



Master's Thesis

Xiangyu Lu

Fiber Break Prediction Using 3D Generative Models

Supervised by Abraham George Smith - University of Copenhagen, Denmark

External supervision by Mahoor Mehdikhani - KU Leuven, Belgium

Data and support provided by the Composite Materials Group at KU Leuven, Belgium

Submitted on: 31/05/2024

Contents

Abstract	2
1 Introduction	2
2 Background	3
2.1 Convolutional neural networks: U-Net	3
2.2 Generative models: GAN	4
2.3 Fiber-reinforced composites and fiber breaks	4
2.4 Related works on FCR microstructure analysis	5
2.5 Aim of the study	6
3 Dataset	6
4 Methods	8
4.1 3D U-Net implementation	8
4.2 Conditional GAN implementation	10
5 Experiments	13
5.1 Data pre-processing	13
5.1.1 Cropping	13
5.1.2 Registration	13
5.1.3 Dataset Split	15
5.2 Training settings	15
5.2.1 Loss function	15
5.2.2 Early stopping	16
6 Results	16
7 Discussion	22
7.1 MAE vs MSE	23
7.2 Impact of GAN architecture	24
7.3 Variability in model fitting on different datasets	25
7.4 Future works	25
8 Conclusion	26
Acknowledgements	27
References	28

Fiber Break Prediction Using 3D Generative Models

Xiangyu Lu¹

¹MSc in IT and Cognition, University of Copenhagen
xilu@di.ku.dk

Abstract

Reliable failure predictions of fiber-reinforced composites (FRCs) are crucial for ensuring the safety of products, reducing costs, and optimizing performance. Investigating the underlying mechanism of tensile failure of individual fibers is an important research direction. In this thesis, we aim to predict potential fiber breaks by synthesizing computed tomography (CT) images of FRCs at higher forces based on CT scans taken under initial force. We developed and evaluated 3D conditional generative adversarial networks (c-GANs) with 3D U-Nets as generators to generate CT images of fiber structures under increased force. Our models obtained MAE and MSE scores of 0.0369 and 0.00278, respectively, on the validation set. The visual quality of the generated images indicates that the models excelled at predicting the movement of the fiber structure under increasing force. However, we observed a weak correlation between the generated and ground truth fiber breaks.

1 Introduction

In the field of structural engineering, fiber breaks within composite materials assumes critical importance as it profoundly influences the reliability and performance of such materials [1]. Particularly under tensile loading conditions, individual fibers embedded within composite structures are susceptible to fracturing, thereby posing substantial risks to the overall structural integrity [2]. The understanding of the mechanics governing fiber breaks is paramount for optimizing the design of composite materials and mitigating the occurrence of catastrophic failures, especially in safety-critical industries like aerospace [3] and automotive engineering [4]. Researchers and engineers endeavor to develop comprehensive models and simulation techniques [5, 6, 7] to elucidate the intricate processes associated with fiber breaks, aiming to inform the design of more resilient and durable composite materials.

Current methods for predicting fiber breaks often rely on empirical models [8, 9] that may not capture the complex interactions within composite materials. With the advancement of computational techniques, generative models have demonstrated their remarkable ability to learn the underlying distribution of data and produce highly realistic instances [10, 11, 12]. This research is motivated by the need for a more comprehensive understanding of fiber break mechanisms. Our objective is to use generative models to learn and predict the structural characteristics of fiber material, including potential fiber breaks. A proficient generative model capable of accomplishing these tasks should possess a thorough understanding of both fiber structure and fiber break mechanisms. Through this approach, we seek to enhance our understanding of how the microstructures of fiber material influence the occurrence of fiber breaks.

Since most of fiber breaks in fiber-reinforced composites (FRCs) occur under high tension [13], our study utilize generative models to predict the fiber breaks of FRCs under increased tensile loading. By employing CT scans of FRCs under initial tension, we aim to generate CT images that represent

the composite structure under higher tension. We developed an end-to-end framework using 3D conditional generative adversarial networks (c-GANs) with 3D U-Nets as generators. We trained our models on two datasets of varying difficulty and qualitatively and quantitatively analyzed the quality of the generated images. Additionally, separate 3D U-Net models were trained without the generative model architecture to explore the impact of c-GAN’s adversarial training on generation performance.

The contributions of this thesis can be summarized as follows:

1. We develop and evaluate an end-to-end framework based on c-GAN to predict FRC CT images under increased tension from corresponding images under initial force. The visual quality of the generated images demonstrates that the framework can accurately predict the movement of the fiber structure as tensile loading increases.
2. We customize a 3D U-Net as the generator for the c-GAN, leveraging its skip connections to integrate the same underlying features between FRC images under high and low forces. Additionally, this 3D U-Net model can be independently used for FRC CT image synthesis.
3. We utilize MAE and MSE as similarity loss for training the generator, which considers the differences between the generated images and real FRC CT images under high tension (ground truth), thereby enhancing the similarity of the generated FRC CT images to the ground truth.
4. Our experiments show that the incorporation of the c-GAN architecture can improve both the qualitative and quantitative performance of standalone 3D U-Nets, and it can also enhance the stability of the training process.

2 Background

2.1 Convolutional neural networks: U-Net

Convolutional Neural Networks (CNNs) are a class of deep learning models specifically designed for processing structured grid data, such as images. CNNs use a variation of multilayer perceptrons and are particularly effective in tasks involving image and video recognition, classification, segmentation, and other areas where spatial hierarchies of features need to be learned [14, 15].

Among the various CNN architectures, U-Net [16] stands out as a specialized model designed for biomedical image segmentation. The U-Net owes its name to its U-shaped structure, which consists of a contracting path to capture context and a symmetric expanding path to enable precise localization. Importantly, U-Net’s skip connections bridge the encoder and decoder layers, combining coarse and fine features to enhance the segmentation accuracy. This helps in retaining the localization information that might otherwise be lost in the down-sampling process [17]. The symmetric nature of U-Net allows for effective information flow from the high-resolution feature maps of the encoder to the decoder, ensuring detailed reconstruction of the output [18].

Building on the success of the 2D U-Net, the 3D U-Net [19] modifies the original architecture to process 3D volumes, thereby providing a solution for segmenting 3D biomedical images such as MRI [20] and CT scans [21]. Due to the three-dimensional nature of 3D data, such adaptation is necessary, where feature information can be distributed across multiple consecutive slices and requires an understanding of context along all three axes [22].

Although initially designed for biomedical image segmentation, U-Net and its variants have demonstrated their effectiveness in material science as well. For example, in metallography, U-Net has been used to identify grain boundaries and phases in microstructure images [23], aiding in the characterization of materials’ mechanical properties and performance. Additionally, in semiconductor manufacturing, U-Net variants have been applied to detect defects and anomalies in integrated circuit images [24], ensuring the quality and reliability of electronic devices.

In this thesis, we select 3D U-Net for synthesizing fiber structure images under increased force due to several reasons. Firstly, the U-Net’s contracting path and expanding path may facilitate the model’s step-by-step understanding of fiber structure features and their synthesis into corresponding images. Secondly, CT images of fiber-reinforced composites under different forces belong to the same modality, sharing a substantial amount of underlying structural information. This makes the U-Net

architecture, with its skip connections, particularly suitable for our generation task. Lastly, since our datasets consist of 3D images, the 3D version of U-Net can better capture the contextual information across multiple consecutive slices, leading to more accurate image synthesis.

2.2 Generative models: GAN

Generative models encompass a variety of models commonly used in machine learning. The purpose of generative models is to learn the generation of data given some training data [25]. A good generative model should be able to produce diverse outputs resembling the training data, rather than exact replicas of the training data.

Generative Adversarial Network (GAN) [26] is one of the most widely used generative models, renowned for its exceptional ability to create highly realistic synthetic data. GANs consist of two neural networks, the generator (G) and the discriminator (D), which are trained simultaneously through adversarial processes. The generator’s role is to create data that is indistinguishable from real data using random noise, while the discriminator aims to differentiate between real and generated data. Through this adversarial training process, both networks continuously improve, each striving to outwit the other. This competitive learning dynamic drives the generator to create increasingly realistic outputs, while the discriminator becomes adept at distinguishing real from fake data.

As the training progresses, the generator learns to capture the underlying data distribution [27], enabling it to generate novel, high-quality samples that exhibit the characteristics of the training data. In recent studies, GANs have demonstrated remarkable capabilities in various domains, including image generation [10, 11], style transfer [28], image-to-image translation [29], and even text and music generation [30, 31].

This broad range of applications is attributed not only to GANs’ exceptional ability to generate diverse and realistic outputs through adversarial training but also to their remarkable architectural flexibility. In GANs, the specific architectures of the generator and discriminator can be customized to address different tasks [32, 33, 34]. Additionally, the number of generators and discriminators can be adjusted to conduct more complex adversarial training [35, 36]. This flexibility enables GANs to integrate seamlessly with various cutting-edge models, giving them robust adaptability.

Moreover, GANs have inspired numerous advancements and extensions, such as conditional GANs (c-GAN) [37], where both the generator and discriminator are conditioned on additional information, enabling controlled generation based on specific attributes or classes. Other variants include Wasserstein GANs [38], which leverage Wasserstein distance to improve training stability and sample quality, and progressive growing GANs, which gradually increase the resolution of generated images during training, yielding high-fidelity outputs.

In this thesis, we choose the c-GAN architecture for our generative model for several reasons. Firstly, the adversarial training process between the generator and discriminator in a GAN structure may help produce high-quality images. Secondly, because we aim to generate corresponding images under increased force from CT images of fiber-reinforced composites obtained under initial force, the conditional mechanism of c-GANs makes them highly suitable for our task. Lastly, the excellent flexibility of the GAN architecture allows it to integrate well with the 3D U-Net.

2.3 Fiber-reinforced composites and fiber breaks

Fiber-reinforced composites (FRCs) are a class of composite materials formed by embedding fiber materials (such as glass fiber, carbon fiber, aramid fiber, etc.) into a matrix material (typically a polymer, metal, or ceramic matrix) [39]. FRCs have emerged as indispensable materials, especially in industries prioritizing weight reduction. Renowned for their exceptional strength-to-weight ratios, fiber-reinforced composites have reshaped design possibilities and operational efficiency across many industries [40]. In the aerospace industry, for instance, fiber-reinforced composites have become integral to the construction of aircraft components, such as wings, fuselages, and tail sections. Their lightweight nature not only contributes to fuel efficiency but also enhances overall aircraft performance, enabling greater payloads and extended flight ranges [41]. Similarly, in the automotive sector, electric cars increasingly rely on fiber-reinforced composites for structural components like chassis, body panels, and battery enclosures. By integrating these advanced materials, automakers

can achieve significant reductions in vehicle weight, thereby extending driving range and enhancing energy efficiency [42].

However, the reliability and performance of FRCs depend on a thorough understanding of the intricate mechanics governing fiber breaks under tensile loading. This aspect holds paramount importance in ensuring the structural integrity and durability of composite materials, as the failure of individual fibers can have profound implications for the overall performance of the composite structure [2].

Tensile loading, or tension, is a common mechanical loading condition experienced by fiber-reinforced composites in various applications. Under tension, the fibers within the composite are subjected to forces that attempt to pull them apart along their length. The behavior of these fibers under such loading conditions is influenced by factors such as fiber orientation [43], material properties [44], and the presence of defects or imperfections [41]. Under the strain of such force, individual fibers within the composite gradually succumb to breaks, posing a significant risk to the structural integrity of the material. This process of fiber failure can propagate rapidly, particularly under high tensile loads [45], leading to the development of cracks and eventual catastrophic failure of the composite structure.

Generally speaking, the load at which each fiber breaks is contingent upon its inherent strength and the surrounding microstructure, which includes factors such as fiber alignment and proximity to neighboring fibers [46]. Despite advancements in characterization techniques such as microscopy, computed tomography, and computational modeling, our ability to accurately quantify and predict the impact of microstructural changes on fiber failure remains limited [47]. The complex interactions between microstructural features and fiber failure mechanisms continue to be a subject of ongoing research and investigation.

2.4 Related works on FCR microstructure analysis

There have been numerous studies investigating the inherent strength of fiber-reinforced composite materials and the impact of their microstructure on failure under loading. In general, the fibers within a bundles follow a Weibull distribution instead of possessing uniform strength [48]. Breite et al. [46] conducted detailed experimental validation of potential mechanisms for the tensile failure of single fibers in two types of carbon fiber/epoxy composites using six state-of-the-art models. All models overestimated the observed fiber break density in the experiments and failed to accurately predict the development of fiber break (and cluster). This indicates the importance of improving experimental methods to accurately characterize the in-situ strength distribution of fibers within composite materials.

Bhaduri et al.[49] employed a U-Net architecture to predict the von Mises stress field under uniaxial tension and fine-tuned it on data from relatively cheaper systems with smaller fiber numbers. The limitation is that the systems with a large number of fibers typically require finer finite element mesh discretization, leading to increased computational costs. Gupta et al.[50] also employed a U-Net deep convolutional neural network to explore how the microstructural heterogeneity in fiber-reinforced composites influences the material’s macroscopic behavior. The models were trained separately to simulate the mapping between the spatial arrangement of fibers and the stress tensor field. However, this mapping is limited to 2D fields, and further research is needed to investigate its applicability to 3D scenarios.

Zhou et al.[51] developed a customized convolutional neural network (CNN) and a VGG16 transfer learning neural network to explore microstructure uncertainties. This framework is generalizable and exhibits excellent scalability. However, its drawback lies in the potential for overfitting when training data volumes are small, which may render the CNN unreliable. Yang et al.[52] proposed an end-to-end deep learning method to predict complete strain and stress tensors for complex composite microstructures. In this method, a conditional generative adversarial network is utilized to predict physical fields from geometry, achieving high accuracy. However, the authors also noted that the model still needs to learn from more general probability distributions to better fit real scenarios.

2.5 Aim of the study

In this study, our objective is to leverage the generative models' capability of learning the underlying distribution of data, enhancing our understanding of how the microstructure of fiber material influences the occurrence of fiber breaks. Therefore, we aim to implement a generative model capable of synthesizing realistic FRC CT images under increased tension from the corresponding images under initial tension. By training the model to learn how increasing tensile loading affects fiber structure and fiber breaks, we hope to observe potential fiber structure movements and fiber breaks in the generated images, even if they have not yet appeared in the input images.

Furthermore, to the best of our knowledge, there is no existing model capable of predicting potential fiber structure movements and fiber breaks by generating FRC CT images. We aim for our models to fill this gap.

3 Dataset

The dataset used in this paper is sourced from [53]. It comprises 17 8-bit RAW files, which represent continuous synchrotron X-ray computed tomography (CT) scans of the same carbon fiber-reinforced composites under varying forces. The forces range from 148 MPa to the point of fiber failure. Each file can be read as a 3D image of varying sizes (see in Table 1), with voxels fine enough to identify individual fibers and any potential fiber breaks therein (see in Figure 1).

Image Number	Length	Width	Height	Stress (in MPa)	No. fiber breaks
00	1737	513	1716	148	0
01	1766	487	1716	530	0
02	1771	487	1716	819	4
03	1771	490	1716	1112	5
04	1813	488	1716	1386	30
05	1789	483	1716	1623	65
06	1819	483	1716	1813	118
07	1838	482	1716	1919	153
08	1792	498	1716	2002	193
09	1775	487	1716	2055	221
10	1775	494	1716	2062	226
11	1785	481	1716	2015	229
12	1775	492	1716	2044	235
13	1781	471	1716	2091	263
14	1777	487	1716	2112	275
15	1775	483	1716	2130	286
16	1766	478	1716	2166	-

Table 1. The size of each image in the original dataset, the number of fiber breaks contained within them, and the corresponding stress levels (in MPa) at the time of their capture. The absence of fiber break quantity for the last image is due to the occurrence of fiber failure.

In these 3D images, the orientation of fibers generally parallels the z-axis, albeit with potential deviations. Through manual observation, the coordinates of fiber voxels across the 17 3D images are inconsistent, exhibiting varying degrees of displacement. Moreover, as the force increases, these displacements tend to become more pronounced (see in Figure 2).

As the force increases, fiber breaks gradually become observable in these 3D images. These breaks typically manifest as small black spots appearing on individual fibers, indicating a break at that point. Fiber breaks are irreversible, and their quantity accumulates continuously as the material is loaded in tension (refer to the "No. fiber breaks" column in Table 1). When the accumulation of fiber breaks reaches a certain threshold, the composites will experience failure [54] (as indicated in the final row of Table 1).

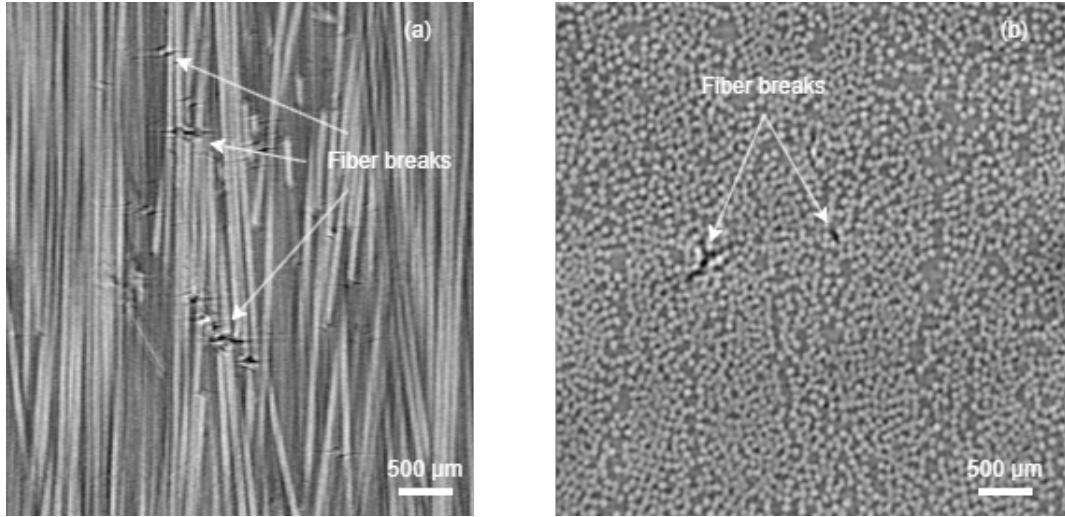


Figure 1. A vertical slice (a) and a horizontal slice (b) from image 14, where several fiber breaks can be observed. In the vertical slice (a), fiber breaks typically appear as small black lines perpendicular to the fiber direction, indicating that the fibers or fiber cluster have broken at these points. In the horizontal slice (b), fiber breaks appear as black dots, resulting from the absence of fiber material cross-sections (depicted as small white dots in the image) due to the breaks.

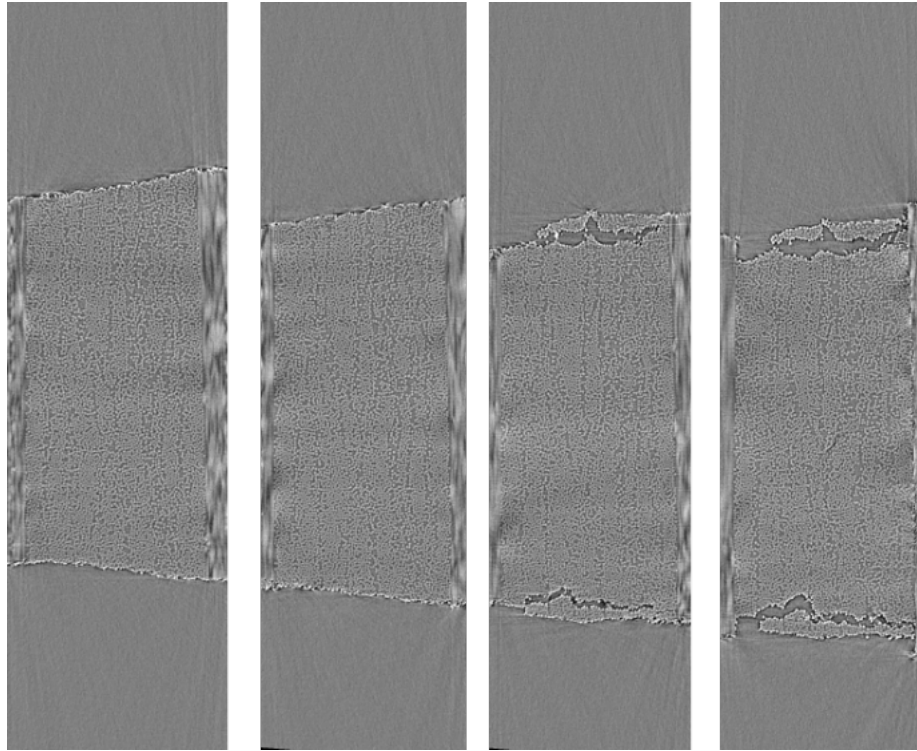


Figure 2. The images above, from left to right, depict the central Z-axis cross-sections of image 00, 02, 05, and 16. As the force increases, the relative positions of fibers within the images undergo displacement, with this displacement becoming more pronounced as the force increases. Meanwhile, ply splits have emerged at the edges of the fiber-reinforced composites, resulting in cracks between the materials.

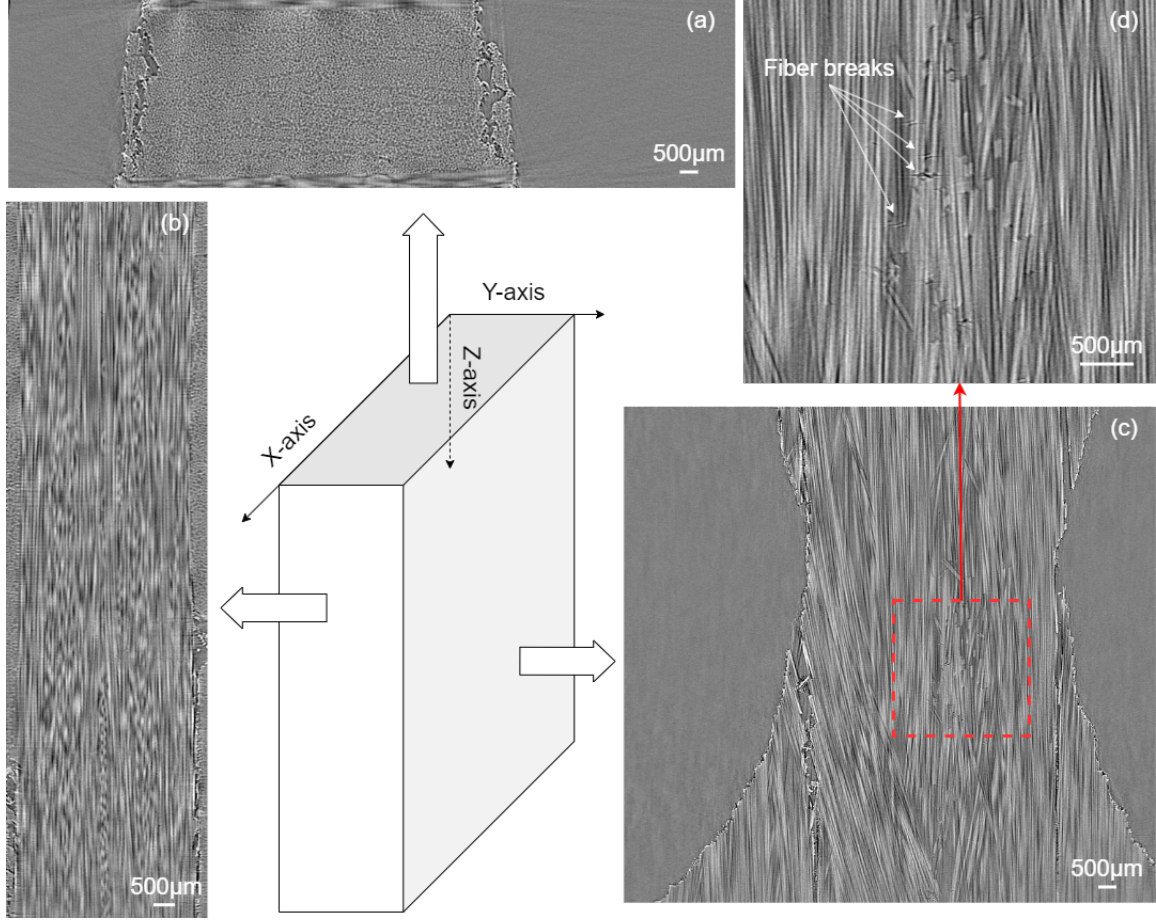


Figure 3. The central cross-sections of image 07 along each axis: the X-axis cross-section (b), the Y-axis cross-section (c), and the Z-axis cross-section (a). In the Y-axis cross-section, partial fiber breaks can be observed (d), manifesting as small black spots appearing on individual fibers.

4 Methods

In this thesis, we develop an end-to-end framework based on 3D conditional generative adversarial networks (c-GANs) to generate CT images of fiber-reinforced composites (FRCs) under high tensile loading from corresponding CT images under initial tension. Specifically, we condition the GAN with FRC CT images under initial force, incorporating these conditions into both the generator and discriminator during training to enhance the similarity of the generated images to the ground truth. To improve the model’s understanding of fiber structure, we develop a 3D U-Net as the generator in our c-GAN, utilizing its skip connections to integrate multilevel features of the FRC CT images. This section will detail the implementation of our 3D U-Net and c-GAN.

4.1 3D U-Net implementation

The 3D U-Net architecture we implement is inspired by [55] and modified accordingly. Given the complexity of FRC structures, we aim to enhance the model’s ability to extract multilevel features by deepening the 3D U-Net. Therefore, we deepen both the downsampling and upsampling paths, increasing from 3 to 4 basic convolutional or deconvolutional modules (see in Figure 4). Regarding skip connections, as the network structure deepens, the number of skip connections increases from 3 to 4. Additionally, we add two convolutional blocks as the U-Net bottleneck, serving as a bridge between the end of the encoder and the start of the decoder, where the two convolutional blocks will not change the shape of the feature maps. Based on the research in [19], models trained with batch normalization [56] demonstrate better training performance and stability. Therefore, we retain batch

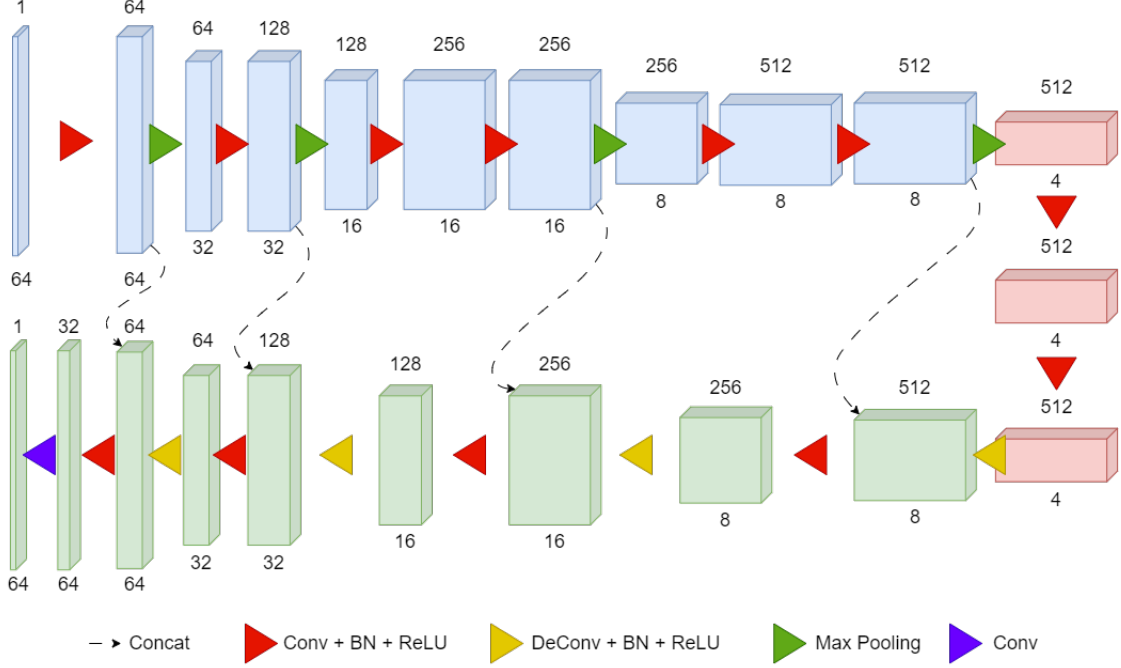


Figure 4. The architecture of the 3D U-Net utilized in this thesis. It takes a single-channel $64 \times 64 \times 64$ 3D image as input and outputs an image of the same shape. Each cuboid represents the input or output of a layer or block in the model. The blue cuboids constitute the downsampling pathway, the green cuboids form the upsampling pathway, and the red cuboids serve as bridges (bottleneck) connecting them. For each cuboid, the number above indicates the number of channels in the corresponding image or feature map, while the number below represents the spatial size of the image or feature map. For instance, the centrally positioned red cuboid indicates a feature map with 512 channels and a shape of $4 \times 4 \times 4$. Throughout the model, there are four skip connections (denoted by dashed arrows) linking the upsampling and downsampling pathways. The feature maps at the ends of the skip connections are concatenated along the channel dimension and then fed into subsequent layers.

normalization layers after each convolutional and deconvolutional layer, which will normalize each mini-batch data, i.e., rescaling and shifting them to meet a standard distribution with a mean of 0 and variance of 1.

There are totally 12 basic convolutional blocks and 4 basic deconvolutional blocks in our 3D U-net, with consistent parameter settings for each block. Among these, the first 6 convolutional blocks form the encoder, progressively downsampling the input image to smaller feature maps. Unlike [19], we preserved the bottleneck structure, consisting of the 5th and 6th convolutional blocks, to capture contextual information from the input data and pass it to subsequent modules for upsampling. The 4 deconvolutional blocks and the last 4 convolutional blocks of the model collectively form the decoder, integrating feature maps from both the bottleneck and the encoder to map semantic information back to the original dimensions. Additionally, max-pooling layers with a kernel size of 2 and a stride of 2 are inserted after the first, second, fourth, and sixth convolutional blocks to decrease the spatial resolution of the feature maps. Finally, an additional 3D convolutional layer with a kernel size of 1 and a stride of 1 is appended after the last decoder to restore the original image size.

Convolutional blocks A basic convolutional block consists of a sequence of layers: a 3D convolutional layer with a kernel size of 3 and a stride of 2, followed by a 3D batch normalization layer, and then a ReLU activation layer. The padding of the 3D convolutional layer is set to 1, ensuring that the convolutional block does not change the input size. The output channels of the block are determined by the 3D convolutional layer.

Deconvolutional blocks A basic deconvolutional block consists of a sequence of layers: a 3D transposed convolutional layer with a kernel size of 4 and a stride of 2, followed by a 3D batch normalization layer, and then a ReLU activation layer. The padding of the 3D transposed convolutional layer is set

to 1, thus doubling the size of the input. The output channels of the block are determined by the 3D transposed convolutional layer.

Skip connection Four skip connections are applied between different convolutional and deconvolutional blocks, designed to allow information to be directly passed between the encoder and decoder to address the issue of gradient vanishing in deep convolutional neural networks [16]. It is worth noting that, unlike in [19], the skip connections we constructed originate from the output of a 3D convolutional layer within a convolutional block rather than the output of the entire block and terminate at the output of a specific deconvolutional block. The starting point of the skip connection is copied and concatenated along the channel dimension.

Block/Layer	Type	Input Shape	Output Shape	Skip Connection	Pooling
Encoder1	CB	[1,64,64,64]	[64,64,64,64]	Decoder7	Size halves
Encoder2	CB	[64,32,32,32]	[128,32,32,32]	Decoder5	Size halves
Encoder3	CB	[128,16,16,16]	[256,16,16,16]	-	-
Encoder4	CB	[256,16,16,16]	[256,16,16,16]	Decoder3	Size halves
Encoder5	CB	[256,8,8,8]	[512,8,8,8]	-	-
Encoder6	CB	[512,8,8,8]	[512,8,8,8]	Decoder1	Size halves
Bottleneck1	CB	[512,4,4,4]	[512,4,4,4]	-	-
Bottleneck2	CB	[512,4,4,4]	[512,4,4,4]	-	-
Decoder1	DB	[512,4,4,4]	[512,8,8,8]	Encoder6	-
Decoder2	CB	[1024,8,8,8]	[256,8,8,8]	-	-
Decoder3	DB	[256,8,8,8]	[256,16,16,16]	Encoder4	-
Decoder4	CB	[512,16,16,16]	[128,16,16,16]	-	-
Decoder5	DB	[128,16,16,16]	[128,32,32,32]	Encoder2	-
Decoder6	CB	[256,32,32,32]	[64,32,32,32]	-	-
Decoder7	DB	[64,32,32,32]	[64,64,64,64]	Encoder1	-
Decoder8	CB	[128,64,64,64]	[32,64,64,64]	-	-
Decoder9	CL	[32,64,64,64]	[1,64,64,64]	-	-

Table 2. Summary of 3D U-Net. In the Type column, CB represents convolutional block, DB represents deconvolutional block, and CL represents 3D convolutional layer.

4.2 Conditional GAN implementation

The 3D U-Net architecture we implement is inspired by [57] and modified accordingly. We remove the noise input pathway of the generator in c-GAN, retaining only the condition as the sole input. This decision is motivated by two considerations. Firstly, we aim for the model to generate deterministic synthetic images based on specific conditions (in our case, the CT images of fiber-reinforced composites). Hence, there is no need for diverse noise inputs to diversify image generation. Secondly, we seek the model to focus more on extracting features from the condition for image generation, since the target image is directly evolved from the input image (condition). Retaining the traditional noise input pathway would distract the generator by making it also encode the noise.

After removing the noise input pathway, we implement our own custom architecture for both the discriminator and the generator to encourage the model to generate images that closely resemble the target images (see in Figure 5).

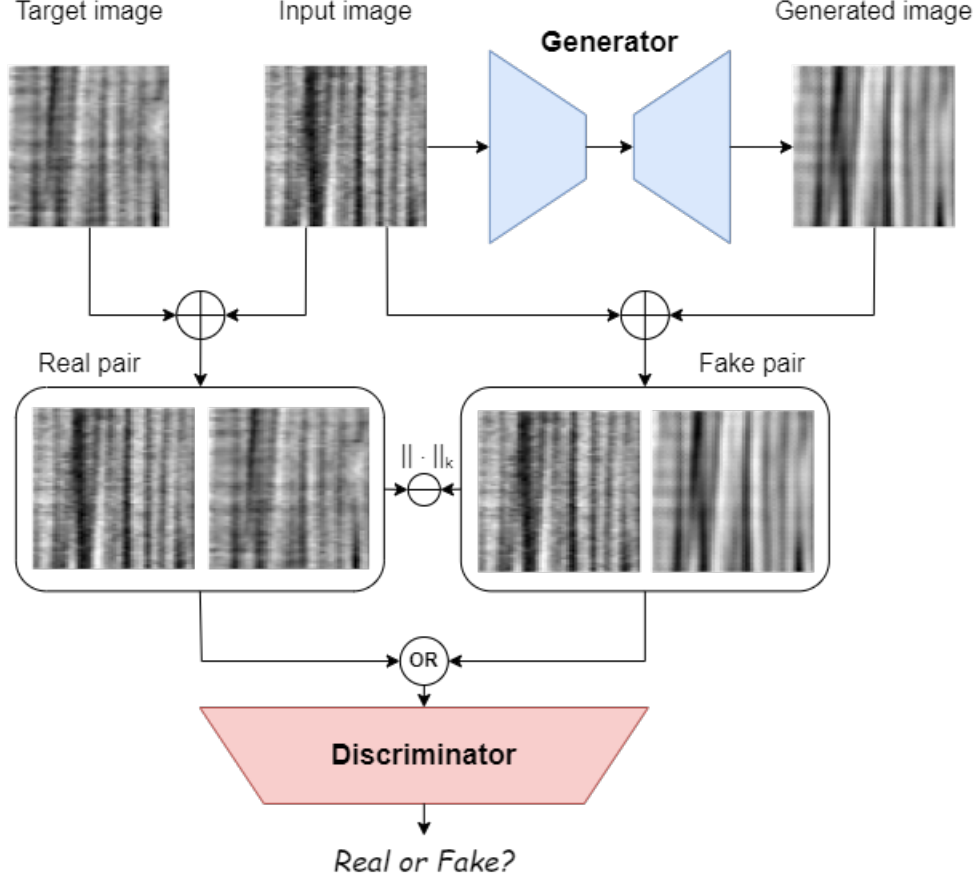


Figure 5. The architecture of our c-GAN. The generator takes an input image and produces a generated image that closely resembles the target image. The input image, target image, and generated image form the real pair and fake pair, respectively. The discriminator receives these image pairs as input and attempts to distinguish between real and fake pairs. Additionally, inspired by [58], we employ the L1 or L2 norm between the input image and the target image as a similarity loss to supervise the improvement of the generator. Introducing this additional loss function can enhance the generator’s ability to synthesize more accurate images.

For the discriminator, our custom implementation is designed based on the following motivations: we not only require it to distinguish whether an image belongs to a real sample or a synthesized one, but also to determine whether an image corresponds to the image under the initial force. Therefore, we modify the discriminator’s input from a single synthesized image or target image to an image pair. This pair consists of either a fake pair composed of an input image and a synthesized image or a real pair composed of an input image and a target image. The discriminator learns to distinguish between real and fake input pairs. Regarding the structure of the discriminator, it comprises 4 convolutional layers and a fully connected layer. [59] have indicated that using LeakyReLU [60] as the activation function is beneficial for preserving gradients and enhancing discriminative capabilities. Therefore, we apply LeakyReLU as the activation function after each convolutional layer in the discriminator. The output of the final convolutional layer is flattened, passed through a fully connected layer, and then processed by a sigmoid function to determine the authenticity of the image pair. The discriminator architecture of our c-GAN is shown in Figure 6. Thus, given an input image x sampled from the distribution $p_x(x)$ and the corresponding target image y sampled from the distribution $p_y(y)$, the discriminator loss of our c-GAN can be expressed as:

$$\mathcal{L}_D = -\frac{1}{2} \mathbb{E}_{x \sim p_{data}(x), y \sim p_y(y)} [\log(D(x, y)) + \log(1 - D(G(x), y))] \quad (1)$$

Here, $D(x, y)$ represents the discriminator’s estimation of the probability that real data x is real given a real pair, $G(x)$ is the generator’s output given input image x , and $D(G(x), y)$ is the discriminator’s

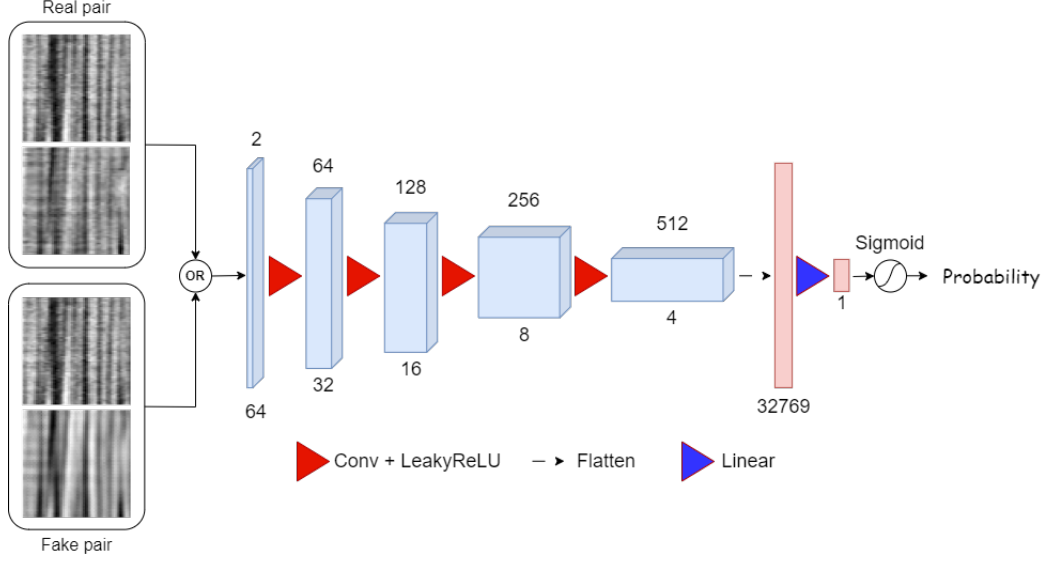


Figure 6. The discriminator architecture of our c-GAN. It receives real image pairs or fake image pairs as input, with images concatenated along the channel axis. These input pairs first pass through 4 3D convolutional layers, with leaky ReLU activation functions after each convolutional layer. The feature map obtained from the last convolutional layer is then flattened and passed through a fully connected layer, followed by a sigmoid function to obtain the final probability of real or fake. The numbers above each cuboid indicate the number of channels in the corresponding image or feature map, while the numbers below the cuboids and rectangles represent the spatial size of the image or feature map.

estimation of the probability that a fake instance is real given a fake pair. The discriminator’s aim is to minimize this loss, thereby improving its accuracy in distinguishing real from fake data.

For the generator, we adopted the 3D U-Net architecture introduced in Section 4.1 to enable it to generate images. However, solely receiving adversarial loss from the discriminator may lead to a scenario where the generator synthesizes images that conform to the structural characteristics of fiber-reinforced composites enough to deceive the discriminator but are not similar to the target images. This occurs because, during training, the features of the target image are not incorporated into the training process of the generator. Consequently, the generator does not learn to produce images that closely approximate the target. This outcome contradicts our objective: we aim for the generator to produce images that not only exhibit the characteristics of fiber-reinforced composites but also closely resemble the target images. To address this, additional constraints are introduced. Inspired by [58], we incorporate a similarity loss to measure the resemblance between the synthesized images and the target images, which is implemented using L1 or L2 norms. Given an input image $x \sim p_x(x)$ and the corresponding target image $y \sim p_y(y)$, the similarity loss of the generator G can be expressed as:

$$\mathcal{L}_{\text{sim}} = \mathbb{E}_{x \sim p_x(x), y \sim p_y(y)} \|y - G(x)\|_k^k \quad (2)$$

Here, $\|\cdot\|_k$ denotes the L1 or L2 norm, depending on the chosen similarity metric. Meanwhile, for the generator’s adversarial loss, we maintain consistency with traditional GAN loss in [26], which can be expressed as:

$$\mathcal{L}_{\text{adv}} = -\mathbb{E}_{x \sim p_x(x), y \sim p_y(y)} \log(D(G(x), y)) \quad (3)$$

Therefore, the total loss of the generator can be expressed as:

$$\mathcal{L}_G = \mathcal{L}_{\text{adv}} + \lambda \cdot \mathcal{L}_{\text{sim}} \quad (4)$$

where λ is a hyperparameter used to balance the adversarial loss and the similarity loss.

Overall, the objective function $V(G, D)$ of our c-GAN can be expressed as:

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{\text{data}}(x), y \sim p_y(y)} [\log(D(x, y)) + \log(1 - D(G(x), y)) + \lambda \|y - G(x)\|_k^k] \quad (5)$$

5 Experiments

5.1 Data pre-processing

The size of the 3D images in the dataset is too large to be directly used for model training (Table 1). We need to split these 3D images into small patches and compose pairs to serve as inputs for our c-GANs. However, the 3D images suffer from displacements over time due to the action of forces (see in Figure 2), leading to a lack of spatial alignment for each voxel in the image sequence as the tension increases. This results in obtaining misaligned image pairs when using the same slice indices across different images. These misaligned image pairs will not only cause generative models to predict images with incorrect positions or shapes but also make it more challenging for them to understand the influence of fiber material microstructures on fiber breaks. Therefore, performing some preprocessing such as cropping and registration to remove these misalignments is necessary for our research.

5.1.1 Cropping

We first use cropping to remove non-fiber material (blank) regions from the 3D images. These blank regions are easily distinguishable from the fiber material on the xy plane (Figure 3a, 2), allowing us to manually crop along the x and y axes to retain the main body of the fiber material. However, manually cropping to directly obtain corresponding regions along the z-axis is challenging because of the high information density in that direction, making it difficult to discern the same features visually. Therefore, image cropping is performed only in the x and y directions, and the alignment of the cropped images is achieved through subsequent registration processes.

Another challenging aspect of cropping is the occurrence of ply splits in fibers under force (Figure 2). These cracks become increasingly obvious with increasing force and significantly disrupt the coherence between adjacent fibers at the edges. This makes registration in the surrounding regions of ply splits very difficult. Moreover, attempting to include these cracks in the registration process may adversely affect the ease of registration for the central regions. Given that these ply splits are not the focus of interest in this paper, we decided to remove them during cropping. However, these cracks only appear under high force, meaning that manual cropping based on the physical characteristics of cracks is not feasible in the initial images. To address this issue, a two-step cropping approach was employed (See in Figure 7). Specifically, all images underwent a first round of cropping to remove only empty regions while preserving all cracks (if any). Subsequently, the cropped image from the last image in the sequence underwent a second round of cropping to eliminate cracks, and the ratio of index changes between the two cropping steps was recorded. Finally, this ratio was applied to all images from the first cropping step to obtain images from the second cropping step, where all existing cracks and potential cracks were removed.

5.1.2 Registration

We use image registration to further align the results of cropping. Since we use patches from the first image as input images in the image pairs, it is reasonable to register all images to the first image. Our two-step cropping reduces misalignment along the x and y axes, but misalignment along the z-axis still exists. Specifically, although the size along the z-axis is uniform (1716 pixels, Table 1), significant movement still occurs along this axis due to the applied force, resulting in the actual aligned region height across the temporal sequence of images being less than 1716 pixels (illustrated in Figure 8).

To address this issue, we extracted the central half region along the z-axis from the first image and further divided it into two equally sized sub-cuboids. Subsequently, we manually selected regions from subsequent images that might match this region and performed registration. Due to the diverse ways in which fibers can move, we utilized three transformations as algorithms for registration, namely rigid, affine, and deformable transformations. All registrations were performed using the Python library of Advanced Normalization Tools (ANTs) [61]. In each case, the transformation type was set to symmetric normalization, with mutual information utilized as the optimization metric.

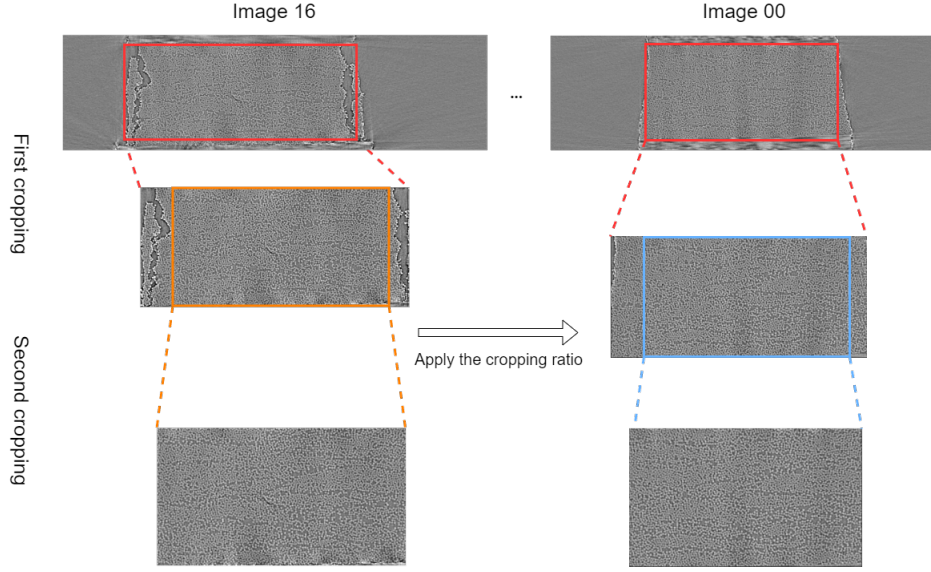


Figure 7. The process of two-step cropping. Firstly, all 3D images undergo a first cropping of their length and width, which retains the majority of fibers and cracks (if any), removes blank portions, and surrounding constrained materials (the red squares). Subsequently, the second cropping aims to eliminate areas of fibers in all images where cracks have already appeared or are likely to occur. Specifically, the last image undergoes cropping to completely remove ply splits (the orange square). Then, the cropping ratio is recorded and applied to all images from the first cropping step to obtain the final cropped images (the blue square).

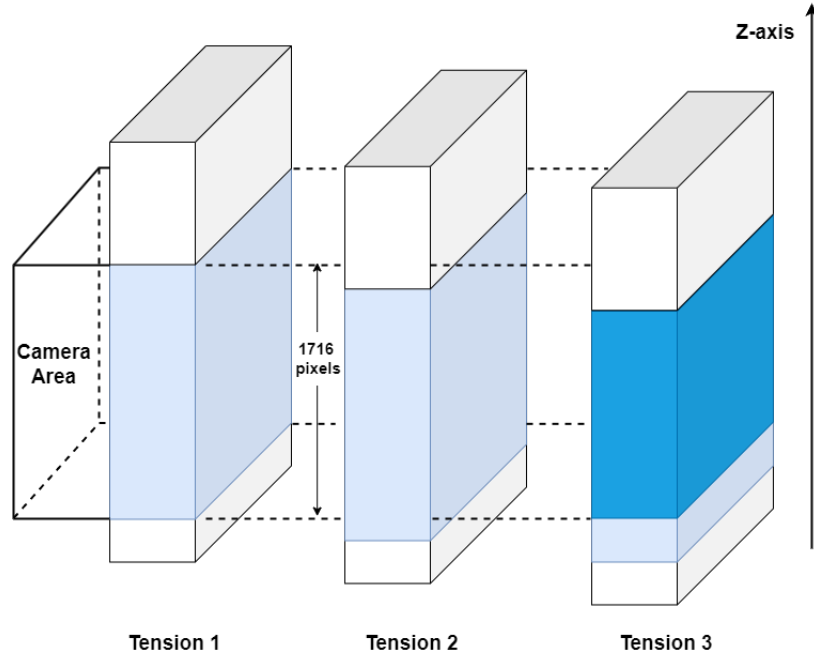


Figure 8. The illustration of the movement of fiber material along the z-axis as tension increases. The three cuboids represent the relative positions of the fiber material under different forces along the z-axis. The light blue area represents the region captured by the camera under tension 1. As the applied force increases from left to right, the fiber material as a whole moves downward, while the camera's position remains fixed. Consequently, the region captured by the camera shifts upward relative to the fiber material. Although the height of the captured region remains constant (1716 pixels), the actual common region (depicted in dark blue) in the captured images under different tensions is less than 1716 pixels in height.

5.1.3 Dataset Split

We selected a subset of the data for the following experiments and paired the selected images for training and validation. We selected two pairs of images for the experiments: the 0th image with the 4th image, and the 0th image with the 12th image. After cropping and registration, the dimensions of the images are all $429 \times 689 \times 410$, which is consistent with the fixed image, i.e., the 0th image.

To further explain how we split the data, we use the image pair of the 0th and 4th images as an example. We divided both the 0th and 4th images into 720 non-overlapping cubes of size $64 \times 64 \times 64$ for ease of model training. Due to the images' dimensions not being divisible by 64, any parts exceeding multiples of 64 voxels but falling short of 64 voxels will be discarded. Consequently, we obtained 720 patches, each containing an input cube from the 0th image and a corresponding target cube from the 4th image. These 720 patches were divided into training, validation, and test sets in a ratio of 80:15:5. The same process was repeated for the image pair of the 0th and 12th images.

After splitting the data, each image pair in dataset 00-to-04 and dataset 00-to-12 consists of two parts: an input image from the 0th image and a target image from either the 4th image or the 12th image. The input images in both datasets are identical, as they are derived from the 0th image in the same way; the difference between the datasets lies solely in the target images. According to the original dataset (Table 1), the 12th image, compared to the 4th image, was obtained under higher tension in fiber-reinforced composites, implying greater deformation and more fiber breaks. This also indicates that for the model, learning the underlying distribution from Image pair 00-to-12 is more challenging due to the larger discrepancy between the input image and the target image.

It's worth noting that patches from different image pairs will not be used in the same model. In other words, a model will only be trained using patches from either the image pair of the 0th and 4th images or the image pair of the 0th and 12th images. The purpose of this approach is to investigate the performance of our models on the simpler dataset (Image pair 00-to-04) and the more challenging dataset (Image pair 00-to-12) to explore the robustness of the models.

5.2 Training settings

Our models' training and evaluation were entirely implemented using PyTorch and utilized an RTX 4090 with 24 GB of vRAM as the accelerator. For all models, we use the Adam [62] optimizer with momentums set to $\text{beta1} = 0.5$ and $\text{beta2} = 0.999$. The complete code can be found here¹.

To investigate the impact of different settings on model performance, we trained a series of models. All models adhered to the specific details outlined in the Section 4 and were trained independently of each other. Table 3 summarizes all the models we implemented and trained.

5.2.1 Loss function

We employed different loss functions to train the 3D U-Net and c-GAN to explore how different loss functions would affect the training process and influence the model's performance.

For the 3D U-Net, we employed both MAE and MSE loss functions separately. In each training step, the 3D U-Net model U receives an input image x and its corresponding target image y , and generates an image $U(x)$. The loss function L can be expressed as:

$$\mathcal{L}_U = \begin{cases} \mathbb{E}[\|y - U(x)\|_1] & \text{for } MAE \\ \mathbb{E}[\|y - U(x)\|_2^2] & \text{for } MSE \end{cases} \quad (6)$$

For the c-GAN, we employed Binary Cross-Entropy (BCE) loss to implement the adversarial loss for both the generator (3) and the discriminator (1). Additionally, similar to the training of the 3D U-Net, we utilized MAE and MSE as the similarity loss (2) for the generator separately. In particular, we chose $\lambda = 100$ as the weight for the similarity loss in (4).

¹<https://github.com/XiangyuLu/Fiber-Break-Prediction>

Model Name	Type	Training Set	Criterion	Learning Rate
UNet-04-A	3D U-Net	Image pair 00-to-04	MAE	1e-4
UNet-04-S	3D U-Net	Image pair 00-to-04	MSE	1e-4
UNet-12-A	3D U-Net	Image pair 00-to-12	MAE	1e-4
UNet-12-S	3D U-Net	Image pair 00-to-12	MSE	1e-4
cGAN-04-A	c-GAN	Image pair 00-to-04	MAE	1e-6
cGAN-04-S	c-GAN	Image pair 00-to-04	MSE	1e-6
cGAN-12-A	c-GAN	Image pair 00-to-12	MAE	1e-6
cGAN-12-S	c-GAN	Image pair 00-to-12	MSE	1e-6

Table 3. Summary of Trained Models. Each model name is composed of three parts, connected by "-". The first part indicates the architecture used, corresponding to the "Type" column. The second part indicates the training set used for the model, where, for example, "04" represents training on the Image pair 00-to-04. The third part indicates the metric used during training and validation, where "A" stands for MAE and "S" stands for MSE.

5.2.2 Early stopping

Early stopping is a form of regularization used to prevent overfitting in training machine learning models [63]. In our experiments, we adopted different early stopping strategies for 3D U-Nets and c-GANs.

During the training of the 3D U-Net, a uniform loss function (either MAE or MSE) was applied throughout both the training and validation processes. Hence, the magnitude of the validation loss can fairly represent the model’s performance. For all 3D U-Net models, a fixed number of 400 epochs was allocated for training, and the validation loss was recorded at the end of each epoch. The model corresponding to the epoch with the minimum validation loss was selected as the final model.

However, the same strategy is not suitable for c-GANs. Firstly, c-GANs comprise two networks with different objectives, making it challenging to define a clear stopping point based on objective function (5), as improvements in one network may lead to degradation in the other. What’s more, unlike the traditional cGAN’s generator loss, which only consists of BCE loss, our c-GAN’s generator loss also incorporates the similarity loss between generated and target images, achieved through MAE or MSE loss. Using this fusion loss to evaluate validation performance does not ensure consistency in measurement standards, making validation loss from different periods incomparable.

To address this issue, at the end of each epoch of training, we evaluated the performance of the generator on the validation set, considering only the similarity loss at this stage, while the traditional generator loss and discriminator loss were ignored. After a specified number of epochs, the model corresponding to the epoch with the minimum validation loss was selected as the final model.

6 Results

Table 4 presents the evaluation results of the trained models on their respective validation sets. Each model’s evaluation metric aligns with the criterion used during training. The models were selected using the early stopping strategy outlined in Section 5.2.2. To mitigate the impact of uncertainties due to random initialization of model weights, each model was independently trained five times. The model with the smallest validation loss at the point of early stopping from these five trials was selected as the final result. Additionally, we further trained the U-Net models from [19] on the Image pair 00-to-04 and Image pair 00-to-12 datasets using MSE as the criterion, serving as the baselines for comparison. It is important to note that we did not implement the baseline architecture ourselves but rather utilized the open-source code provided in [64].

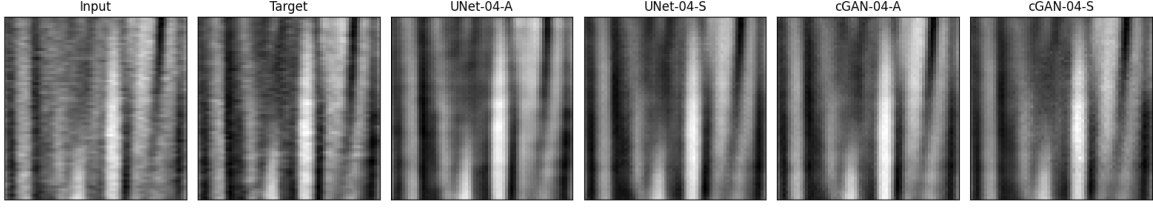


Figure 9. A generation example of models trained on 00-to-04 dataset

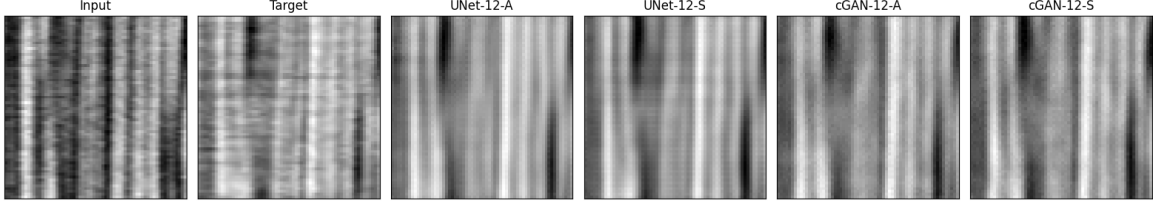


Figure 10. A generation example of models trained on 00-to-12 dataset

Model Name	Validation Set	Evaluation Matric	Validation Loss
UNet-04-A	Image pair 00-to-04	MAE	0.0265
cGAN-04-A	Image pair 00-to-04	MAE	0.0263
UNet from [19]	Image pair 00-to-04	MSE	0.0205
UNet-04-S	Image pair 00-to-04	MSE	0.00114
cGAN-04-S	Image pair 00-to-04	MSE	0.00107
UNet-12-A	Image pair 00-to-12	MAE	0.0373
cGAN-12-A	Image pair 00-to-12	MAE	0.0369
UNet from [19]	Image pair 00-to-12	MSE	0.0222
UNet-12-S	Image pair 00-to-12	MSE	0.00286
cGAN-12-S	Image pair 00-to-12	MSE	0.00278

Table 4. The evaluation results of the trained models on their respective validation sets. Each model name corresponds to those listed in Table 3. Additionally, we further trained the UNet models from [19] as the baselines. Bold text indicates models with lower validation loss under the same training settings (training set and criterion), signifying that the model with this architecture performs better for the given conditions.

To more intuitively assess the visual quality of the generated images, we randomly selected a series of generation examples in validation set for analysis. It should be noted that, for the sake of presenting the synthesized 3D images, all generation examples in this section are the central cross-sections along the y-axis.

First, we give overall examples of generated images from each model on their corresponding validation sets in Figures 9 and 10. For ease of comparison, examples of generated images from models trained on the same dataset are grouped together in a single figure.

To provide a more specific demonstration of the performance of different models on various datasets, we present Figure 11 and Figure 12. Figure 11 shows a generation example of UNet-04-A and UNet-04-S on the 00-to-04 validation set, while Figure 12 illustrates a generation example of cGAN-12-A and cGAN-12-S on the 00-to-12 validation set. Notably, no fiber breaks were observed in these two figures. Additionally, two generation examples of cGAN-12-A and cGAN-12-S on the

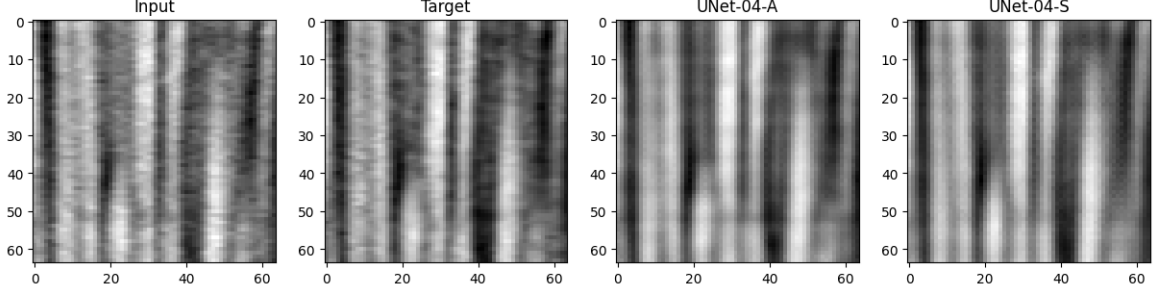


Figure 11. A generation example of UNet-04-A and UNet-04-S on 00-to-04 validation set. The models demonstrate the ability to synthesize images similar to the target images based on the input images.

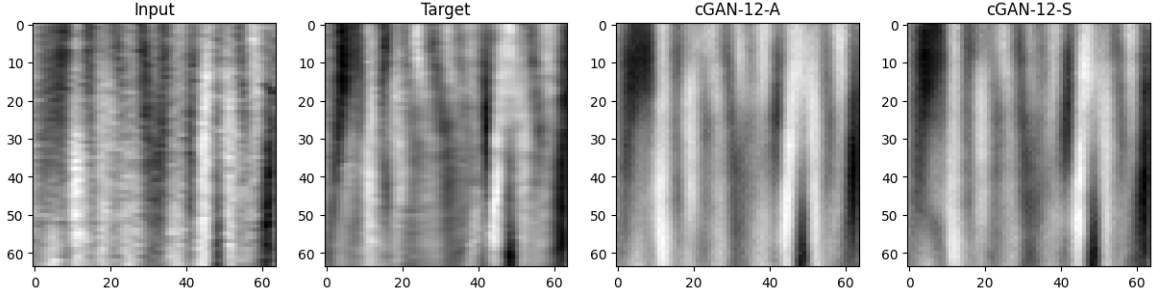


Figure 12. A generation example of cGAN-12-A and cGAN-12-S on 00-to-12 validation set. The models can still generate images that are similar to the target when the input image and target image exhibit greater differences.

00-to-12 validation set, where fiber breaks appear in the target image, are presented in Figure 13. Our models fail to predict the location or shape of the fiber breaks in these examples.

Our experimental setup allows us to explore how MAE and MSE impact the performance of models. We compared models trained on the same dataset but using different criteria. Specifically, Figure 14 shows a generation example from the validation set using the 3D U-Net models trained with MAE and MSE, respectively. Figure 15 shows a generation example from the validation set for cGAN models trained with MAE and MSE, respectively.

We are curious whether the introduction of the c-GAN architecture can help the model obtain better performance. Therefore, we compared the quality of generated images between c-GANs and standalone 3D U-Net. Figure 16 shows a generated example from the 00-to-04 validation set by UNet-04-S and cGAN-04-S. It can be observed that in the generated image by UNet-04-S, some fibers are incorrectly enlarged (as indicated by the green circle). In contrast, the image generated by cGAN-04-S more closely resembles the target image. Figure 17 shows a generated example from the 00-to-12 validation set by UNet-12-A and cGAN-12-A. In the images generated by UNet-12-A, some fiber edges blend into the background (highlighted by the red circles), making the boundaries between fibers indistinct. In contrast, the images generated by cGAN-12-A exhibit sharper edges, making the fiber boundaries clear and easy to distinguish. Moreover, The results from UNet-12-A sometimes fail to capture relative positions of the fibers (highlighted by the blue circles), making certain fibers appear to lie on the same plane.

In addition to evaluation metrics and generated image examples, we also recorded training statistics. We report the training process records for every model we trained in Table 5. Although all models were trained for 400 epochs, the epoch number at which each model’s final version was selected varies due to the early stopping. Additionally, we recorded the time required for each model to complete one epoch of training. This time represents the duration needed for the model to train once over the entire training set, excluding the time spent on evaluation on the validation set after each epoch.

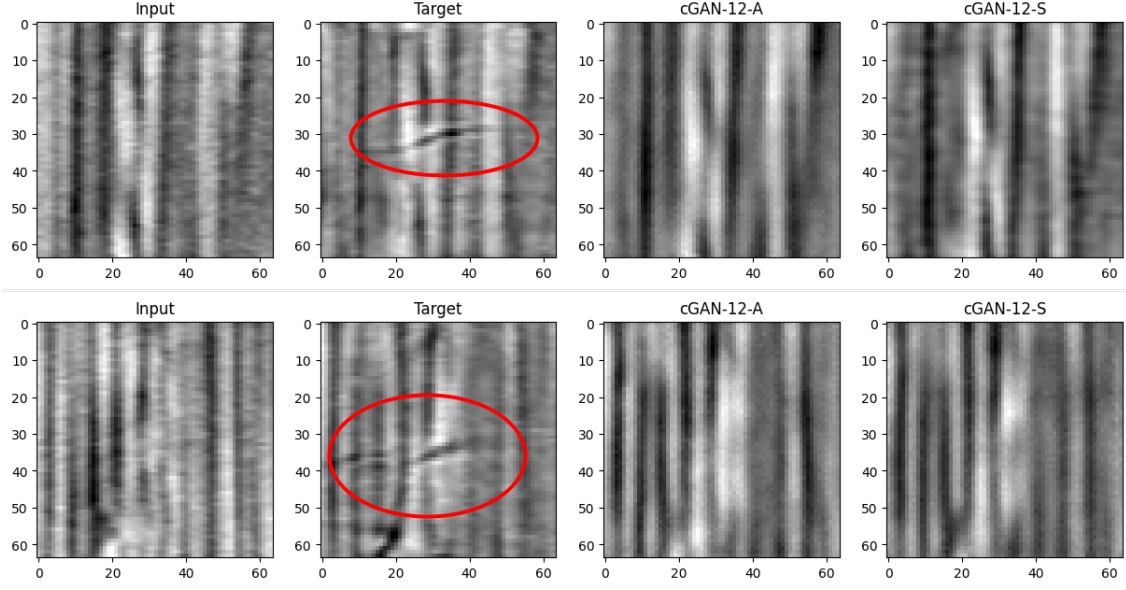


Figure 13. Two generation example of cGAN-12-A and cGAN-12-S on 00-to-12 validation set. In the examples, neither model successfully predicted fiber breaks. However, both models generated other details of the fiber structure.

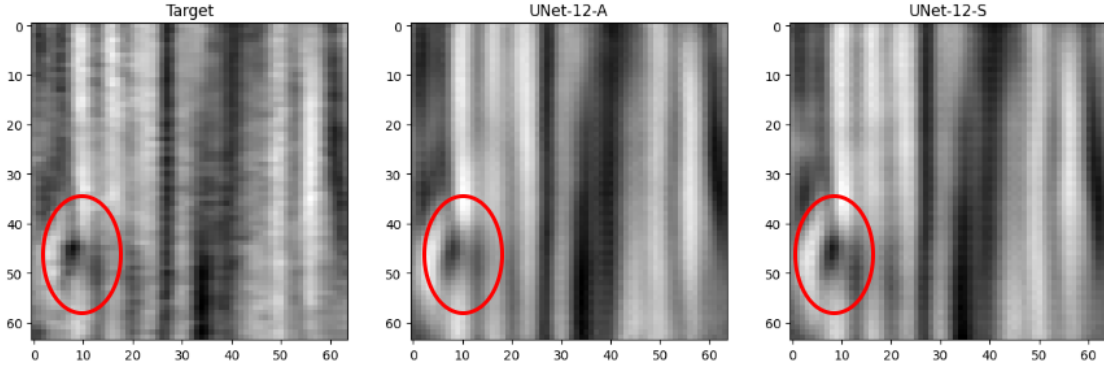


Figure 14. A generation example of UNet-12-A and UNet-12-S on 00-to-12 validation set. The model trained with MAE produces smoother images while losing a lot of detail. In contrast, the model trained with MSE generates more accurate details (highlighted by the circles).

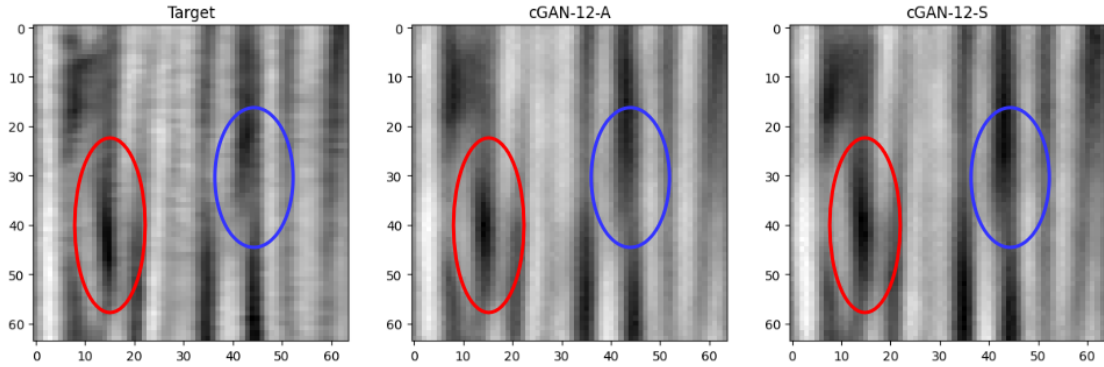


Figure 15. A generation example of UNet-12-A and UNet-12-S on 00-to-12 validation set. UNet-12-S exhibited issues such as incorrectly enlarging gaps between fibers (highlighted by the red circles) and overlooking fiber branches (highlighted by the blue circles) when generating the fiber structure. In comparison, UNet-12-A performed better in these aspects.

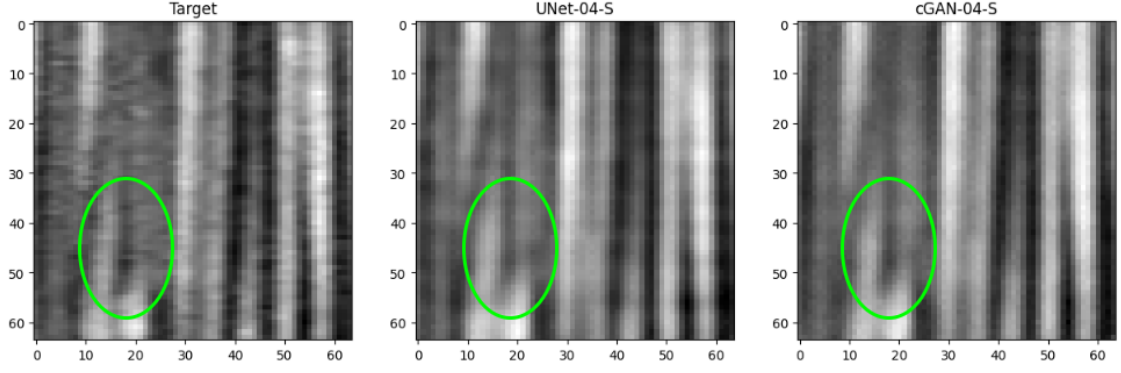


Figure 16. A generation example of UNet-04-S and cGAN-04-S on 00-to-04 validation set. Compared to UNet-04-S, the fibers generated by cGAN-04-S are finer and more closely resemble the target (green circles).

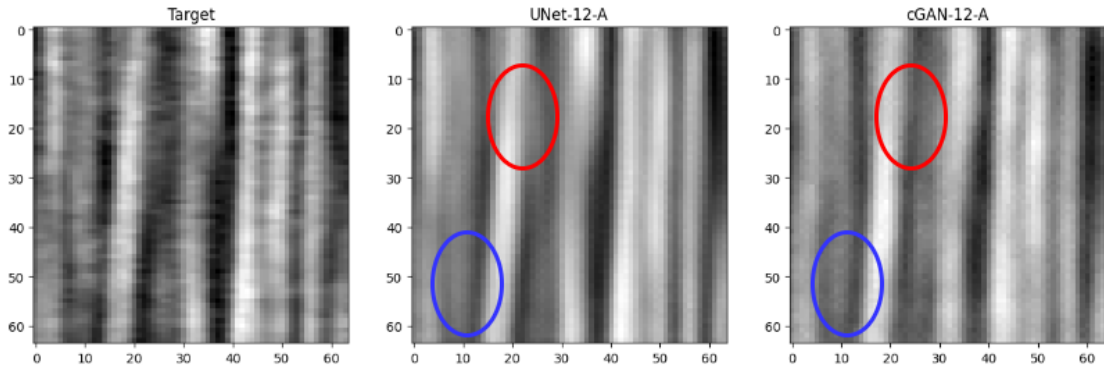


Figure 17. A generation example of UNet-12-A and cGAN-12-A on 00-to-12 validation set. Compared to UNet-12-A, cGAN-12-A demonstrates superior fiber edge processing capabilities (red circles) and a better perception of the relative positions between fibers (blue circles).

Model Name	Training Set	Final Model Epoch	Epoch Time (s)
UNet-04-A	Image pair 00-to-04	47	18
cGAN-04-A	Image pair 00-to-04	147	23
UNet from [19]	Image pair 00-to-04	42	27
UNet-04-S	Image pair 00-to-04	35	18
cGAN-04-S	Image pair 00-to-04	215	23
UNet-12-A	Image pair 00-to-12	48	18
cGAN-12-A	Image pair 00-to-12	147	23
UNet from [19]	Image pair 00-to-12	72	27
UNet-12-S	Image pair 00-to-12	80	18
cGAN-12-S	Image pair 00-to-12	176	23

Table 5. The training process records. Each model name corresponds to those listed in Table 3. The "Final Model Epoch" column records the epoch number at which the final version of each model was selected. "Epoch Time (s)" indicates the time (in seconds) each model took to complete one training epoch on the entire training set. This time does not include the evaluation time on the validation set at the end of each epoch.

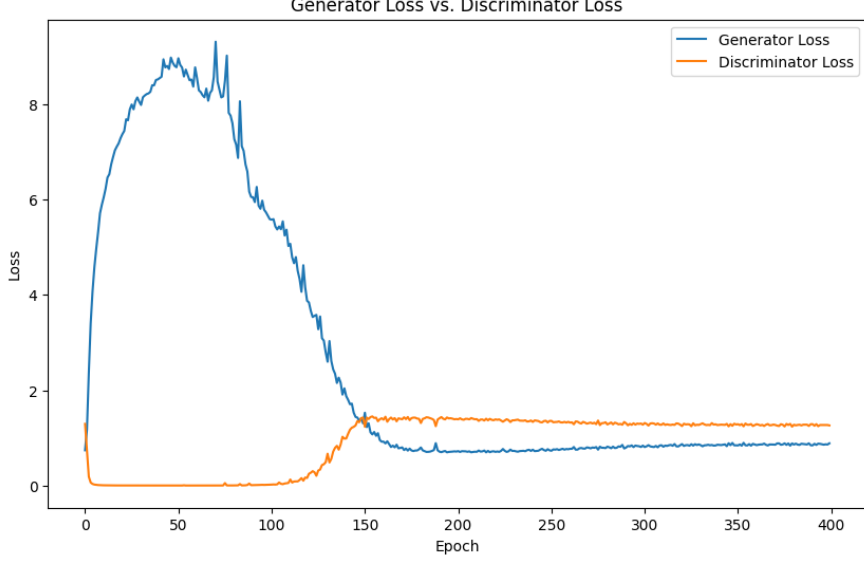


Figure 18. The loss curves of the generator and discriminator of the cGAN-12-S model over epochs. The generator loss plotted in the figure excludes the similarity loss (2), containing only the adversarial loss (3) incurred during the generator training process.

We report the loss curves of the generator and discriminator of the cGAN-12-S model over epochs in Figure 18 as an example to illustrate the adversarial training process. It is important to note that, to solely display the adversarial interaction, the generator loss plotted excludes the similarity loss (2), retaining only the traditional adversarial loss (3), as the similarity loss does not directly influence the discriminator’s improvement. In the early stages of training (first 10 epochs), the discriminator quickly learns to distinguish the generated images from real ones, causing the generator’s loss to spike. As adversarial training continues (50-200 epochs), the generator progressively learns the features of the fiber structure and begins to produce increasingly realistic images. After 200 epochs, both the generator and discriminator losses stabilize.

We report the validation loss curves over epochs for models UNet-04-S and cGAN-04-S in Figure 19. It is worth noting that although the two models use different learning rates ($1e-4$ and $1e-6$, respectively), the weight for the similarity loss in the generator loss function of cGAN-04-S is set to 100, making the equivalent learning rates for the MSE similarity loss consistent at $1e-4$ for both models. This allows for a fair comparison of the validation loss at the same epoch for UNet-04-S and cGAN-04-S.

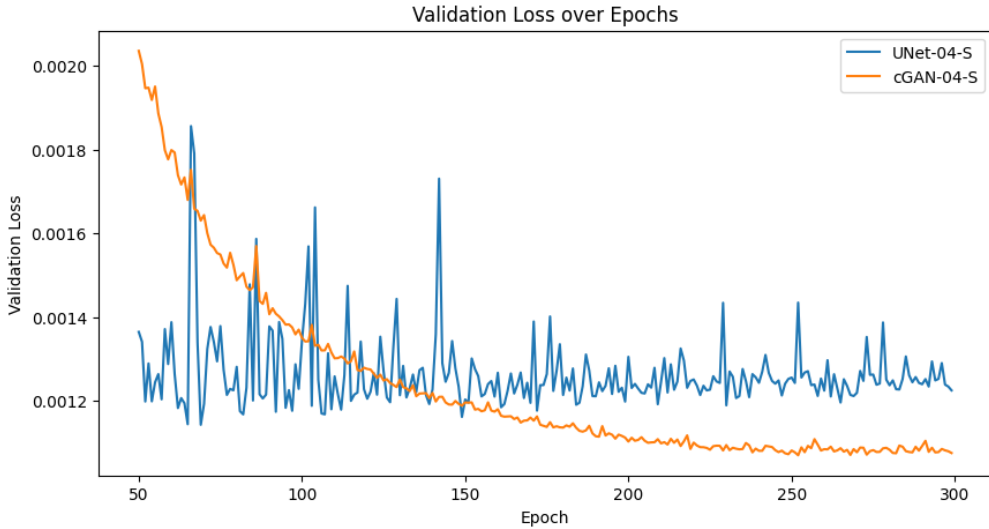


Figure 19. The validation loss curves over epochs for models UNet-04-S and cGAN-04-S.

7 Discussion

Our 3D U-Nets and c-GANs achieved lower MSE losses on the validation set compared to the baseline from [19] (Table 4), indicating that our implemented architectures have superior capability in generating FRC CT images. Additionally, all of our c-GANs achieved lower validation losses than the standalone 3D U-Nets under all training settings, suggesting that the introduction of the c-GAN architecture enhances the understanding and reconstruction of fiber structures by the 3D U-Net. The quality of the generated images (Figure 9, 10) demonstrates that all our 3D U-Nets and c-GANs perform well in predicting potential fiber structure movements under increasing tension, further indicating that our models have an excellent understanding of fiber structure features.

Specifically, our implemented UNet-04-S and UNet-12-S models both achieve lower MSE losses compared to the baseline (see in Tabel 4), indicating that our custom 3D U-Net outperforms the baseline in synthesizing FRC CT images. This improvement is likely due to the deepening of both the downsampling and upsampling paths (Section 4.1), enhancing the model’s ability to extract fiber structure information and reconstruct images from feature maps [65]. Furthermore, our modified skip connections likely contribute positively as well. In our 3D U-Net, the skip connections link feature maps with the same number of channels, instead of connecting feature maps with different channel numbers as in the baseline. This uniform channel connection approach may enable the model to better understand the underlying information at the same level, thereby gaining a deeper comprehension of the fiber structure characteristics.

Additionally, we observed an interesting phenomenon: despite our 3D U-Net models being deeper and containing more convolutional and deconvolutional blocks compared to the baseline, they require less time to train for one epoch on the same dataset (Table 5). This indicates that our models are more efficient in data processing while maintaining high performance. We suspect this efficiency gain is due to our reduction in the number of convolutional layers in the first and second convolutional blocks and these layers contribute minimally to learning the underlying distribution of the training data. The potential performance loss from removing these layers is likely compensated for by the deeper convolutional blocks and the extended bottleneck.

However, these quantitative metrics can provide an approximation of the overall difference between the generated and target images but may not adequately reflect finer details and visual quality. This is because the microstructure of fiber-reinforced composites is often highly complex [66], and the models’ ability to understand different fiber structures may vary. Therefore, average values might not accurately reflect the actual performance. Therefore, we further evaluated the performance of our models by visually inspecting the quality of the generated images.

We observe that for image pairs without fiber breaks, our models accurately synthesize images that closely resemble the target images, regardless of whether the input and target in the pair have small differences (Figure 11) or large differences (Figure 12). This indicates that the models have a deep understanding of the structural characteristics of fiber-reinforced composites and can predict how the fiber structure will change under different forces. However, when fiber breaks appear in the target images, our models fail to predict their location or shape effectively (Figure 13). Instead, they treat such cases as image pairs without fiber breaks, merely predicting how fibers will move. This failure to predict fiber breaks is not an isolated incident; in all the generation examples we manually inspected where fiber breaks appear in the target images, our models failed to predict fiber breaks (we only show two of them in Figure 13). Although due to time constraints, we did not manually inspect all generated images on the validation set, we still tend to suspect that none of our models have learned how to predict fiber breaks. Despite the presence of fiber breaks, the models still demonstrate excellent predictive capabilities for fiber structure on these pairs, accurately generating fiber structures similar to those in the target images (Figure 9, 10, 11, 12, 13). This situation occurs consistently across all models.

This situation indicates that our models lack the ability to capture and extract specific detailed features such as fiber breaks during training. Instead, they tend to focus on global features like the overall fiber structure. As a result, the generator (i.e., 3D U-Net) primarily learns the distribution of the fiber structure and emphasizes reconstructing this information during the upsampling phase. [67, 38] report the same phenomenon where the GANs have unbalanced performance within one sample. We suspect that this issue arises from using overly simplistic measurements to evaluate

the quality of the generated images during training. This approach likely fails to guide the model to extract local features and accurately reproduce them in the generated images. Although [68, 69] improved the performance of GANs in generating local details of images by increasing model complexity and introducing latent representations, this approach demands greater computational resources and requires even more for more complex tasks (such as 3D scenes). Using a loss function specifically tailored to evaluate the quality of fiber breaks might enhance the model’s ability to generate fiber breaks [70, 71], but such a highly specific loss function may be challenging to generalize to other application scenarios.

Despite our models’ failure to predict fiber breaks, their excellent performance in predicting potential fiber structure movements under increasing force still holds application potential. For instance, the generated FRC CT images may be further utilized for fiber orientation prediction [72], which can be applied to modify the manufacturing process of FRCs or to control the mechanical performance of the resulting parts. Additionally, the unseen FRC CT images under high force generated by our models may be used as new datasets to evaluate existing FRC stiffness prediction models [73, 74], thereby reducing the cost of dataset creation.

Our experimental setup has also highlighted several aspects worth exploring. These aspects have influenced model performance to varying degrees:

1. (Section 7.1) How does using different loss functions (MAE and MSE) affect models’ performance?
2. (Section 7.2) Does using a cGAN structure lead to a positive improvement over using only 3D-Unet?
3. (Section 7.3) Do models exhibit different performance when trained with image pairs under different forces?

Additionally, we conducted a detailed analysis of why our models failed to focus on and generate fiber breaks, and we proposed feasible future work to address these issues (Section 7.4).

7.1 MAE vs MSE

Overall, we observe MAE and MSE models are effective at predicting the specific morphology of fiber structures under a given force, although they both fall short in predicting potential fiber breaks (Table 4, Figure 11, 12, 13). This suggests that neither MAE nor MSE enables the model to focus on analyzing the relationship between the occurrence of fiber breaks and the underlying microstructure.

In terms of performance differences, we observe that the 3D U-Net model trained with MAE produces smoother images, losing many details (Figure 14). In contrast, the model trained with MSE successfully generates more details that match the target image. This may be because MAE penalizes the absolute difference between predicted and target values, causing it to prioritize reducing the overall magnitude of errors across all voxels rather than focusing on specific details [75]. In contrast, the squared calculation in MSE makes it more sensitive to outliers and tends to focus on reducing these larger errors [76], thereby encouraging the generator to focus more on details during training.

However, MSE is not always positive. We observe the c-GAN model trained with MSE overly emphasizes local details (Figure 15), resulting in some fiber gaps being incorrectly enlarged (highlighted by the red circles) and some fiber branches being overlooked (highlighted by the blue circles). In contrast, the model trained with MAE performs better in this regard. We suspect that during training, the characteristic of MSE, which tends to prioritize minimizing large errors, causes the model to sacrifice global coherence and structure in favor of focusing excessively on matching every pixel value precisely, even if outliers are present as noise. The model may incorrectly amplify small inconsistencies or noise in the input data, leading to details in the generated fiber structure that are opposite to those in the target image.

Through the visual quality observation of the generated images (Figure 14, 15), it is difficult to directly determine whether MAE or MSE is more suitable for training 3D U-Net and c-GAN models. In most cases, both metrics successfully guide the models in learning the underlying data distribution (Table 4), with only minor differences in the details of the generated images (Figure

15, 14). Therefore, in experimental setups, the choice of training metric should be made based on the specific context and requirements. Generally speaking, MSE is more suitable for datasets that are relatively clean and have few outliers or noise [75], which can be achieved through thorough data preprocessing [77, 78]. However, in this thesis, the experiments only involved cropping and registration of the dataset, neglecting the cleaning of noise and outliers. This resulted in MAE, with its lower sensitivity to outliers, performing better on some samples compared to MSE (Figure 15).

7.2 Impact of GAN architecture

We observe a complementary relationship between the generator and discriminator losses (Figure 18). Specifically, in the generator and discriminator, the loss of one decreases as the other’s increases, because good performance by one indicates poor performance by the other. The loss curves demonstrate that the generator and discriminator of the introduced c-GAN architecture are indeed engaged in adversarial training. Furthermore, this obvious adversarial process manifested in the loss curves also indicates that our models did not suffer from mode collapse [79, 80], suggesting that the models learned a more comprehensive underlying distribution of the generated images rather than focusing on a few concentrated modes.

From Table 4, we observe that all of the models incorporating the c-GAN architecture exhibit lower validation loss compared to those using only the 3D U-Net across different training sets and criterion. This indicates that the adversarial training process inherent to the c-GAN architecture can improve the performance of the models. We speculate that this is due to the effective enhancement of the discriminator’s performance by the c-GAN architecture, and the excellent discriminative ability of the discriminator further advises the generator to generate higher quality images [26]. Specifically, we suspect that by conditioning the models with input images, the discriminator learns how to distinguish whether the generated images and target images are reasonable in terms of fiber structure, based on the underlying distribution of the given input images [81]. This ability of the discriminator further guides the generator to learn the features of the fiber structure during adversarial training and learn a consistent distribution of generated data with the ground truth [27].

From the perspective of visual quality, we observe models using the c-GAN architecture generate images with more detail and closer resemblance to the target images, whether on the 00-to-04 dataset or the 00-to-12 dataset (Figures 16 and 17). This indicates that the c-GAN architecture enhances the refinement and quality of the generated images and helps the model gain a deeper understanding of the microstructure of the fiber material. Specifically, we find that the c-GAN models exhibit superior fiber edge processing capabilities (red circles in Figure 17) and a better perception of the relative positions of different fibers (blue circles in Figure 17). We think this is because the introduction of conditioning allows the adversarial training process of c-GANs to act as a form of regularization [82], thereby improving generalization to unseen data and making the model more robust. Additionally, we suspect the introduction of similarity loss encourages the generator network to produce images that are not only visually realistic but also preserve fine details. Compared to standalone 3D U-Nets, the generator-discriminator architecture of c-GANs amplifies this guidance.

Additionally, we observe that the introduction of the c-GAN architecture contributes to enhanced stability during model training (Figure 19). The model with the c-GAN structure exhibits smoother validation loss curves during training compared to UNet-04-S, reducing a lot of oscillations. This suggests that the adversarial training employed by c-GANs contributes to a more stable training process for the model. This stability enhancement may be attributed to the minimax game dynamics between the generator and discriminator during training, which maintains a dynamic equilibrium, leading to smoother parameter updates for the generator.

At the same time, we find that all c-GAN models require longer training time to reach the best performance compared to 3D U-Nets (Table 5). It is worth noting that although c-GANs and 3D U-Nets use different learning rates during training, the hyperparameter λ in (4) balances the contribution of the similarity loss to parameter updates in each epoch for both architectures. In other words, this difference in convergence time stems from the introduction of the c-GAN architecture. This indicates that the adversarial training process of the generator and discriminator requires more time to learn to effectively counteract each other’s strategies. Considering the imbalanced performance of our c-GAN models in predicting fiber breaks, we suspect this phenomenon also exists during our models’ training process [83]. The poor generation of local details may have hindered the convergence speed,

necessitating a longer training period for the models to learn effectively.

7.3 Variability in model fitting on different datasets

From the validation loss perspective, we notice it is more challenging for the models to learn the common underlying distribution from Image pair 00-to-12 compared to Image pair 00-to-04 (Table 4). This suggests that the greater differences in the image pairs of dataset 00-to-12 make it more difficult for the models to learn the underlying distribution of the fiber structure under higher tension. This is reasonable because larger tensions typically result in more intense and complex fiber movement and deformation [84], making it harder for the models to generalize the changes in different fiber microstructures under tensile loading.

Additionally, we observe that our c-GAN exhibits greater robustness compared to standalone 3D U-Nets (Table 4). This is reflected in the greater disparity in performance between the 3D U-Net and c-GAN on Image pair 00-to-12 compared to Image pair 00-to-04. On the simpler dataset (Image pair 00-to-04), all c-GAN models outperform the standalone 3D U-Net models according to the validation loss. This advantage is magnified on the more challenging dataset (Image pair 00-to-12). We speculate this is because the more difficult image pairs impose higher demands on the model’s feature extraction and reconstruction capabilities, and the generator-discriminator architecture of the c-GAN compensates for this performance gap [57].

The same phenomenon occurs with our customized 3D U-Net compared to the baseline from [19]: when the training and validation datasets change from Image pair 00-to-04 to Image pair 00-to-12, the performance advantage of the 3D U-Net over the baseline becomes more pronounced, as indicated by the difference in validation loss (Table 4). Considering that our 3D U-Nets have deeper downsampling and upsampling paths as well as a longer bottleneck, we suspect these additional convolutional and deconvolutional blocks enhance the model’s ability to handle more complex data [85].

One insight gained from the above discussion is that appropriately increasing model complexity can enhance the robustness of the models. However, model complexity and model expressive capacity (the ability of models to approximate complex problems) are often not equivalent [86]. Our c-GANs achieved a minimum MSE loss of $6.28\text{e-}05$ on the training set Image pair 00-to-12 after 1600 epochs of training. This is obviously smaller than their validation set loss ($2.78\text{e-}3$, see in Table 4), suggesting that our models are over-parameterized and have sufficient potential expressive capacity for the complex dataset. However, our models fail to capture both the fiber breaks in Image pair 00-to-04 and Image pair 00-to-12, indicating that our training process has not fully exploited their expressive capacity. Although increasing the complexity of the generator or discriminator may indirectly enhance the expressive capacity of c-GAN, this strategy also carries the risk of overfitting [87].

7.4 Future works

Class imbalance The imbalance of voxel numbers between fiber breaks and the fiber structure background is a significant issue in our dataset. Even in image 15, which contains the highest number of fiber breaks, there are only 286 fiber breaks (see in Table 1). Based on manual observation, individual fiber breaks in the CT images of the dataset typically have a length and width of 10-50 voxels and a height of only 1-5 voxels. Even under the most conservative estimates, the volume of fiber breaks accounts for merely 0.65% of the entire volume. It is extremely challenging for the model to learn how the microscopic structure of the fiber material influences the occurrence of fiber breaks and to accurately predict fiber breaks in unseen samples from such a sparse distribution [88]. In reality, the model is more likely to disregard the few fiber breaks as potential noise [89] and focus solely on the more common fiber structure features.

To ensure that the model learns more about the fiber breaks features, it is necessary to address the class imbalance. One potential solution is to filter the training set and use only patches containing a higher number of fiber breaks for training [90]. This approach effectively increases the proportion of fiber break voxels but requires more datasets to replace the discarded patches. Additionally, such dataset filtering based on the number of fiber breaks can be labor-intensive.

Another potential solution is to utilize a loss function that specifically targets fiber breaks instead of relying on MAE or MSE as the similarity loss. Many region-based loss functions [91, 92] focus on

specific areas of interest within an image by providing a label mask for computing the loss. Although the dataset used in this thesis does not come with annotated masks for fiber breaks, these masks can be obtained through segmentation. Some segmentation models [93, 94] have demonstrated their ability to segment fiber breaks. Through segmentation, we can treat the fiber structure background in the generated and target images as 0, considering only the fiber breaks area as 1. Under these conditions, applying region-based loss to the processed images can effectively reduce the loss proportion of non-fiber break regions in the training metric. However, training with this solution not only depends on the accuracy of the segmentation model but also requires additional computational resources.

Integrating Force into conditions We suspect that incorporating the force difference between the target image and the input image as an additional condition for the model could potentially enhance its performance. In this thesis, each model is trained solely on image pairs obtained under specific forces. During this process, the only information related to the applied tension is conveyed to the network through the similarity loss. However, due to the depth of the network, the gradients of this loss may rapidly vanish during training, limiting the supervision of the model’s deep structure [95, 96]. This could lead to the encoder structure of the model ignoring hidden force features, which are directly related to the occurrence of fiber breaks. In fact, when force is applied to fiber-reinforced composites, the fibers in the material undergo a certain amount of deformation [97], which may also be reflected in the feature maps corresponding to the fiber structure.

[98, 68] have demonstrated that progressively integrating conditions into the various feature extraction blocks of the model can effectively enhance the model’s encoding and understanding of conditions, thereby generating better quality data. One possible direction for the work in this thesis could be to incorporate the difference in tension between the input image and target image in the image pair as an additional independent condition input to the generator and discriminator. For example, a mask derived from the force difference mapping could be added after each encoder block. By repeatedly emphasizing the influence of applied tension on the extraction of various feature maps in the model, it may be possible to enable the model to understand how force affects different fiber microstructures, thereby predicting potential fiber breaks.

Other generative models In this thesis, we employ a c-GAN architecture to enhance the image synthesis capabilities of the 3D U-Net. However, the adversarial training process of GANs tends to focus on shaping the global features of the generated images [99, 98]. This suggests that models emphasizing local features may perform better, as we are particularly interested in how microstructures influence fiber breaks. Recent research has seen the adoption of diffusion models [100], which utilize a diffusion process based on physics and probability theory to transform noise into complex, realistic data samples. Compared to GANs, diffusion models benefit from their forward and reverse processes, often generating data with finer details and textures [101]. Replacing GANs with diffusion architectures could potentially supervise models to generate more detailed fiber breaks, reflecting the impact of surrounding fiber material microstructures. Another possible advantage of using diffusion models is their step-by-step denoising process, which can provide deeper insights into how the models understand local features [102] and determine the presence of fiber breaks. This distributed generation process aids in comprehending how microstructures influence the occurrence of fiber breaks.

8 Conclusion

Our custom 3D U-Net and c-GAN models outperform the baseline from [16] in synthesizing FRC CT images under increased tension from the corresponding images under initial tension. In terms of visual quality, our 3D U-Net and c-GAN both accurately predict the movement of fiber structures under higher tensile loading across all datasets. However, we observed a weak correlation between the generated and ground truth fiber breaks. Furthermore, our experiments demonstrate that the introduction of the c-GAN architecture enhances both the qualitative and quantitative performance of standalone 3D U-Nets, suggesting that adversarial training aids in improving the models’ understanding of fiber microstructures. Additionally, we find the c-GAN architecture contributes to increased stability during model training.

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisor, Abraham George Smith, for his patience and guidance throughout this project. Without his insights, support, and encouragement, I would not have been able to complete this thesis.

Secondly, I would like to thank Mahoor Mehdikhani and his research team for providing such a valuable dataset and for answering many of my questions about material science during this project, as well as for offering numerous suggestions.

I am also grateful to Sofie Sylvest Aastrup, who worked on the same datasets during the same period as I did, for her insightful discussions that greatly contributed to my understanding and progress in this project.

Lastly, I would like to thank my family and friends for their unwavering support and encouragement throughout my studies. Their belief in me has been a constant source of motivation.

References

- [1] W. Curtin, “Stochastic damage evolution and failure in fiber-reinforced composites,” *Advances in applied mechanics*, vol. 36, pp. 163–253, 1998.
- [2] Y. Swolfs, I. Verpoest, and L. Gorbatikh, “A review of input data and modelling assumptions in longitudinal strength models for unidirectional fibre-reinforced composites,” *Composite Structures*, vol. 150, pp. 153–172, 2016.
- [3] D. J. Thomas, “Complexity of understanding the failure of aerospace composite structures,” *Journal of Failure Analysis and Prevention*, vol. 16, no. 4, pp. 513–514, 2016.
- [4] A. S. H. Makhoul and M. Aliofkhazraei, *Handbook of materials failure analysis with case studies from the aerospace and automotive industries*. Butterworth-Heinemann, 2015.
- [5] G. D. Goh, V. Dikshit, A. P. Nagalingam, G. L. Goh, S. Agarwala, S. L. Sing, J. Wei, and W. Y. Yeong, “Characterization of mechanical properties and fracture mode of additively manufactured carbon fiber and glass fiber reinforced thermoplastics,” *Materials & Design*, vol. 137, pp. 79–89, 2018.
- [6] Y. Qian, Z. Li, X. Zhou, T. Xia, Y. Zhang, Z. Yang, D. Hu, and Z. Lu, “The strength prediction model of unidirectional fiber reinforced composites based on the renormalization group method,” *Composites Science and Technology*, p. 110639, 2024.
- [7] J. J. Mathavan, M. H. Hassan, and G. Franz, “Modeling of damage evaluation and failure of laminated composite materials,” in *Structural Integrity and Monitoring for Composite Materials*. Springer, 2023, pp. 27–38.
- [8] J. Vanegas-Jaramillo, A. Turon, J. Costa, L. Cruz, and J. Mayugo, “Analytical model for predicting the tensile strength of unidirectional composites based on the density of fiber breaks,” *Composites Part B: Engineering*, vol. 141, pp. 84–91, 2018.
- [9] J. D. Vanegas-Jaramillo and I. D. Patiño-Arcila, “Fragmentation model for the tensile response of unidirectional composites based on the critical number of fiber breaks and the correction of the fiber-matrix interfacial strength,” *Latin American Journal of Solids and Structures*, vol. 16, p. e217, 2019.
- [10] G. Huang and A. H. Jafari, “Enhanced balancing gan: Minority-class image generation,” *Neural computing and applications*, vol. 35, no. 7, pp. 5145–5154, 2023.
- [11] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, “Attngan: Fine-grained text to image generation with attentional generative adversarial networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1316–1324.
- [12] D. Z. Chen, Y. Siddiqui, H.-Y. Lee, S. Tulyakov, and M. Nießner, “Text2tex: Text-driven texture synthesis via diffusion models,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 18 558–18 568.
- [13] C. M. Landis, I. J. Beyerlein, and R. M. McMeeking, “Micromechanical simulation of the failure of fiber reinforced composites,” *Journal of the Mechanics and Physics of Solids*, vol. 48, no. 3, pp. 621–648, 2000.
- [14] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, “A survey of convolutional neural networks: analysis, applications, and prospects,” *IEEE transactions on neural networks and learning systems*, vol. 33, no. 12, pp. 6999–7019, 2021.
- [15] M. Krichen, “Convolutional neural networks: A survey,” *Computers*, vol. 12, no. 8, p. 151, 2023.
- [16] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.

- [17] M. Drozdal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal, “The importance of skip connections in biomedical image segmentation,” in *International workshop on deep learning in medical image analysis, international workshop on large-scale annotation of biomedical data and expert label synthesis*. Springer, 2016, pp. 179–187.
- [18] C. Williams, F. Falck, G. Deligiannidis, C. C. Holmes, A. Doucet, and S. Syed, “A unified framework for u-net design and analysis,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [19] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, “3d u-net: learning dense volumetric segmentation from sparse annotation,” in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17–21, 2016, Proceedings, Part II 19*. Springer, 2016, pp. 424–432.
- [20] H. Hwang, H. Z. U. Rehman, and S. Lee, “3d u-net for skull stripping in brain mri,” *Applied Sciences*, vol. 9, no. 3, p. 569, 2019.
- [21] Y. Wu, S. Qi, M. Wang, S. Zhao, H. Pang, J. Xu, L. Bai, and H. Ren, “Transformer-based 3d u-net for pulmonary vessel segmentation and artery-vein separation from ct images,” *Medical & Biological Engineering & Computing*, vol. 61, no. 10, pp. 2649–2663, 2023.
- [22] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3d convolutional networks,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.
- [23] P. Shi, M. Duan, L. Yang, W. Feng, L. Ding, and L. Jiang, “An improved u-net image segmentation method and its application for metallic grain size statistics,” *Materials*, vol. 15, no. 13, p. 4417, 2022.
- [24] F. Kälber, O. Köpüklü, N. H. Lehment, and G. Rigoll, “U-net based zero-hour defect inspection of electronic components and semiconductors,” in *VISIGRAPP (5: VISAPP)*, 2021, pp. 593–601.
- [25] L. Ruthotto and E. Haber, “An introduction to deep generative modeling,” *GAMM-Mitteilungen*, vol. 44, no. 2, p. e202100008, 2021.
- [26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Advances in neural information processing systems*, vol. 27, 2014.
- [27] L. Mi, M. Shen, and J. Zhang, “A probe towards understanding gan and vae models,” *arXiv preprint arXiv:1812.05676*, 2018.
- [28] L. Wei, S. Zhang, W. Gao, and Q. Tian, “Person transfer gan to bridge domain gap for person re-identification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 79–88.
- [29] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.
- [30] Y. Zhang, Z. Gan, K. Fan, Z. Chen, R. Henao, D. Shen, and L. Carin, “Adversarial feature matching for text generation,” in *International conference on machine learning*. PMLR, 2017, pp. 4006–4015.
- [31] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang, “Midinet: A convolutional generative adversarial network for symbolic-domain music generation,” *arXiv preprint arXiv:1703.10847*, 2017.
- [32] D. Torbunov, Y. Huang, H. Yu, J. Huang, S. Yoo, M. Lin, B. Viren, and Y. Ren, “Uvcgan: Unet vision transformer cycle-consistent gan for unpaired image-to-image translation,” in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2023, pp. 702–712.
- [33] Z. Wang, K. Jiang, P. Yi, Z. Han, and Z. He, “Ultra-dense gan for satellite imagery super-resolution,” *Neurocomputing*, vol. 398, pp. 328–337, 2020.

- [34] L. Ma, R. Shuai, X. Ran, W. Liu, and C. Ye, “Combining dc-gan with resnet for blood cell image classification,” *Medical & biological engineering & computing*, vol. 58, pp. 1251–1264, 2020.
- [35] Z. Yi, H. Zhang, P. Tan, and M. Gong, “Dualgan: Unsupervised dual learning for image-to-image translation,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2849–2857.
- [36] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.
- [37] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014.
- [38] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *International conference on machine learning*. PMLR, 2017, pp. 214–223.
- [39] S. Prashanth, K. Subbaya, K. Nithin, and S. Sachhidananda, “Fiber reinforced composites-a review,” *J. Mater. Sci. Eng.*, vol. 6, no. 03, pp. 2–6, 2017.
- [40] D. K. Rajak, D. D. Pagar, P. L. Menezes, and E. Linul, “Fiber-reinforced polymer composites: Manufacturing, properties, and applications,” *Polymers*, vol. 11, no. 10, p. 1667, 2019.
- [41] M. Alves, C. C. Junior, and S. K. Ha, “Fiber waviness and its effect on the mechanical performance of fiber reinforced polymer composites: An enhanced review,” *Composites Part A: Applied Science and Manufacturing*, vol. 149, p. 106526, 2021.
- [42] M. Patel, B. Pardhi, S. Chopara, and M. Pal, “Lightweight composite materials for automotive-a review,” *Carbon*, vol. 1, no. 2500, p. 151, 2018.
- [43] T. Heitkamp, S. Kuschmitz, S. Girth, J.-D. Marx, G. Klawitter, N. Walddt, and T. Vietor, “Stress-adapted fiber orientation along the principal stress directions for continuous fiber-reinforced material extrusion,” *Progress in Additive Manufacturing*, vol. 8, no. 3, pp. 541–559, 2023.
- [44] R. Talreja and A. M. Waas, “Concepts and definitions related to mechanical behavior of fiber reinforced composite materials,” *Composites Science and Technology*, vol. 217, p. 109081, 2022.
- [45] P. Zhang, W. Yao, X. Hu, and T. Q. Bui, “3d micromechanical progressive failure simulation for fiber-reinforced composites,” *Composite Structures*, vol. 249, p. 112534, 2020.
- [46] C. Breite, A. Melnikov, A. Turon, A. B. de Moraes, C. Le Boulrot, E. Maire, E. Schöberl, F. Otero, F. Mesquita, I. Sinclair *et al.*, “Detailed experimental validation and benchmarking of six models for longitudinal tensile failure of unidirectional composites,” *Composite structures*, vol. 279, p. 114828, 2022.
- [47] L. Wan, Y. Ismail, Y. Sheng, J. Ye, and D. Yang, “A review on micromechanical modelling of progressive failure in unidirectional fibre-reinforced composites,” *Composites Part C: Open Access*, vol. 10, p. 100348, 2023.
- [48] W. Weibull, “A statistical distribution function of wide applicability,” *Journal of applied mechanics*, 1951.
- [49] A. Bhaduri, A. Gupta, and L. Graham-Brady, “Stress field prediction in fiber-reinforced composite materials using a deep learning approach,” *Composites Part B: Engineering*, vol. 238, p. 109879, 2022.
- [50] A. Gupta, A. Bhaduri, and L. Graham-Brady, “Accelerated multiscale mechanics modeling in a deep learning framework,” *Mechanics of Materials*, vol. 184, p. 104709, 2023.
- [51] K. Zhou, H. Sun, R. Enos, D. Zhang, and J. Tang, “Harnessing deep learning for physics-informed prediction of composite strength with microstructural uncertainties,” *Computational Materials Science*, vol. 197, p. 110663, 2021.

- [52] Z. Yang, C.-H. Yu, K. Guo, and M. J. Buehler, “End-to-end deep learning method to predict complete strain and stress tensors for complex hierarchical composite microstructures,” *Journal of the Mechanics and Physics of Solids*, vol. 154, p. 104506, 2021.
- [53] C. Breite, A. Melnikov, A. Turon, A. de Morais, C. Le Bourlot, E. Maire, E. Schöberl, F. Otero, F. Mesquita, I. Sinclair *et al.*, “A synchrotron computed tomography dataset for validation of longitudinal tensile failure models based on fibre break and cluster development,” *Data in brief*, vol. 39, p. 107590, 2021.
- [54] R. Guo, C. Li, Y. Niu, and G. Xian, “The fatigue performances of carbon fiber reinforced polymer composites—a review,” *Journal of Materials Research and Technology*, vol. 21, pp. 4773–4789, 2022.
- [55] X. Yang, L. Yu, S. Li, H. Wen, D. Luo, C. Bian, J. Qin, D. Ni, and P.-A. Heng, “Towards automated semantic segmentation in prenatal volumetric ultrasound,” *IEEE transactions on medical imaging*, vol. 38, no. 1, pp. 180–193, 2018.
- [56] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning*. pmlr, 2015, pp. 448–456.
- [57] Y. Wang, B. Yu, L. Wang, C. Zu, D. S. Lalush, W. Lin, X. Wu, J. Zhou, D. Shen, and L. Zhou, “3d conditional generative adversarial networks for high-quality pet image estimation at low dose,” *Neuroimage*, vol. 174, pp. 550–562, 2018.
- [58] A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves *et al.*, “Conditional image generation with pixelcnn decoders,” *Advances in neural information processing systems*, vol. 29, 2016.
- [59] B. Xu, N. Wang, T. Chen, and M. Li, “Empirical evaluation of rectified activations in convolutional network,” *arXiv preprint arXiv:1505.00853*, 2015.
- [60] A. L. Maas, A. Y. Hannun, A. Y. Ng *et al.*, “Rectifier nonlinearities improve neural network acoustic models,” in *Proc. icml*, vol. 30, no. 1. Atlanta, GA, 2013, p. 3.
- [61] B. B. Avants, N. Tustison, G. Song *et al.*, “Advanced normalization tools (ants),” *Insight j*, vol. 2, no. 365, pp. 1–35, 2009.
- [62] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [63] R. Caruana, S. Lawrence, and C. Giles, “Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping,” *Advances in neural information processing systems*, vol. 13, 2000.
- [64] A. Wolny, L. Cerrone, A. Vijayan, R. Tofanelli, A. V. Barro, M. Louveaux, C. Wenzl, S. Strauss, D. Wilson-Sánchez, R. Lymbouridou *et al.*, “Accurate and versatile 3d segmentation of plant tissues at cellular resolution,” *Elife*, vol. 9, p. e57613, 2020.
- [65] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, “Unet++: A nested u-net architecture for medical image segmentation,” in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*. Springer, 2018, pp. 3–11.
- [66] L. Mishnaevsky Jr and P. Brøndsted, “Micromechanisms of damage in unidirectional fiber reinforced composites: 3d computational analysis,” *Composites Science and Technology*, vol. 69, no. 7-8, pp. 1036–1044, 2009.
- [67] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv preprint arXiv:1511.06434*, 2015.
- [68] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.

- [69] Y. Jiang, S. Chang, and Z. Wang, “Transgan: Two pure transformers can make one strong gan, and that can scale up,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 14 745–14 758, 2021.
- [70] W. Wang, Y. Sun, and S. Halgamuge, “Improving mmd-gan training with repulsive loss function,” *arXiv preprint arXiv:1812.09916*, 2018.
- [71] Z. Wang, L. Wang, S. Duan, and Y. Li, “An image denoising method based on deep residual gan,” in *Journal of Physics: Conference Series*, vol. 1550, no. 3. IOP Publishing, 2020, p. 032127.
- [72] S. K. Kugler, A. Kech, C. Cruz, and T. Osswald, “Fiber orientation predictions—a review of existing models,” *Journal of Composites Science*, vol. 4, no. 2, p. 69, 2020.
- [73] Z.-M. Huang, C.-C. Zhang, and Y.-D. Xue, “Stiffness prediction of short fiber reinforced composites,” *International Journal of Mechanical Sciences*, vol. 161, p. 105068, 2019.
- [74] S. Shah, R. S. Choudhry, and S. Mahadzir, “A new approach for strength and stiffness prediction of discontinuous fibre reinforced composites (dfc),” *Composites Part B: Engineering*, vol. 183, p. 107676, 2020.
- [75] T. O. Hodson, “Root mean square error (rmse) or mean absolute error (mae): When to use them or not,” *Geoscientific Model Development Discussions*, vol. 2022, pp. 1–10, 2022.
- [76] T. O. Hodson, T. M. Over, and S. S. Foks, “Mean squared error, deconstructed,” *Journal of Advances in Modeling Earth Systems*, vol. 13, no. 12, p. e2021MS002681, 2021.
- [77] S. V. M. Sagheer and S. N. George, “A review on medical image denoising algorithms,” *Biomedical signal processing and control*, vol. 61, p. 102036, 2020.
- [78] K. Maharana, S. Mondal, and B. Nemade, “A review: Data pre-processing and data augmentation techniques,” *Global Transitions Proceedings*, vol. 3, no. 1, pp. 91–99, 2022.
- [79] H. Thanh-Tung and T. Tran, “Catastrophic forgetting and mode collapse in gans,” in *2020 international joint conference on neural networks (ijcnn)*. IEEE, 2020, pp. 1–10.
- [80] V. Kushwaha, G. Nandi *et al.*, “Study of prevention of mode collapse in generative adversarial network (gan),” in *2020 IEEE 4th Conference on Information & Communication Technology (CICT)*. IEEE, 2020, pp. 1–6.
- [81] Y. Hong, U. Hwang, J. Yoo, and S. Yoon, “How generative adversarial networks and their variants work: An overview,” *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–43, 2019.
- [82] H. Zhang, Z. Zhang, A. Odena, and H. Lee, “Consistency regularization for generative adversarial networks,” *arXiv preprint arXiv:1910.12027*, 2019.
- [83] Y. Liu, H. Fan, X. Yuan, and J. Xiang, “Gl-gan: Adaptive global and local bilevel optimization for generative adversarial network,” *Pattern Recognition*, vol. 123, p. 108375, 2022.
- [84] H. Ku, H. Wang, N. Pattarachaiyakoo, and M. Trada, “A review on the tensile properties of natural fiber reinforced polymer composites,” *Composites Part B: Engineering*, vol. 42, no. 4, pp. 856–873, 2011.
- [85] H. Huang, L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto, X. Han, Y.-W. Chen, and J. Wu, “Unet 3+: A full-scale connected unet for medical image segmentation,” in *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2020, pp. 1055–1059.
- [86] X. Hu, L. Chu, J. Pei, W. Liu, and J. Bian, “Model complexity of deep learning: A survey,” *Knowledge and Information Systems*, vol. 63, pp. 2585–2619, 2021.
- [87] A. Brock, J. Donahue, and K. Simonyan, “Large scale gan training for high fidelity natural image synthesis,” *arXiv preprint arXiv:1809.11096*, 2018.

- [88] J. M. Johnson and T. M. Khoshgoftaar, “Survey on deep learning with class imbalance,” *Journal of Big Data*, vol. 6, no. 1, pp. 1–54, 2019.
- [89] K. Ghosh, C. Bellinger, R. Corizzo, P. Branco, B. Krawczyk, and N. Japkowicz, “The class imbalance problem in deep learning,” *Machine Learning*, pp. 1–57, 2022.
- [90] X.-Y. Liu, J. Wu, and Z.-H. Zhou, “Exploratory undersampling for class-imbalance learning,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 2, pp. 539–550, 2008.
- [91] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso, “Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations,” in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 14, Proceedings 3*. Springer, 2017, pp. 240–248.
- [92] Y. Y. T. L. L. D. T. A. T. R. Brosch, T., “Deep convolutional encoder networks for multiple sclerosis lesion segmentation.” *Medical Image Computing and Computer*, vol. 9351, 2015.
- [93] A. Badran, D. Marshall, Z. Legault, R. Makovetsky, B. Provencher, N. Piché, and M. Marsh, “Automated segmentation of computed tomography images of fiber-reinforced composites by deep learning,” *Journal of Materials Science*, vol. 55, pp. 16 273–16 289, 2020.
- [94] R. A. Kopp, “X-ray micro-computed tomography and deep learning segmentation of progressive damage in hierarchical nanoengineered carbon fiber composites,” Ph.D. dissertation, Massachusetts Institute of Technology, 2021.
- [95] L. Borawar and R. Kaur, “Resnet: Solving vanishing gradient in deep networks,” in *Proceedings of International Conference on Recent Trends in Computing: ICRTC 2022*. Springer, 2023, pp. 235–247.
- [96] S. Hochreiter, “The vanishing gradient problem during learning recurrent neural nets and problem solutions,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.
- [97] M. Northolt and J. Baltussen, “The tensile and compressive deformation of polymer and carbon fibers,” *Journal of applied polymer science*, vol. 83, no. 3, pp. 508–538, 2002.
- [98] L. Zhang, A. Rao, and M. Agrawala, “Adding conditional control to text-to-image diffusion models,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3836–3847.
- [99] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel *et al.*, “Scaling rectified flow transformers for high-resolution image synthesis,” *arXiv preprint arXiv:2403.03206*, 2024.
- [100] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui, and M.-H. Yang, “Diffusion models: A comprehensive survey of methods and applications,” *ACM Computing Surveys*, vol. 56, no. 4, pp. 1–39, 2023.
- [101] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.
- [102] B. Chen, W. Tan, Y. Wang, and G. Zhao, “Distinguishing between natural and gan-generated face images by combining global and local features,” *Chinese Journal of Electronics*, vol. 31, no. 1, pp. 59–67, 2022.