure 6: Generator Network Architecture of traditional GAN

5.3 Wasserstein GAN (WGAN) Approach

5.3.1 Architecture Design

Generator Network:

In the WGAN approach, the generator encompasses eleven layers, each serving a specific purpose in the image classification process. The flattened data is received as a 100-dimensional vector by the input layer, which represents the image as a sequence of pixel values. Dense layer 2 transforms this vector into a higher-dimensional representation using 256 neurons, enabling the network to extract more abstract features from the raw data. Dense layer 3 improves the representation even more by increasing the number of neurons and employing a ReLU activation function, preserving the crucial features while reducing the dimensionality. Convolutional layer 1D 2 extracts features from the image using 64 filters, sliding across the image to generate a set of feature maps that capture various patterns and textures. Leaky ReLU layer 2 applies a non-linear transformation to the output, introducing a slight positive slope for negative inputs, preventing gradient vanishing during training and improving the network's ability to learn. Convolutional layer 1D 3 further refines the feature maps generated by the previous layer with 32 filters to extract finer-grained details. This process helps to identify more complex patterns and enhances the network's ability to distinguish between different classes. Leaky ReLU layer 3 applies another non-linear activation function to maintain the network's ability to capture complex patterns and prevent gradient vanishing issues.

Convolutional layer 1D 4 reduces the number of feature maps to a more manageable size by using only 3 filters, enabling the network to focus on the most relevant features. This step helps to reduce the dimensionality of the representation and prepare it for the final classification step. The output layer receives the reduced representation and applies a softmax activation function, normalizing the output to represent the probability of each class. This allows the network to make a final prediction about the object or scene depicted in the image.

In the design of the Wasserstein Generative Adversarial Network (WGAN), the decision to employ an 11-layer architecture for the generator was driven by a combination of task complexity, dataset characteristics, and architectural considerations. Empirical investigations were conducted to explore different layer configurations, revealing that an 11-layer generator architecture provided the necessary expressive power to model the intricate relationships within the data. The increased depth allowed for the extraction of hierarchical features, enabling the generator to produce high-quality and diverse synthetic samples.

Discriminator Network:

This discriminator network comprises eight layers, each serving a distinct purpose in

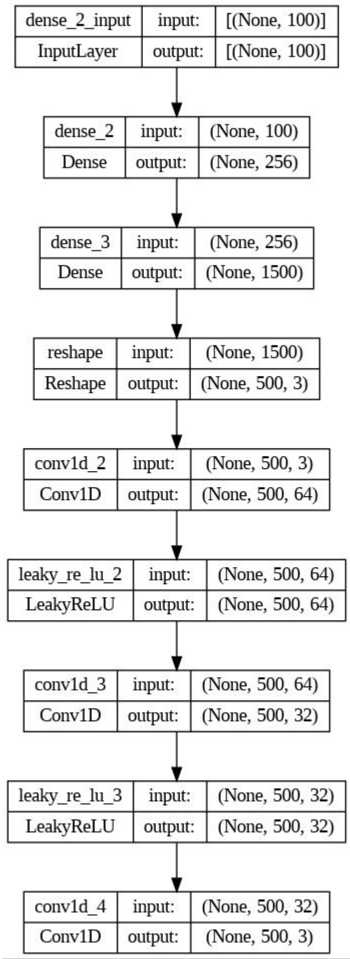


Figure 7: Generator Network architecture of WGAN

the classification process. The input layer receives the data as a three-dimensional tensor with dimensions (None, MAX POINTS, 3), representing the data sample as a collection of XYZ coordinates. The first convolutional layer (Conv1D(64, 5)) engages 64 filters with a kernel size of 5, sliding across the input tensor to extract local features. The 'same' padding ensures that the output tensor retains the spatial dimensions of the input tensor. A LeakyReLU(0.2) activation function transforms the output of the convolutional layer. This non-linear activation introduces a small positive slope for negative inputs, preventing gradient vanishing and enhancing the network's learning capacity.

The second convolutional layer (Conv1D(128, 5)) utilizes 128 filters with the same kernel size, maintaining the spatial dimensions of the representation. The filters extract more intricate features from the data. Another LeakyReLU(0.2) activation function follows the second convolutional layer, preserving the non-linearity and preventing gradient vanishing issues. The flatten layer flattens the output tensor into a one-dimensional vector, facilitating the connection between convolutional and fully connected layers.

The first fully connected layer introduces 256 neurons and applies a ReLU activation function to the flattened vector, transforming it into a higher-dimensional representation. The final fully connected layer comprises a single neuron, applying a sigmoid activation function. This outputs a value between 0 and 1, indicating the probability that the input data is genuine. This value serves as the foundation for determining whether the data is real or synthetic.

5.3.2 Phases of WGAN model

The Wasserstein Generative Adversarial Network (WGAN) model, similar to the traditional GAN, is divided into three main phases: data preprocessing, model training, and data generation.

5.3.3 Data Preprocessing

This process involves the systematic extraction and organization of information from multiple files within a specified directory. The mean values are calculated for each 3D point cloud in the dataset. The mean values are calculated independently along each dimension (x, y, z) of the 3D points. The mean values of the data are subtracted from each dimension to centre the data around the origin. This process is often employed to remove biases or to standardize the data for improved model training. The mean values calculated represent the central tendencies of the point cloud data in terms of its spatial distribution and these mean values are stored in an array to use in the further process.

5.3.4 Training Procedure

Training loop:

Unlike traditional GAN approach, the WGAN approach iterates through 5000 epochs. The decision to train the Wasserstein Generative Adversarial Network (WGAN) for 5000 epochs was based on empirical observations like the generator and training loss, stability, convergence and the quality of the generated point clouds.

Discriminator Training:

The discriminator, in the context of this Wasserstein Generative Adversarial Network (WGAN), plays the role of a critic. Its objective is to assess the authenticity of both real and generated 3D point clouds. The discriminator training is conducted through multiple iterations within each epoch. Real point clouds are randomly sampled from the dataset and padded to a consistent size. The discriminator is then optimized to distinguish between real and generated point clouds by computing the wasserstein loss, using real labels of -1 for actual data. Simultaneously, synthetic point clouds produced by the generator are used to compute the wasserstein loss with labels set to 1. The discriminator's weights are then adjusted to maximise this loss. Importantly, a lipschitz constraint is enforced by clipping the weights of the discriminator within a specified range (-0.01 to 0.01). This iterative training process enhances the discriminator's ability to discern between real and generated point clouds.

Generator Training:

The generator, a crucial component of the WGAN, is trained to produce synthetic 3D point clouds that can convincingly deceive the discriminator. During the training loop, random noise vectors are input into the generator, and the wasserstein loss is computed using the output of the discriminator as the measure of how well the generated point clouds fool the discriminator. The generator aims to minimize this loss, effectively learning to create realistic 3D point clouds. The generator is updated using the wasserstein loss, and this process is iterated over multiple epochs. The periodic training of the generator ensures its continuous improvement in generating synthetic point clouds that exhibit characteristics similar to those in the real dataset. The final trained generator can subsequently be employed to generate new, realistic 3D point clouds with diverse spatial features.

Adversarial Training:

Adversarial training within the framework of Wasserstein Generative Adversarial Networks (WGAN) unfolds as a sophisticated switch between the generator and the critic, each playing a distinct yet interconnected role. Unlike traditional GAN, WGAN leverage the wasserstein distance to quantify the dissimilarity between the distributions of real and generated data. The discriminator, functioning as a discriminator in this context, undergoes training to assign low scores (negative values) to real data and high scores (positive values) to generated data. This training process is distinctive, employing the wasserstein loss with labels set to -1 for real data and 1 for generated data, steering the critic toward approximating the wasserstein distance accurately. Concurrently, the generator's task transcends the conventional objective of deceiving the discriminator with binary classifications. Instead, it strives to minimize the wasserstein loss, generating synthetic data so realistic that the discriminator perceives it similar to real data. The wasserstein loss encapsulates the wasserstein distance, fostering the convergence of the two data distributions. Critical to the stability of WGAN is the lipschtiz continuity of the discriminator, often enforced by weight clipping. This constraint ensures a well-defined wasserstein distance and contributes to training stability. The iterative nature of training involves multiple critic updates for each generator update, a process regulated by the 'n critic' parameter, which mitigates issues such as mode collapse and promotes a robust learning trajectory. Adversarial training in WGAN, propelled by the wasserstein distance, orchestrates a nuanced interplay between the generator and the critic, leading to the synthesis of diverse, high-fidelity data representations across diverse domains.

5.3.5 Data Generation

Similar to the GAN approach, the 20 synthetic point clouds are generated with a random noise vector and the pre-trained generator model.

5.3.6 Hyper-parameter Tuning

In the endeavour to train a Wasserstein Generative Adversarial Network (WGAN) for the synthesis of realistic 3D point clouds, the meticulous tuning of hyperparameters assumes a pivotal role in optimizing model performance and stability. The MAX POINT parameter, determining the size of 3D point clouds, is fine-tuned to a value of 500, ensuring uniformity in the input size for the discriminator. The latent dim hyperparameter, specifying the dimensionality of random noise vectors, is set to 100, influencing the diversity and complexity of the generated point clouds. The number of training epochs is established at 5000, with a batch size of 32, striking a balance between computational efficiency and model convergence. The clip value parameter, vital for enforcing lipschtiz continuity in the discriminator, is carefully adjusted to 0.01, stabilizing training and mitigating gradient-related challenges. The determination of the optimal n critic value, representing the number of discriminator iterations per generator iteration, is critical and set to 5 for achieving a well-trained and stable model. The RMSprop learning rate parameter, governing the learning rate in the optimizer, is set to 0.00005 to facilitate smooth convergence. Additionally, the architecture of the generator, featuring Conv1D layers, introduces complexity, and the number and size of these layers are tuned to enhance the model's capacity to capture intricate patterns in the data. This comprehensive and tuned suite of hyperparameters collectively orchestrates the training dynamics of the WGAN, ensuring the synthesis of high-quality 3D point clouds.

5.4 PointNet Approach

At first, the PointNet architecture which is well-known for its efficiency in segmentation and classification tasks seemed well-suited for the challenge of creating artificial point clouds. PointNet seems to be well-suited to manage the intrinsic difficulties associated with point cloud data because of its special permutation invariance, which was created to handle unordered point sets.

Even while the PointNet technique worked well for classification and segmentation, there were significant issues when trying to use it to create artificial point clouds. The permutation invariance of the architecture proved to be a drawback when attempting to represent the complex spatial linkages and global structures present in point cloud data, although being advantageous for handling unordered point sets. PointNet faced challenges since point clouds have no intrinsic order; it ignores any predetermined order or permutation in the input point cloud.

Furthermore, PointNet's architectural design may not be naturally suited for generative tasks that require the modeling of structured latent spaces in order to capture and recreate underlying data distributions. The selection of an appropriate loss function is critical for effective training in generative tasks, and the distinct properties of point clouds frequently need specialized distance measurements, such as Chamfer distance or Earth Mover's Distance (EMD). Mismatched loss functions may impair the model's capacity to learn and recreate desired point cloud distributions.

Given these considerations, while PointNet performs well in some domains, its subtle limits highlight the need for alternate designs or hybrid techniques. PointNet faces obstacles like as the unstructured nature of point clouds, complex spatial interactions, and the need for specialized loss functions. This acknowledgement emphasizes the importance of investigating alternate approaches adapted to the particular qualities of point cloud data, paving the way for more effective and successful synthetic point cloud generation in future attempts.

6 Results

The results section unfolds the outcomes of point cloud generation achieved through both traditional GAN and Wasserstein GAN methodologies. This chapter meticulously explores the synthetic point clouds, employing a dual perspective of conventional GAN and the Wasserstein GAN (WGAN) approaches. By training both models using 164 point clouds, Overall, 20-point clouds are generated. To assess the generated results, a quantitative analysis is carried out where Earth Mover's Distance, Mahalanobis Distance and Fréchet Point Cloud Distance evaluation metrics are applied. These metrics act as quantitative benchmarks, providing an evaluation of the authenticity and overall quality of the synthetic point clouds. By contrasting the results of traditional GAN and WGAN and subjecting them to thorough metric analysis and discussion on generator and discriminator loss plots, this section aims to offer a nuanced comprehension of each approach's effectiveness and distinct attributes in the domain of point cloud synthesis.

6.1 Quantitative Analysis

Figures 8 and 9 depict images of synthetic point cloud created through a conventional GAN methodology, while Figures 10 and 11 showcase synthetic point cloud generated employing a WGAN approach.

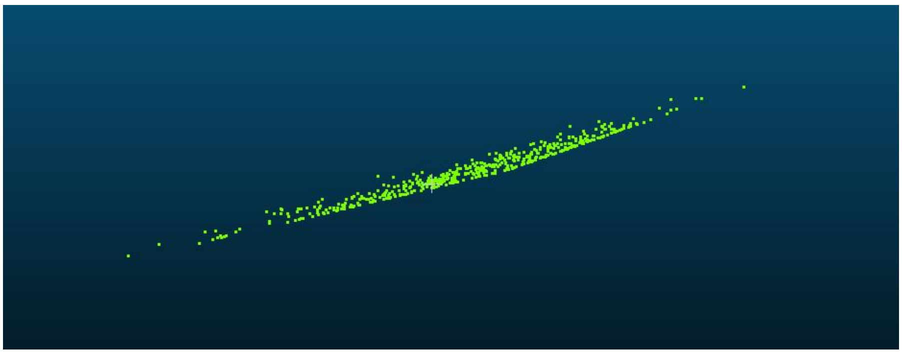


Figure 8: Synthetic crack generated by traditional GAN approach-1

6.1.1 Earth Mover's Distance

Earth Mover's Distance is used as one of the evaluation metrics to measure the reconstruction quality of the point clouds. The Earth Mover's Distance (EMD) is a measure of the dissimilarity between two probability distributions. It is often used to assess the similarity between the distribution of generated samples and the distribution of real

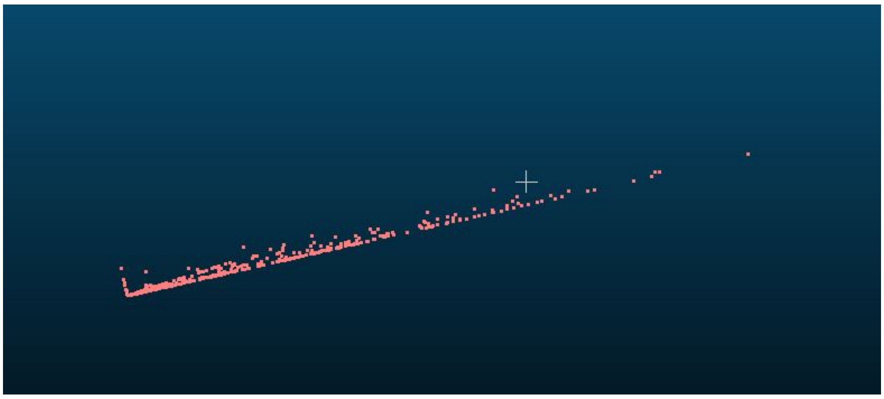


Figure 9: Synthetic crack generated by traditional GAN approach-2

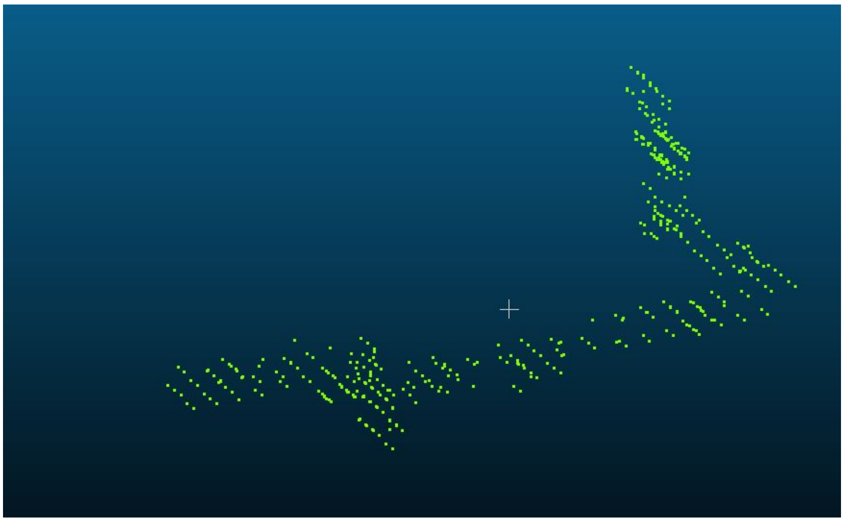


Figure 10: Synthetic crack generated by WGAN approach-1

samples. The basic idea is to imagine that each point in the distributions has a certain "mass" or "weight", and the goal is to find the most efficient way to move the mass from one distribution to the other. The distance metric is determined by the amount of mass moved and the distance it travels.

In the context of evaluating the efficacy of the traditional Generative Adversarial Network (GAN) in generating realistic 3D point clouds, an Earth Mover's Distance (EMD) comparison is conducted between the ground truth and generated dataset. This quantitative assessment involved measuring the dissimilarity between the two distributions, considering both spatial arrangement and overall distribution characteristics. The EMD computation was performed on normalized point clouds to ensure consistency in scaling. The minimum EMD value, signifying the closest match between generated and ground truth distributions, was recorded, along with the corresponding pair of normalized point clouds. This analysis not only quantifies the model's ability to replicate the underlying data distribution but also provides valuable insights into

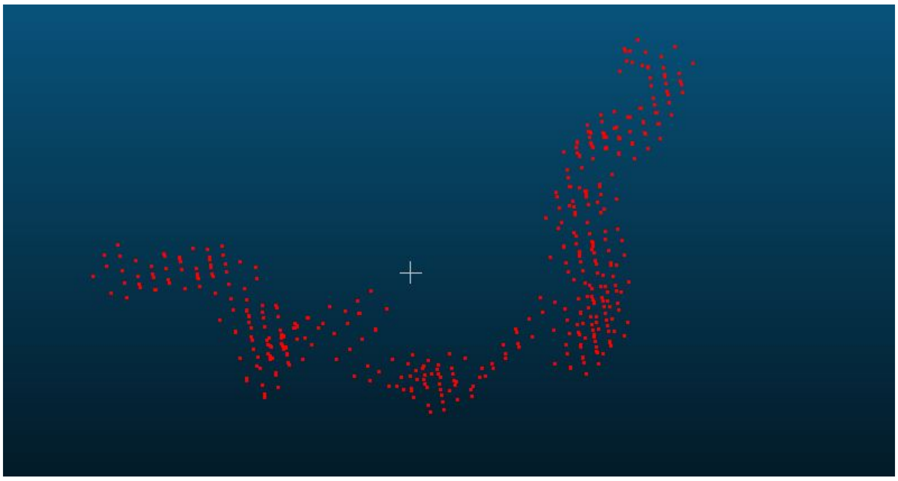


Figure 11: Synthetic crack generated by WGAN approach-2

the spatial coherence of the generated point clouds. Figure 12 and 13 are the ground truth and generated point clouds with minimum EMD value obtained using traditional GAN approach.

The Earth Mover's Distance (EMD) can be mathematically expressed as:

Given two probability distributions P and Q defined over a metric space X with distance function d(x,y), the Earth Mover's Distance between P and Q is defined as the minimum cost of transporting the mass P and Q, where the cost is proportional to the distance travelled.

Mathematically, it can be expressed as an optimization problem:

$$EMD(P,Q) = \min_{\gamma} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} \cdot \gamma_{ij}$$

Subject to the constraints:

1. $\sum_{j=1}^{n} \gamma_{ij} = p_i$ for all $i = 1, 2, \ldots, m$ (Mass conservation for P).

2. $\sum_{i=1}^{m} \gamma_{ij} = q_j$ for all $j = 1, 2, \ldots, n$ (Mass conservation for Q).

3. $\gamma_{ij} \geq 0$ for all $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$ (Non-negativity).

where,

$P = (p_1, p_2, \ldots, p_m)$ and $Q = (q_1, q_2, \ldots, q_n)$ are the probability distributions.

x_{ij} represents the amount of mass to be transported from i in P to j in Q.

$c_{ij} = d(x_i, y_j)$ is the cost (distance) of transporting one unit of mass from i to j, where

$d(x_i, y_j)$ is the distance metric between points x_i and y_j

Likewise, the Earth Mover's Distance (EMD) is computed for point clouds generated through the WGAN approach, and Figures 14 and 15 depict the point clouds associated with the lowest EMD values. For the traditional GAN approach, the minimum EMD value is 75.605089, while for the WGAN approach, the minimum EMD value is 147.05 9859.

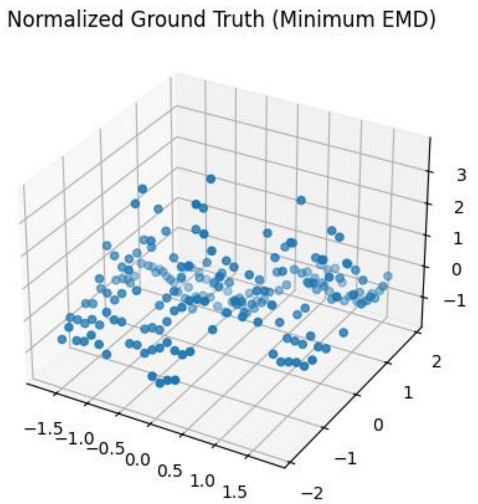


Figure 12: Ground truth point cloud with minimum EMD value for GAN approach

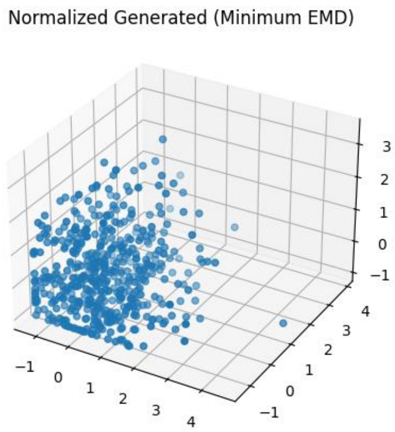


Figure 13: Generated point cloud with minimum EMD value for GAN approach

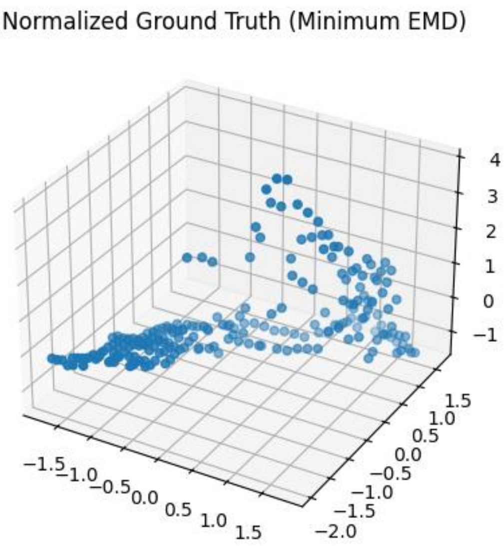


Figure 14: Ground truth point cloud with minimum EMD value for WGAN approach

6.1.2 Fréchet Point Cloud Distance

Fréchet Point Cloud Distance (FPD) metric is taken into consideration as for the evaluation of the point clouds as it enhances the quantitative assessment of the models performance in generating realistic 3D point clouds. The FPD metric captures both the spatial distribution and the structural characteristics of the generated point clouds by considering the mean vector and covariance matrix. The calculation involves processing pairs of ground truth and generated point clouds, extracting their mean vectors and covariance matrices, and subsequently computing the Fréchet Distance using a specific formula that quantifies the dissimilarity between the two distributions. The resulting Fréchet Distance values provide a comprehensive measure of how closely the generated point clouds resemble the ground truth.

For instance, the formula used for FPD incorporates the Euclidean norm of the difference between mean vectors and the trace of matrices involving covariance matrices, reflecting both location and spread differences. This metric, being sensitive to both central tendency and shape, offers a nuanced understanding of the quality of synthetic point clouds generated by the GAN model. Fréchet Point Cloud Distance is mathematically expressed as:

$$FPD(P,Q) = \|\mathbf{m}_P - \mathbf{m}_Q\|_2^2 + \mathrm{Tr}(\mathbf{\Sigma}_P + \mathbf{\Sigma}_Q - 2(\mathbf{\Sigma}_P \mathbf{\Sigma}_Q)^{\frac{1}{2}})$$

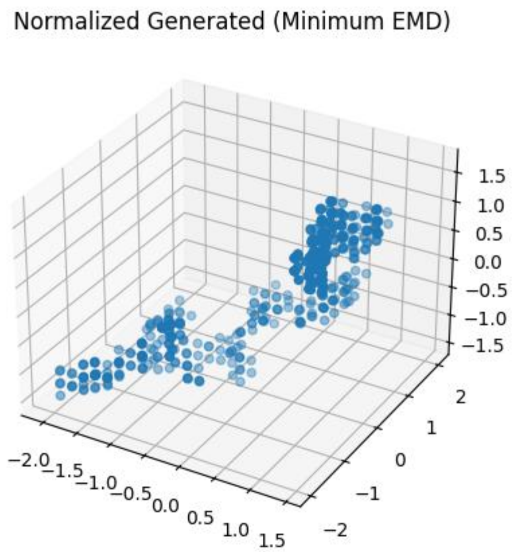


Figure 15: Generated point cloud with minimum EMD value for WGAN approach

where,

$FPD(P,Q)$ is the Fréchet Point Cloud Distance between point clouds P and Q.

$||\mathbf{m}_P - \mathbf{m}_Q||_2^2$is the squared Euclidean distance between mean vectors $\mathbf{m}_P$ and $\mathbf{m}_Q$.

$\mathrm{Tr}(\mathbf{\Sigma}_P + \mathbf{\Sigma}_Q - 2(\mathbf{\Sigma}_P\mathbf{\Sigma}_Q)^{\frac{1}{2}})$ represents the trace of the covariance matrices.

$\mathbf{m}_P$ is the mean vector of point cloud P.

$\mathbf{m}_Q$ is the mean vector of point cloud Q.

$\mathbf{\Sigma}_P$ is the covariance matrix of point cloud P.

$\mathbf{\Sigma}_Q$ is the covariance matrix of point cloud Q.

Figure 16 displays point cloud characterized by the minimum Fréchet Point Cloud Distance (FPD) value achieved through the traditional GAN approach, while Figure 17 exhibits point cloud with the minimum FPD value attained through the WGAN approach, with values of 117030.799274 and 462.595765, respectively.

6.1.3 Mahalanobis Distance

To identify discrepancies between generated and ground truth point clouds, Mahalanobis Distance (MD) is employed. This metric is utilized to detect points in the generated cloud that deviate from the corresponding ground truth by quantifying their dissimilarity. By calculating MD, we assess the extent of each point's deviation from the

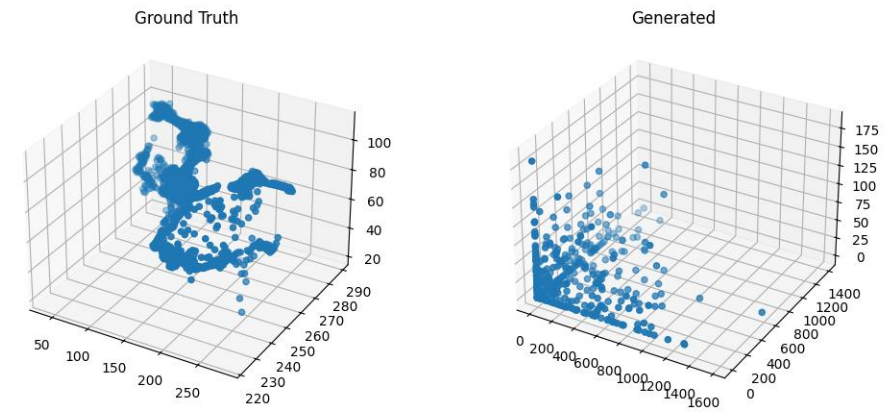


Figure 16: Minimum FPD value for GAN approach

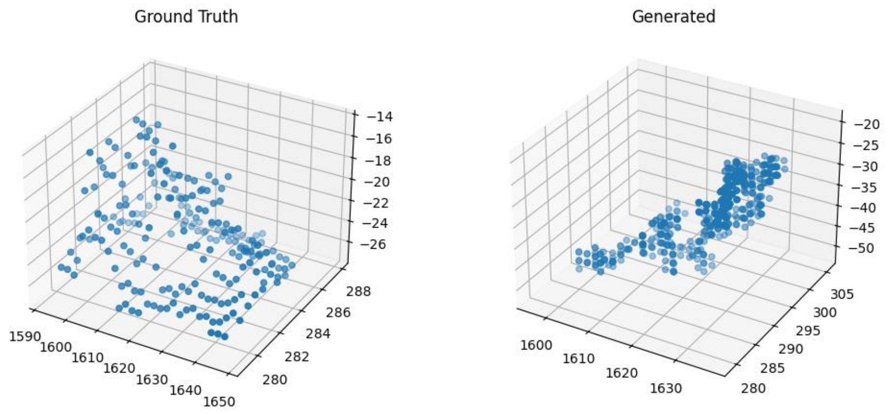


Figure 17: Minimum FPD value for WGAN approach

mean of the distribution, considering the covariance structure. This allows for the identification of anomalous points in the generated cloud that significantly differ from the expected distribution present in the ground truth point cloud. This metric relies on a robust model trained through the Minimum Covariance Determinant (MCD) estimation, which captures the covariance structure of the ground truth point clouds. The MD is computed individually for each point within the generated point clouds. Anomalies, indicative of deviations from the learned distribution, are identified by comparing the calculated Mahalanobis Distance against a predefined threshold.

In this implementation, the threshold value is set at 2.5 for both the approaches. This threshold plays a crucial role in determining what is considered an anomaly. Points with Mahalanobis Distance values exceeding this threshold are flagged as anomalies. The choice of the threshold is a crucial aspect and can be fine-tuned based on the specific characteristics of the dataset and the desired level of sensitivity to anomalies.

A higher threshold may result in fewer anomalies being detected, emphasizing more stringent similarity criteria, while a lower threshold may lead to the identification of a broader range of anomalies. The selection of the threshold value, established at 2.5 for Mahalanobis Distance-based anomaly detection, is the outcome of a comprehensive decision-making process. Grounded in statistical considerations, the threshold denotes a significant departure from the mean, aligning with a 99 percent confidence interval within a normal distribution. This strategic threshold selection significantly identifies the point clouds which are deviated from the ground truth point cloud distribution. Figure 18 and 19 shows the number of anomalies identified with each point cloud generated.

The Mahalanobis Distance (MD) is given by:

$$D_M(x) = \sqrt{(x-\mu)^T \Sigma^{-1} (x-\mu)}$$

Where,

$D_M(x)$ is the Mahalanobis Distance for a point x.
μ is the mean vector.
Σ is the covariance matrix.
$(x-\mu)$ is the difference vector between the point x and the mean μ.
Σ^{-1} is the inverse of the covariance matrix.

Generated File	Anomalies Detected	Total Points
0.pts	374	500
1.pts	281	500
10.pts	445	500
11.pts	333	500
12.pts	362	500
13.pts	321	500
14.pts	324	500
15.pts	290	500
16.pts	316	500
17.pts	323	500
18.pts	479	500
19.pts	269	500
2.pts	394	500
3.pts	347	500
4.pts	386	500
5.pts	451	500
6.pts	440	500
7.pts	335	500
8.pts	372	500
9.pts	397	500

Figure 18: Anomalies detected in each point clouds generated using traditional GAN approach

Generated File	Anomalies Detected	Total Points
0.pts	0	500
1.pts	0	500
10.pts	0	500
11.pts	0	500
12.pts	0	500
13.pts	0	500
14.pts	163	500
15.pts	0	500
16.pts	0	500
17.pts	0	500
18.pts	0	500
19.pts	500	500
2.pts	0	500
3.pts	0	500
4.pts	0	500
5.pts	0	500
6.pts	0	500
7.pts	500	500
8.pts	0	500
9.pts	500	500

Figure 19: Anomalies detected in each point clouds generated using WGAN approach

Mahalanobis Distance and Fréchet Point Cloud Distance are distinct metrics used in different fields with varying applications. Mahalanobis Distance, rooted in statistics and machine learning, quantifies the dissimilarity of a data point from the mean of a distribution, accounting for correlations between variables through the covariance matrix. Commonly used for outlier detection, it offers insights into how far a point deviates from the center of a distribution. On the other hand, Fréchet Point Cloud Distance, a concept in computational geometry, is employed for shape analysis and trajectory comparison. It assesses the similarity between two point clouds by identifying the closest continuous paths in each cloud, emphasizing synchronized movement. This distance is pivotal in applications involving shape matching and similarity analysis of patterns represented as point clouds. Ultimately, the choice between the two distances depends on the specific analytical context and the nature of the data under consideration.

6.2 Discriminator and Generator Loss

The exploration of point cloud generation involves a meticulous analysis of the generator and discriminator loss plots, prominently featured in Figures 20 and 21. These graphical depictions encapsulate the intricate evolution of both traditional Generative Adversarial Network (GAN) and Wasserstein GAN (WGAN) approaches.

Delving into Figure 20, we gain a comprehensive understanding of the nuanced dynamics between the generator and discriminator losses throughout the training process of the conventional GAN. In the early phases of training, both the generator and discriminator losses exhibit a relatively higher magnitude, highlighting the initial learning challenges faced by the model. However, a notable convergence occured around the 400-epoch mark, suggesting a stabilization in the training dynamics. Despite this convergence, the generated point clouds at this stage have not meet the desired quality standards. Recognizing this, the model undergoes continued training beyond the convergence point until satisfactory results in point cloud generation are achieved.

It is imperative to note that the decision to extend training beyond the initial convergence is driven by the aim to enhance the quality of the generated point clouds. Nevertheless, a crucial consideration arises in the form of potential overfitting and mode collapse. To address this, a prudent decision is made to conclude the training at 1000 epochs, mitigating the risk of compromising the model's generalization capabilities.

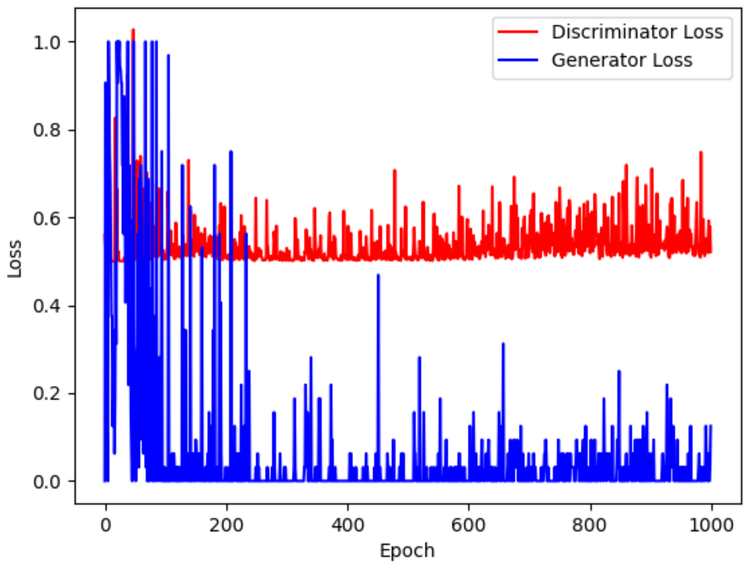


Figure 20: Loss graph of GAN approach

The loss plot of the Wasserstein GAN (WGAN) model, illustrated in Figure 21, reveals a distinctive trend where the discriminator loss displays more pronounced oscillations compared to the generator loss. Similar to the conventional GAN model, the WGAN model undergoes training until it consistently produces high-quality synthetic crack point clouds, precisely up to 5000 epochs. Going beyond this epoch limit leads to overfitting, resulting in a diminishing ability of the model to generate effective point clouds. This emphasizes the significance of determining an optimal training duration to achieve peak performance in point cloud generation.

This finding implies that the generator has successfully generated high-quality point clouds, even though the discriminator is still sharping its ability to distinguish between genuine and synthetic point clouds. In conclusion, the loss plot serves as a valuable tool for assessing the performance of a WGAN model. By incorporating this plot into your results section, you can provide readers with a clearer understanding of the model's training process and its overall performance

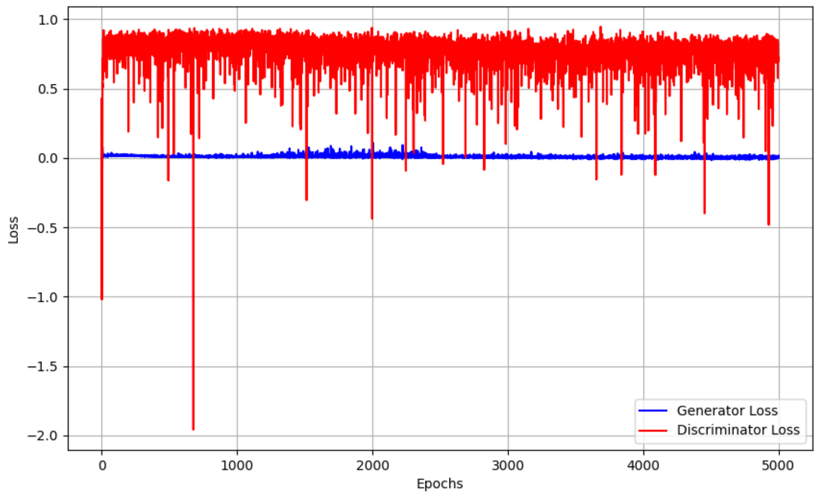


Figure 21: Loss graph of WGAN approach

7 Discussion

This section thoroughly explores the interpretation and detailed analysis of the acquired results, delving into the implications of the findings, comparing both approaches, addressing limitations, and suggesting avenues for future research.

The results of the point cloud generation using both the traditional Generative Adversarial Network (GAN) approach and the Wasserstein Generative Adversarial Network (WGAN) approach provide valuable insights into the feasibility of employing deep learning methodologies for crack detection within aviation maintenance.

7.1 Analysis of the findings

The point clouds produced through the conventional GAN method exhibit a pattern where a notable concentration of points is observed in specific areas, such as the centre or sides of the structure. This starkly deviates from the characteristic structure of genuine crack point clouds. The traditional GAN generated point clouds appear to have a more linear structure when viewed in 2D, and they significantly differ from the intricate structure of actual crack point clouds. On the contrary, the Wasserstein GAN (WGAN) approach demonstrates a substantial enhancement in capturing intricate details from real point clouds, including the spread, structure, and orientation of the points. The point clouds generated by the WGAN approach portray a more authentic representation of crack structures, surpassing the point clouds generated by the traditional GAN method. The results from WGAN are highly promising, closely mirroring the characteristics of genuine crack point clouds.

Analysis of quantitative metrics, the structure of the generated point clouds obtained for Minimum Earth Mover's Distance (EMD) for the GAN approach as mentioned, the points are more gathered towards left as seen in the Figure 13, where as in the ground truth point are well oriented in structure. In case of WGAN, the point form a structural shape can be seen in Figure 15. The EMD analysis reveals that the distributions of the two point clouds ground truth and synthetic point clouds of the WGAN approach exhibit significant similarity in their distributions compared to traditional GAN. Coming to Fréchet Point Cloud Distance, a similar observation to that of the Earth Mover's Distan-
ce can me made from Figures 16 and 17. The point clouds generated from WGAN are more structural and oriented compared to GAN approach. This inference is based on the significantly lower values observed in WGAN, contrasting with the notably higher values in traditional GAN. Additionally, it is noteworthy that the traditional GAN approach exhibits a higher number of outliers compared to the WGAN approach.

The Mahalanobis Distance analysis within the traditional GAN approach reveals a substantial deviation of most points in the generated point clouds from the actual distribution observed in the ground truth point cloud. This discrepancy is vividly illustrated in Figure 18, where the number of anomalies aligns closely with the total number of points, indicating that the structural perspective of the generated point clouds is significantly distant from that of the authentic point clouds.

In contrast, when scrutinizing the anomalies in the point clouds generated by the WGAN model, distinct patterns emerge. The anomalies are either at a count of 0 or in close proximity to 500. This nuanced observation implies that points generated with anomalies detected at 0 closely resemble the distribution of the actual points, whereas point clouds with anomalies closer to 500 are noticeably distant from the structural characteristics exhibited by the authentic point clouds.

In the GAN model's loss plot, illustrated in Figure 20, the initial phase exhibits a higher generator loss, gradually decreasing as training progresses. This trend indicates that the generator is learning to generate more realistic samples over time. Small oscillations in the loss, while common, contribute to the dynamic nature of adversarial training. The WGAN model's loss plot, shown in Figure 21, displays generator loss values closer to 0, signifying effective synthesis of data challenging for the discriminator to differentiate from real data.

In the WGAN context, the generator loss represents the Wasserstein distance, a metric quantifying the difference between the distributions of generated and real data. A generator loss near 0 implies minimized Wasserstein distance, indicating a high-quality generator. Conversely, the discriminator loss in WGAN oscillates throughout training, consistently remaining higher than the generator loss. A higher discriminator loss suggests the generator is adept at producing samples challenging for the discriminator to distinguish as real.

The rhythmic oscillation in the discriminator loss implies stable training, with the generator and discriminator adjusting strategies responsively. This balanced state, where neither component holds a distinct advantage, is desired in GAN training. It signifies effective learning, reaching an equilibrium where both components are optimized.

As a result, the synthetic point clouds generated by WGAN exhibit a higher fidelity in replicating the intricate features of real-world combustion chambers. These findings align with the evolving literature on the application of GANs in addressing data challenges in industries with limited datasets. The utilization of GANs, particularly WGANs, for generating synthetic data has proven effective in mitigating the data deficit prevalent in aviation maintenance. The WGAN approach's success in capturing the complexity of point cloud data is a promising advancement in the realm of crack detection.

7.2 Implications and Challenges

The successful generation of synthetic point clouds using Generative Adversarial Networks (GANs) for crack detection in aviation maintenance carries several promising implications. The ability to create realistic synthetic data capable of mimicking intricate surface features has immediate applications in training and validating crack detection models. This synthetic dataset can act as a valuable supplement to the limited real-world data, facilitating the development and evaluation of robust models for identifying submillimeter cracks within combustion chambers.

Moreover, the utilization of synthetic data addresses the perennial challenge of data scarcity in the aviation industry. The generated point clouds serve as a bridge to overcome the data deficit, providing a diverse set of scenarios for training models. This, in turn, enhances the adaptability and effectiveness of crack detection systems across varying combustion chamber conditions.

Despite the optimistic implications, challenges persist in optimizing the synthetic data generation process. Fine-tuning GAN architectures to precisely capture the subtle distinctions between surface roughness and genuine cracks remains a nuanced task. Striking a balance between sensitivity and specificity is crucial, and further research is needed to enhance the fidelity of synthetic point clouds, ensuring they accurately reflect the complexities of real-world combustion chambers. Interpreting the generated synthetic data also presents challenges in understanding the intricacies of the underlying patterns. Establishing a clear link between synthetic point clouds and real-world crack characteristics is essential for the successful deployment of these models in aviation maintenance practices.

8 Conclusion

In conclusion, implementation of both the traditional Generative Adversarial Network (GAN) approach and the Wasserstein Generative Adversarial Network (WGAN) approach for generating synthetic point clouds has yielded valuable insights in the realm of aviation maintenance like the challenge of detecting submillimeter cracks within combustion chambers, complicated by sensor data intricacies involving fine surface features and roughness, prompted the exploration of advanced neural network architectures and deep learning methodologies.

The primary aim of implementation was to address the prevalent data deficit in aircraft maintenance. Utilizing Generative Adversarial Network (GAN), successfully established a robust pipeline dedicated to generating synthetic training samples. This process included the creation of artificial cracks, where a GAN trained with authentic crack data produced synthetic cracks with diverse patterns. Specifically, the outcomes of implementation highlighted the superior effectiveness of the WGAN approach in closely replicating real-world point clouds compared to the traditional GAN approach. WGAN demonstrated advanced capabilities in efficiently capturing the intricate features of combustion chambers, showcasing its potential to bridge the gap created by data scarcity. The success of WGAN in generating synthetic data that closely mirrors real-world scenarios marks a significant advancement can be correlated to its capacity to offer more stable training dynamics, avoiding typical problems like mode collapse that standard GANs encounter. The architecture of WGAN is one of the reason for the model to successfully generate point clouds with high quality.

In practical terms, implementation contributes to the refinement of data-driven methodologies in safety-critical industries, particularly aviation maintenance. The implementation of both approaches provides a comparative understanding of their efficacy in generating synthetic data for crack detection models. This research outcome holds promise for future applications, offering innovative solutions to enhance aviation maintenance practices through the utilization of generative models and advanced training techniques.

9 Future Research Directions

The trajectory of future research directions in the domain of synthetic point cloud generation for aviation maintenance seeks to propel the field beyond current boundaries. Building upon the advancements achieved, this forward-looking exploration is poised to refine and expand the capabilities of generative models, particularly Generative Adversarial Networks (GANs). The optimization of GAN architectures, specifically tailored for point cloud data, stands as a pivotal objective. By delving into modifications or novel approaches within the GAN framework, the goal is to enhance the generation of intricate surface features, ultimately contributing to the realism and accuracy of synthetic point clouds.

Optimization of GAN Architectures for Point Clouds: Future research should delve into the optimization of GAN architectures specifically tailored for point cloud data. Exploring modifications or novel approaches within the GAN framework to improve the generation of intricate surface features will contribute to the realism and accuracy of synthetic point clouds.

Enrichment of Synthetic Data Variability: Expanding the synthetic dataset to encompass a broader range of combustion chamber configurations and surface conditions is crucial. Future research can focus on enriching the variability within the synthetic data, ensuring that models are trained on a representative set of scenarios to enhance generalization to real-world situations.

Integration with Real-world Data: Further investigations should explore methodologies for effectively integrating synthetic point clouds with existing real-world datasets. This fusion of synthetic and real data can enhance the robustness of crack detection models, enabling them to adapt to diverse conditions and anomalies encountered during aviation maintenance inspections.

In conclusion, future research directions should be centered on refining the synthesis of point cloud data, expanding dataset variability, and validating the efficacy of models in practical scenarios. The ultimate goal is to ensure that synthetic point clouds play a pivotal role in enhancing the capabilities of crack detection models within aviation maintenance, addressing the unique challenges posed by limited real-world data.

References

[1] ArcGIS: *Point Clouds*. https://doc.arcgis.com/en/3d/workflows/content/import-point-clouds.htm#:~:text=Point%20cloud%20data%20is%20a,%2C%20roads%2C%20and%20other%20feature. Version: May 17, 2019

[2] Armeni, Iro ; Sener, Ozan ; Zamir, Amir R. ; Jiang, Helen ; Brilakis, Ioannis ; Fischer, Martin ; Savarese, Silvio: 3d semantic parsing of large-scale indoor spaces. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, S. 1534–1543

[3] Bello, Saifullahi A. ; Yu, Shangshu ; Wang, Cheng ; Adam, Jibril M. ; Li, Jonathan: Review: Deep Learning on 3D Point Clouds. In: *Remote Sensing* 12 (2020), Nr. 11. http://dx.doi.org/10.3390/rs12111729. – DOI 10.3390/rs12111729. – ISSN 2072–4292

[4] Brownlee, Jason: *A Gentle Introduction to Generative Adversarial Networks (GANs)*. https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/. Version: July 19, 2019

[5] Charles R. Qi, Kaichun Mo-Leonidas J. G. Hao Su S. Hao Su: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In: *arXiv 2020* https://doi.org/10.48550/arXiv.1612.00593

[6] Chen, B.; Feng C.; Vallespi-Gonzalez C.; Wellington C. S.; Liu L. S.; Liu: A3D Point Cloud Processing and Learning for Autonomous Driving. In: *arXiv 2020* arXiv:2003.00601

[7] Chen, Haiyang: Challenges and Corresponding Solutions of Generative Adversarial Networks (GANs): A Survey Study. In: *Phys.: Conf. Ser. 1827 012066* (2021)

[8] Creswell, Antonia ; White, Tom ; Dumoulin, Vincent ; Arulkumaran, Kai ; Sengupta, Biswa ; Bharath, Anil A.: Generative Adversarial Networks: An Overview. In: *IEEE Signal Processing Magazine* 35 (2018), Nr. 1, S. 53–65. http://dx.doi.org/10.1109/MSP.2017.2765202. – DOI 10.1109/MSP.2017.2765202

[9] Dai, Angela ; Ruizhongtai Qi, Charles ; Niessner, Matthias: Shape completion using 3d-encoder-predictor cnns and shape synthesis. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, S. 5868–5877

[10] Dimitrov, Andrey ; Golparvar-Fard, Mani: Segmentation of building point cloud models including detailed architectural/structural features and MEP systems. In: *Automation in Construction* 51 (2015), 03. http://dx.doi.org/10.1016/j.autcon.2014.12.015. – DOI 10.1016/j.autcon.2014.12.015

[11] Divya Saxena, Jiannong C.: Generative Adversarial Networks: Challenges,Solutions, and Future Directions.

[12] DWIVEDI, Harshit: *Understanding GAN Loss Functions*. https://neptune.ai/blog/gan-loss-functions. Version: 29th August, 2023

[13] FABIO, R: From point cloud to surface: The modeling and visualization problem. In: *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci* 34 (2003), S. W10

[14] FAN, Haoqiang ; SU, Hao ; GUIBAS, Leonidas: A Point Set Generation Network for 3D Object Reconstruction from a Single Image. (2016), 12

[15] GIGABYTE: *Point Cloud*. https://www.gigabyte.com/Glossary/point-cloud

[16] GONFALONIERI, Alexandre: *A Dealing with the Lack of Data in Machine Learning*. https://medium.com/predict/dealing-with-the-lack-of-data-in-machine-learning-725f2abd2b92. Version: May 17, 2019

[17] GOODFELLOW, Ian ; POUGET-ABADIE, Jean ; MIRZA, Mehdi ; XU, Bing ; WARDE-FARLEY, David ; OZAIR, Sherjil ; COURVILLE, Aaron ; BENGIO, Yoshua: Generative Adversarial Networks. In: *Commun. ACM* 63 (2020), oct, Nr. 11, 139–144. http://dx.doi.org/10.1145/3422622. – DOI 10.1145/3422622. – ISSN 0001–0782

[18] GUO, Yulan ; SOHEL, Ferdous ; BENNAMOUN, Mohammed ; WAN, Jianwei ; LU, Min: A novel local surface feature for 3D object recognition under clutter and occlusion. In: *Information Sciences* 293 (2015), 196-213. http://dx.doi.org/https://doi.org/10.1016/j.ins.2014.09.015. – DOI https://doi.org/10.1016/j.ins.2014.09.015. – ISSN 0020–0255

[19] HUI, Jonathan: *GAN — Wasserstein GAN WGAN-GP*. https://jonathan-hui.medium.com/gan-wasserstein-gan-wgan-gp-6a1a2aa1b490. Version: Jun 14, 2018

[20] KHARROUBI, Abderrazzaq ; POUX, Florent ; BALLOUCH, Zouhair ; HAJJI, Rafika ; BILLEN, Roland: Three Dimensional Change Detection Using Point Clouds: A Review. In: *Geomatics* 2 (2022), 10, S. 457–486. http://dx.doi.org/10.3390/geomatics2040025. – DOI 10.3390/geomatics2040025

[21] LE, Robert ; NGUYEN, Minh ; YAN, Weiqi: Machine Learning with Synthetic Data – a New Way to Learn and Classify the Pictorial Augmented Reality Markers in Real-Time, 2020, S. 1–6

[22] LI, Chun-Liang ; ZAHEER, Manzil ; ZHANG, Yang ; POCZOS, Barnabas ; SALAKHUTDINOV, Ruslan: Point cloud gan. In: *arXiv preprint arXiv:1810.05795* (2018)

[23] LI, Yangyan ; BU, Rui ; SUN, Mingchao ; WU, Wei ; DI, Xinhan ; CHEN, Baoquan: PointCNN: Convolution On X-Transformed Points. In: BENGIO, S. (Hrsg.) ; WALLACH, H. (Hrsg.) ; LAROCHELLE, H. (Hrsg.) ; GRAUMAN, K. (Hrsg.) ; CESA-BIANCHI, N. (Hrsg.) ; GARNETT, R. (Hrsg.): *Advances in Neural Information Processing Systems* Bd. 31, Curran Associates, Inc., 2018

[24] MARTIN ARJOVSKY, Léon B. Soumith Chintala C. Soumith Chintala: Wasserstein GAN. http://dx.doi.org/https://arxiv.org/abs/1701.07875. – DOI https://arxiv.org/abs/1701.07875

[25] NURUNNABI, Abdul ; WEST, Geoff ; BELTON, David: Outlier detection and robust normal-curvature estimation in mobile laser scanning 3D point cloud data. In: *Pattern Recognition* 48 (2015), Nr. 4, 1404-1419. http://dx.doi.org/https://doi.org/10.1016/j.patcog.2014.10.014. – DOI https://doi.org/10.1016/j.patcog.2014.10.014. – ISSN 0031–3203

[26] PARK, D.; Ku M.; Jung-I.; Jeong C. J.; Seo S. J.; Seo: Multiple 3D Object Tracking using ROI and Double Filtering for Augmented Reality. In: *In Proceedings of the 2011 Fifth FTRA International Conference on Multimedia and Ubiquitous Engineering, Loutraki, Greece* (2011), S. 317–322

[27] PLANCHE, Benjamin ; WU, Ziyan ; MA, Kai ; SUN, Shanhui ; KLUCKNER, Stefan ; CHEN, Terrence ; HUTTER, Andreas ; ZAKHAROV, Sergey ; KOSCH, Harald ; ERNST, Jan: *DepthSynth: Real-Time Realistic Synthetic Data Generation from CAD Models for 2.5D Recognition*. 2017

[28] POMERLEAU, F.; Siegwart R. F.; Colas C. F.; Colas: A Review of Point Cloud Registration Algorithms for Mobile Robotics. Found. In: *Trends Robot* 4 (2015), S. 1–104

[29] POMERLEAU, François ; COLAS, Francis ; SIEGWART, Roland: A Review of Point Cloud Registration Algorithms for Mobile Robotics. In: *Foundations and Trends® in Robotics* 4 (2015), Nr. 1, 1-104. http://dx.doi.org/10.1561/2300000035. – DOI 10.1561/2300000035. – ISSN 1935–8253

[30] QI, Charles R. ; SU, Hao ; MO, Kaichun ; GUIBAS, Leonidas J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, S. 652–660

[31] QI, Charles R. ; YI, Li ; SU, Hao ; GUIBAS, Leonidas J.: PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In: GUYON, I. (Hrsg.) ; LUXBURG, U. V. (Hrsg.) ; BENGIO, S. (Hrsg.) ; WALLACH, H. (Hrsg.) ; FERGUS, R. (Hrsg.) ; VISHWANATHAN, S. (Hrsg.) ; GARNETT, R. (Hrsg.): *Advances in Neural Information Processing Systems* Bd. 30, Curran Associates, Inc., 2017

[32] RADFORD, Alec ; METZ, Luke ; CHINTALA, Soumith: *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. 2015

[33] RIZZOLI, Alberto: *An Introductory Guide to Quality Training Data for Machine Learning*. https://www.v7labs.com/blog/quality-training-data-for-machine-learning-guide. Version: July 11, 2022

[34] SAMEERA RAMASINGHE1, Nick B. Salman Khan K. Salman Khan ; GOULD, Stephen: Spectral-GANs for High-Resolution 3D Point-cloud Generation. In: *Some journal* 21 (2013), S. 498–516. http://dx.doi.org/10.xxxx. – DOI 10.xxxx

[35] SCHWARZ, Brent: Mapping the world in 3D. In: *Nat. Photonics* 4 (2010), S. 428–430

[36] SHU, Dong W. ; PARK, Sung W. ; KWON, Junseok: 3D Point Cloud Generative Adversarial Network Based on Tree Structured Graph Convolutions. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019

[37] TANG, D.; Akinci B.; Lipman-R.; Lytle A. P.; Huber H. P.; Huber: Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. In: *Autom. Constr.* 19 (2010), S. 829–843

[38] WANG, Y.K.; Kim C. C.; Cho C. C.; Cho: Automatic BIM component extraction from point clouds of existing buildings for sustainability applications. In: *Autom. Constr* 56 (2015), S. 1–13

[39] WANG, Zhengwei ; SHE, Qi ; WARD, Tomas E.: *Generative Adversarial Networks in Computer Vision: A Survey and Taxonomy*. 2020

[40] WENG, Lilian: *From GAN to WGAN*. 2019

[41] WHITTY, Mark ; COSSELL, Stephen ; DANG, Kim S. ; GUIVANT, Jose ; KATUPITIYA, Jayantha: Autonomous navigation using a real-time 3d point cloud. In: *2010 Australasian Conference on Robotics and Automation*, 2010, S. 1–3

[42] YANG, Ximing ; WU, Yuan ; ZHANG, Kaiyi ; JIN, Cheng: CPCGAN: A controllable 3D point cloud generative adversarial network with semantic label generating. In: *Proceedings of the AAAI Conference on Artificial Intelligence* Bd. 35, 2021, S. 3154–3162

[43] ZHU, Qiuchen ; DINH, Tran H. ; PHUNG, Manh D. ; HA, Quang: Hierarchical Convolutional Neural Network with Feature Preservation and Autotuned Thresholding for Crack Detection. (2021), 04

Declaration

I hereby certify that I have written this thesis independently and that I have not used any sources or aids other than those indicated, that all passages of the work which have been taken over verbatim or in spirit from other sources from other sources have been marked as such and that the work has not yet been has not yet been submitted to any examination authority in the same or a similar form.

Y. Nishitha

Erlangen, January 22, 2024 my signature