

Faculdade de Engenharia da Universidade do Porto

Enhancing Lung CT 3D Image Synthesis with Generative Adversarial Networks

João Afonso Martins Domingues Andrade



Master in Informatics and Computing Engineering

Supervisor: Helder Filipe Pinto de Oliveira, PhD

Co-supervisor: Tânia Maria Pereira Lopes, PhD

Tutor: Tiago Filipe Sousa Gonçalves, MSc

Tutor: Pedro Fernandes Sousa, MSc

July 26, 2024

Enhancing Lung CT 3D Image Synthesis with Generative Adversarial Networks

João Afonso Martins Domingues Andrade

Master in Informatics and Computing Engineering

Approved by ...:

President: Ana Paula Cunha da Rocha

Referee: António Manuel Trigueiros da Silva Cunha

July 26, 2024

Resumo

O cancro do pulmão é uma das doenças mais graves e de rápida propagação, afectando milhões de pessoas em todo o mundo todos os anos. É conhecido por ser um tipo de cancro agressivo, especialmente nas fases iniciais da doença, o que torna o seu tratamento difícil. Devido a estes factos alarmantes, há uma necessidade urgente de melhorar e facilitar os tratamentos existentes para esta doença. Um dos problemas mais preocupantes é a falta de informação a que os médicos podem aceder relativamente às imagens de tomografia computadorizada (TC) do pulmão.

Para contrariar este problema, as comunidades clínicas e de investigação estão agora a procurar novas alternativas, como a inteligência artificial (IA), que podem ajudar significativamente os médicos e os doentes. Estes modelos de IA são utilizados para criar ferramentas de diagnóstico melhores e mais eficientes. No entanto, para serem úteis e exactos, precisam de muitos dados. Este é o foco principal deste trabalho, que tem como objetivo encontrar uma forma de gerar esses dados de forma eficiente e sem falhas.

Os investigadores desenvolveram agora novos modelos generativos designados por redes adversárias generativas (GANs) que lhes permitem sintetizar imagens de TC pulmonar volumétricas 3D de alta resolução com características realistas e coerência espacial. Entre estes modelos, foram escolhidos 3 para serem estudados neste trabalho: WGAN, α -WGAN e HA-GAN. Destes 3, WGAN e α -WGAN foram avaliados na geração de imagens de tamanho 64 e 128, e HA-GAN em imagens de tamanho 128.

Estas imagens geradas foram depois avaliadas com base em diferentes métricas de avaliação (MS-SSIM, que mede a semelhança estrutural entre as imagens, MMD, que mede a diferença entre as distribuições das imagens geradas e reais, e FID, que calcula a distância entre os vectores de características gerados e reais) e também por profissionais médicos através de um teste de turing visual (VTT), que consiste em fazer com que os seres humanos respondam a perguntas para testar se um sistema consegue reconhecer objectos em imagens e compreender as suas características e ligações. Este teste é considerado bem sucedido se os humanos não conseguirem distinguir entre os resultados falsos do sistema e os reais.

Este trabalho baseia-se nestes modelos desenvolvidos anteriormente e tem como objetivo melhorar a arquitetura global do conjunto de modelos. Apesar de existirem algumas digitalizações irrealistas pelo meio, os 3 modelos conseguiram sintetizar digitalizações espacialmente coerentes e realistas para ambas as resoluções. Dos 3 modelos, o que resultou em imagens geradas com maior qualidade foi o HA-GAN, que também teve os melhores resultados globais em termos de métricas. Relativamente ao VTT, mostrou algumas limitações, como a resolução dos exames gerados, o que resultou em alguns resultados inconclusivos. Embora os resultados não tenham sido perfeitos, os modelos mostraram um forte potencial de melhoria.

Abstract

Lung cancer is one of the most serious and rapidly spreading diseases, affecting millions of people worldwide each year. It is known for being an aggressive type of cancer, especially in the early stages of the disease, making it challenging to treat. Due to these alarming facts, there is an urgent need to improve and facilitate the existing treatments for this disease. One of the more concerning problems is the lack of information doctors can access regarding lung computed tomography (CT) images.

To counter this problem, clinical and research communities are now looking at new alternatives like artificial intelligence (AI), which might help doctors and patients significantly. These AI models are used to create better and more efficient diagnostic tools. However, for them to be useful and accurate they need a lot of data. This is the main focus of this work, having as an objective to find a way to efficiently and faultlessly generate this data.

Researchers have now developed new generative models called generative adversarial networks (GANs) that allow them to synthesise high-resolution 3D volumetric lung CT images with realistic characteristics and spatial coherence. Among these models, 3 were chosen to be studied in this work: WGAN, α -WGAN, and HA-GAN. From these 3, WGAN and α -WGAN were evaluated on the generation of size 64 and 128 images, and HA-GAN on size 128 images.

These generated images were then evaluated on different evaluation metrics (MS-SSIM which measures the structural similarity between images, MMD which measures how the distributions of the generated and real images differ, and FID which calculates the distance between the generated and real feature vectors) and also by medical professionals with a visual turing test (VTT) which involves getting humans to answer questions to test if a system can recognise objects in images and understand their features and connections. This test is considered successful if humans cannot distinguish between the fake outputs from the system and the real ones.

This work builds upon these previously developed models and aims to improve the overall architecture of the model set. Despite having some unrealistic scans in between, the 3 models were able to synthesise spatially coherent and realistic scans for both resolutions. Out of the 3 models, the one that resulted in the highest quality generated images was the HA-GAN which also had the best overall results in terms of metrics. Regarding the VTT, it showed some limitations, such as the resolution of the generated scans, resulting in some inconclusive results. Even though the results were not perfect, the models showed strong potential for improvement.

Agradecimentos

Firstly, I want to start by expressing my gratitude to my supervisors and co-supervisors, Hélder Oliveira, Tânia Pereira, Tiago Gonçalves, and Pedro Sousa, for all of the challenges, encouragement, and insightful suggestions. My skills have improved significantly over the past year, and my enthusiasm for this research area has only increased. Thanks for always being there whenever I needed help, no matter how stupid my doubts were.

To my whole family which has always believed in me, supported me, and did their best to make sure that I always had everything and more. You did more than I could ever ask for, and it wouldn't have been possible without you. I want to give my eternal gratitude to my mother, father, and grandmother, who have a very special place in my heart and have raised me with the best principles. I wouldn't be the man I am today, or even be writing this if it wasn't for you. Thank you. Lastly, I want to show my appreciation for my grandfather, who sadly, is not here to witness this important milestone in my life, but I know he is somewhere out there watching it. I hope you're proud of me.

To all the amazing people I have met during these amazing 5 years. You have all marked me for life and it has been a pleasure to be able to know you. There is no space to enumerate all the people with whom I spent, until now, the best 5 years of my life but there are some that I cannot leave behind. So a special thanks to Sérgio Estêvão, André Santos, Maria Carneiro, Edgar Torre, Rodrigo Tuna, and Sofia Germer. You were always there in the darkest moments, but also in the brightest. Thank you for sharing this stage with me. It was easier with you.

To all my other friends, the ones not from University. Thank you for being present in all the important moments in my life, and for the many years of happiness that we have spent together. I'd also like to give a special thanks to some of my closest friends, that have supported me throughout many years. Thank you João Lopes, Ana Viterbo, Ana Cunha, Rui Silva, Luís Alves, Luís Busto, Jónatas, Gonçalo Roncha, Inês Silva, Gonçalo Barros, Beatriz Sousa, António Cardoso, Carolina Soares, Franciso Silva, Inês Teixeira, Joana Teixeira, José Picão, Raquel Pereira, Hugo Castro, Hugo Bastos, Clara Sousa, João Guimarães, and Bia Cunha. Lastly to all the people I met during ERASMUS that, even though are more recent, still have a place in my heart. Thank you Yara Alves, Carlos Ferreira, Miguel Silveira, Guilian, Manon, Flavie, Caroline, Giulia, Valentina, Misia, Georgia, Isabel Lobo, Vanessa, etc.

This project has received funding from the European Union's Horizon Europe research and innovation programme under the Grant Agreement 101095384.

*“Anni, amori e bicchieri di vino.
Non se contano mai”*

Anthony Capella

*“Carpe diem. Seize the day, boys.
Make your lives extraordinary”*

Dead Poets Society

Contents

1	Introduction	1
1.1	Context	1
1.1.1	Lung Cancer	1
1.1.2	Imaging Techniques for Lung Cancer	2
1.2	Motivation	3
1.3	Objectives	4
1.4	Contributions	4
1.5	Sustainable Development Goals	4
1.5.1	Goal 3: Good Health and Well-being	5
1.5.2	Goal 8: Decent Work and Economic Growth	6
1.5.3	Goal 9: Industry, Innovation, and Infrastructure	6
1.6	Document Structure	6
2	Medical Background	8
2.1	Overview about Cancer	8
2.2	Lung Anatomy	9
2.3	Lung Imaging	9
2.4	Lung Cancer	11
2.4.1	Types of Lung Cancer	11
2.4.2	Lung Cancer Statistics	12
2.4.3	Lung Cancer Survival Rate	12
2.5	Summary	13
3	Literature Review	15
3.1	Introduction	15
3.2	Variational Auto-Encoders	16
3.3	Autoregressive Generative Models	17
3.4	Generative Adversarial Networks	19
3.4.1	Conditional GAN	21
3.4.2	Image-to-image Translation	22
3.4.3	Deep Convolutional GAN	25
3.4.4	Progressive Growing GAN	27
3.4.5	Wasserstein GAN	29
3.4.6	α -WGAN	30
3.4.7	Style GAN	31
3.4.8	3D-GANs	32
3.4.9	3D-GANs for Medical Imaging	35
3.5	Normalising Flows	41

3.5.1	Context	41
3.5.2	Introduction	42
3.5.3	Generative Flows - Glow	43
3.6	Diffusion Models for 3D Medical Imaging	44
3.6.1	DPMs	45
3.6.2	Conditional Diffusion Probabilistic Model	45
3.6.3	Sequence-Aware Diffusion Model	47
3.6.4	MedGen3D	47
3.6.5	3D medical image generation with DPMs	49
3.7	MedSyn	51
3.7.1	Medical BERT	52
3.7.2	Text-Conditioned Volume Generation	53
3.7.3	Joint Generative Modelling	53
3.7.4	Anatomically Controllable Synthesis	53
3.7.5	Efficient 3D Attention U-Net	53
3.7.6	Datasets and Results	54
3.8	Evaluation Metrics	54
3.8.1	Inception Score	54
3.8.2	Mode Score	55
3.8.3	Fréchet Inception Distance	55
3.8.4	Structural Similarity	55
3.8.5	Multi-Scale Structural Similarity	56
3.8.6	Maximum-Mean Discrepancy Score	56
3.8.7	Peak Signal-To-Noise Ratio	56
3.8.8	Normalised Root Mean Square Error	57
3.8.9	Learned Perceptual Image Patch Similarity	57
3.8.10	Visual Turing Test	57
3.9	Summary	57
4	3D Lung Image Generation	59
4.1	Dataset	59
4.1.1	LIDC-IDRI	59
4.2	Hardware	62
4.3	Architectures	62
4.3.1	WGAN	63
4.3.2	α -WGAN	66
4.3.3	HA-GAN	68
5	Results	73
5.1	Overview	73
5.2	Visual Turing Test	74
6	Conclusion and Future Work	76
6.1	Overview	76
6.2	Future Work	77
	References	78

List of Figures

1.1	The 17 SDGs.	5
2.1	Anatomy of the Lungs	10
2.2	A cartoon depiction of a typical CT scanner	11
2.3	CT lung scan in a 60-year-old male smoker	11
2.4	One-year cancer survival rates	13
2.5	Five-year cancer survival rates	14
3.1	Generative Models Types	16
3.2	Autoencoder architecture	16
3.3	A Graphical Representation of VAE	17
3.4	Autoregressive Row LSTM architecture	18
3.5	Diagonal BiLSTM	18
3.6	Visualisation of the PixelCNN	19
3.7	Generative Adversarial Networks	20
3.8	Conditional Adversarial Net	21
3.9	Comparison between samples from an AC-GAN and state-of-the-art model	22
3.10	Skip connections impact on encoder-decoder outcomes	23
3.11	Variation of patch sizes on PatchGAN	24
3.12	Examples of the results of Pix2Pix on different scenarios	24
3.13	Cycle GAN architecture	24
3.14	Cycle consistency loss behaviour	26
3.15	DCGAN generator architecture	27
3.16	Training process of PGGANs	27
3.17	1024 x 1024 photos produced with the CELEBA-HQ collection	28
3.18	Comparison between WGAN and standard GAN outputs	29
3.19	WGAN algorithm where the G and D are DCGANs	30
3.20	Slices of normal brain 3D samples from α -WGAN	31
3.21	Comparison between traditional generators and Style GAN generator	32
3.22	Results of the application of noise inputs at different layers	33
3.23	Examples of stochastic variation	34
3.24	The generator in 3D-GAN	34
3.25	Objects created from vectors by 3D-GANs without the need for a reference object or image	35
3.26	Examples of single image 3D results using the IKEA dataset	35
3.27	Detailed design of the suggested model	36
3.28	Samples of normal brain MRI, both real and created	37
3.29	CCE-GAN Architecture	37

3.30	Examples of Real, VAE-GAN generated, and CCE-GAN generated brains .	38
3.31	Architecture of HA-GAN	38
3.32	Inference with the hierarchical generator and encoder	40
3.33	Real photos and random images produced by various models	40
3.34	Architecture of the styleGAN3D generator	41
3.35	Glow architecture	43
3.36	LSUN Church and Bedroom samples	45
3.37	Architecture of cDPM	46
3.38	Three MRI images produced by seven models	46
3.39	Overview of the SADMM model	47
3.40	Qualitative comparison between baselines	48
3.41	Overview of the proposed MedGen3D	48
3.42	Qualitative comparison between different generation models and visualisation of synthetic 3D brain MRI slices	49
3.43	Generated data samples from two public datasets that were utilised in the training of the diffusion model	50
3.44	The effects of different compression factors in the reconstruction quality for two distinct samples using the VQ-GAN autoencoder	51
3.45	Overview of the generative model	52
3.46	Effective low-resolution generative model with input of clinical tokens . . .	52
3.47	Real and generated lung images from HA-GAN, MedSyn, Medical Diffusion	54
4.1	Examples of lesions that satisfy the LIDC/IDRI definition	60
4.2	Examples of slices and corresponding masks generated	61
4.3	Generated images using the WGAN architecture	64
4.4	Unrealistic samples generated by WGAN	65
4.5	Mode collapse using the WGAN architecture for $64 \times 64 \times 64$ images	66
4.6	Generated images using the α -WGAN architecture	67
4.7	Unrealistic samples generated by α -WGAN	68
4.8	Generated images using the HA-WGAN architecture	71
4.9	Unrealistic samples generated by HA-GAN	72

List of Tables

4.1	Minimum, maximum, and average number of slices per patient	62
4.2	Average quantitative results for WGAN.	65
4.3	Average quantitative results for α -WGAN.	68
4.4	Average quantitative results for HA-GAN.	70
5.1	Metric comparison for $64 \times 64 \times 64$ images.	73
5.2	Metric comparison for $128 \times 128 \times 128$ images.	74
5.3	Confusion Matrix and Performance Metrics for the Radiologist	75

Abbreviations

3D-CNN	3D-Convolutional Neural Network
3D-GAN	3D GANs
ACDC	Automated Cardiac Diagnosis Challenge
AC-GAN	Auxiliary Classifier GAN
AdaIN	Adaptive Instance Normalisation
ADAM	Adaptive Moment Estimation
AdamW	Adam with Weight Decay
AI	Artificial Intelligence
α -WGAN	Alpha-WGAN
BiLSTM	Bidirectional LSTM
CAD	Computer-Aided Diagnostic
CAT	Computerised Axial Tomography
CCE-GAN	Cycle Consistent Embedding GAN
CD	Code Discriminator
cDPMs	Conditional Diffusion Probabilistic Models
CGANs	Conditional GANs
CT	Computed Tomography
D	Discriminator
DCGAN	Deep Convolutional GAN
DCNNs	Deep Convolutional Neural Networks
Diagonal BiLSTM	Diagonal Bidirectional Long Short-Term Memory
DICOM	Digital Imaging and Communications in Medicine
DPMs	Diffusion Probabilistic Models
E	Encoder
EM	Earth Mover
FID	Fréchet Inception Distance
FOR	False Omission Rate
G	Generator
GAN	Generative Adversarial Network
HA-GAN	Hierarchical Amortized GAN
HIPAA	Health Insurance Portability and Accountability Act
HU	Hounsfield units
IS	Inception Score
JS	Jensen-Shannon
KL	Kullback-Leibler
LPIPS	Learned Perceptual Image Patch Similarity

LSUN	Large-Scale Scene Understanding
LSTM	Long Short-Term Memory
MC-DPM	Multi-Condition Diffusion Probabilistic Model
MMD	Maximum Mean Discrepancy
MRI	Magnetic Resonance Imaging
MS-SSIM	Multi-Scale Structural Similarity
NF	Normalising Flows
NICE	Non-linear Independent Components Estimation
NMRSE	Normalised Root Mean Square Error
NN	Neural Network
NPV	Negative Predicted Value
NSCLC	Non-Small Cell Lung Cancer
PGGANs	Progressive Growing GANs
PixelCNN	Pixel Convolutional Neural Network
PixelRNNs	Pixel Recurrent Neural Networks
PixelSNAIL	Parallelized Pixel Recurrent Neural Network
PSNR	Peak Signal-To-Noise Ratio
RealNVP	Real-valued Non-Volume Preserving
RELU	Rectified Linear Unit
RNNs	Recurrent Neural Networks
Row LSTM	Row Long Short-Term Memory
SADM	Sequence-Aware Diffusion Model
SDGs	Sustainable Development Goals
SDM	Semantic Diffusion Refiner
SE	Sub-Encoder
SCLC	Small Cell Lung Cancer
Tanh	Hyperbolic Tangent
TV	Total Variation
VAE	Variational Auto Encoders
VQVAE	Vector Quantised Variational Autoencoder
WGAN	Wasserstein GAN
WGAN-GP	Wasserstein GAN with Gradient Penalty

Chapter 1

Introduction

Contents

1.1	Context	1
1.1.1	Lung Cancer	1
1.1.2	Imaging Techniques for Lung Cancer	2
1.2	Motivation	3
1.3	Objectives	4
1.4	Contributions	4
1.5	Sustainable Development Goals	4
1.5.1	Goal 3: Good Health and Well-being	5
1.5.2	Goal 8: Decent Work and Economic Growth	6
1.5.3	Goal 9: Industry, Innovation, and Infrastructure	6
1.6	Document Structure	6

1.1 Context

The prevalence and impact of lung cancer underscore the necessity of advancing diagnostic and treatment methodologies. Lung cancer remains one of the most common and lethal cancers globally, characterised by late-stage diagnosis and poor survival rates [10].

1.1.1 Lung Cancer

Lung cancer research holds paramount importance due to its significant mortality rate and the challenges associated with early detection. While other cancers like breast, prostate, and colorectal cancer have seen improvements in early detection and survival rates, lung cancer continues to be diagnosed at advanced stages, leading to limited treatment options and lower survival prospects. Consequently, there is a pressing need for enhanced screening methods, innovative treatment strategies, and a comprehensive understanding of lung cancer biology [7].

Cancer is defined by the uncontrolled proliferation of cells and their potential to spread to other body parts. The global burden of cancer is concerning, especially the disparities in access to quality healthcare and the resulting differences in survival rates between developed and developing nations.

Lung cancer is categorised into two primary types: Non-Small Cell Lung Cancer (NSCLC) and Small Cell Lung Cancer (SCLC). NSCLC, which includes sub-types such as adenocarcinoma, squamous cell carcinoma, and large cell carcinoma, accounts for 80-85% of lung cancer cases [11]. SCLC, known for its rapid growth and spread, represents 10-15% of cases. In 2020, lung cancer was one of the most prevalent cancers globally, with significant incidence rates among both men and women. It also has one of the lowest survival rates among major cancers, primarily due to late-stage diagnosis and the aggressive nature of the disease [12]. The five-year survival rate remains low, emphasising the need for early detection [13]. Only about 16% of lung cancer cases are diagnosed at an early stage, significantly impacting treatment success and overall survival. The high mortality rate is compounded by the difficulty in early detection and the limited effectiveness of current treatments.

1.1.2 Imaging Techniques for Lung Cancer

Computed tomography (CT) scans are pivotal for the early detection and management of lung cancer. Advancements in CT technology, such as helical and multi-detector CT scanners, provide high-resolution, three-dimensional images of the lungs. These techniques are vital for identifying pulmonary nodules and other abnormalities, facilitating early diagnosis, and improving treatment outcomes [3].

Improving imaging techniques and early detection methods is crucial for better patient outcomes and reducing mortality rates. A deeper understanding of lung cancer statistics and survival trends can inform more effective screening programs and resource allocation in healthcare systems. By focusing on these areas, advancements can be made in both clinical practice and public health policy to combat this deadly disease.

One technology that has been created to help medical personnel enhance diagnosis is computer-aided diagnostic (CAD) systems. As a result, a lot of work has gone into creating these systems to accurately recognise and describe lung diseases in CT scans. Hussein et al. [49] classified lung nodules with 91.2% accuracy using a 3D Convolutional Neural Network (3D-CNN). But just 1018 CT images were included in the dataset, which is insufficient for deep learning models.

However, access to medical data is often restricted, making it extremely difficult to get. This is highlighted in the instance of lung imaging data, where specialists' annotations of the data are required to identify and categorise diseases, yet doing so can be costly and time-consuming.

Because generative models can synthesise new data, they have been utilised as a data augmentation strategy to address the issue of insufficient data [92]. There are many

types of generative models used for image synthesis: Autoregressive models, Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), Normalising Flows, and Diffusion Models and each has its pros and cons.

- **Autoregressive:** High-quality but inefficient to train;
- **VAE:** Easy to train but produces blurry outputs;
- **GAN:** High-quality but difficult to train;
- **Normalising Flows:** Exact likelihoods but computationally expensive;
- **Diffusion Models:** High sample quality but slow sampling process.

In medical imaging, most studies focus on generating 2D scan slices. However, 3D data, like lung CT scans, are often needed for accurate analysis.

Overall, while advancements have been made, there is still much room for growth in 3D medical imaging synthesis.

1.2 Motivation

A critical issue lies in the lack of access that doctors and healthcare professionals face regarding medical images and diagnostic tools. This limitation is a pivotal motivation behind this dissertation, with the primary aim being the use of AI, which might help doctors and patients significantly. Through the intricate capabilities of machine learning models, we seek to generate high-quality and efficient diagnostic tools. However, for them to be useful and accurate they need a lot of data. This is the main focus of this work, having as an objective to find a way to efficiently and faultlessly generate this data.

The profound impact of this approach extends beyond mere accessibility, as these artificially generated medical images hold immense potential to revolutionise the landscape of disease detection, treatment planning, and diagnostic accuracy. The inherent quality of these generated images provides a unique advantage in enabling health professionals to discern and comprehend intricate details related to various medical conditions, particularly in the context of lung health.

As health professionals delve into the study of these meticulously crafted images, the disease identification process becomes more streamlined. The heightened quality not only facilitates the early detection of diseases but also empowers practitioners to make more informed decisions regarding treatment modalities. This, in turn, contributes to an elevated standard of care by enabling healthcare providers to tailor treatments more precisely to individual patient needs.

In essence, the core motivation behind this dissertation is to harness the capabilities of AI to break down the barriers of medical diagnosis tools and ultimately pave the way for more effective and personalised healthcare interventions [49].

1.3 Objectives

The main objective of this dissertation is to advance the field of 3D image generation using machine learning, particularly by adapting generative models that have already been successfully used in this field, to the generation of 3D lung CT images. The overarching objectives can be broken down into several key components:

- **Generative Models Research and Application:** This involves exploring and using previously state-of-the-art models;
- **Evaluation of Model’s Difficulties:** Addressing challenges in 3D image generation, such as data variability, limited dataset sizes, computational power, and complex representation of intricate anatomical structures;
- **Validation with Real Data:** Validation of the model’s performance using real-world data, specifically 3D lung images, to ensure accurate representation of lung structures and abnormalities, using quantitative metrics;
- **Involvement of Health Professionals:** Collaborate with healthcare professionals, especially those with expertise in interpreting lung X-rays, to obtain valuable insights and feedback. This step is very important, especially for conducting Visual Turing Tests (VTTs), which further help validate the clinical relevance and accuracy of the generated 3D images.

1.4 Contributions

The contributions of this thesis were the following:

- Examination of the existing limitations on the synthesis of 3D medical data;
- Adaptation and analysis of GAN architectures for generation of 3D lung CT scans;
- Comprehensive evaluation of these model’s performance with quantitative and qualitative assessment of the generated images with metrics such as MMD, 3D-MS-SSIM, and FID, as well as a VTT with a radiologist.

1.5 Sustainable Development Goals

The United Nations approved the Sustainable Development Goals (SDGs)¹ in 2015, and they consist of 17 objectives with precise targets that must be met by 2030. Among many other global issues, the project addresses poverty, hunger, healthcare, and climate change. Each of the seventeen goals is shown in figure 1.1. This dissertation is integrated

¹<https://sdgs.un.org/goals>

into three of them: Goal 3 - Good Health and Well-being, Goal 8 - Decent Work and Economic Growth, and Goal 9 - Industry, Innovation, and Infrastructure ².



Figure 1.1: The 17 SDGs.

1.5.1 Goal 3: Good Health and Well-being

Goal 3 has as an objective to ensure healthy lives and promote well-being for all at all ages.

High-quality medical imaging produced by AI immediately improves diagnostic accuracy, which is important for the early detection of disorders, especially those related to the lungs. Patient survival rates and treatment outcomes are greatly enhanced by this early identification. Additionally, a significant barrier in many healthcare settings is addressed by democratising access to medical imaging using AI-generated images, guaranteeing that more healthcare practitioners, especially in locations with few resources, have access to crucial diagnostic tools. AI also produces high-quality images that allow for more in-depth analysis, which helps medical professionals better customise therapies to meet the demands of each unique patient. Patient's well-being is promoted by this individualised approach, which not only raises the bar for overall care but also makes sure that each patient's therapies are tailored to their particular medical needs.

²The content of this publication has not been approved by the United Nations and does not reflect the views of the United Nations or its officials or Member States.

1.5.2 Goal 8: Decent Work and Economic Growth

Goal 8 has as an objective to promote sustained, inclusive, and sustainable economic growth, full and productive employment, and decent work for all.

The integration of AI into healthcare is encouraged by this study since it has the potential to generate new employment possibilities in the healthcare and AI industries. As AI technologies advance and are used, there will be a greater need for qualified individuals to oversee, maintain, and develop these systems, which will result in the creation of jobs. Furthermore, through the reduction of medical imaging expenses, AI-driven solutions can greatly improve the economic efficiency in the healthcare industry. This is accomplished by reducing the frequency of repeat scans and accelerating the diagnosis process, both of which reduce healthcare expenses and increase the affordability of services. Better patient outcomes are also brought about by enhanced diagnostic and treatment capabilities, which lower the financial burden of illness. Because they are more productive and use fewer healthcare resources overall, healthier populations also benefit the economy as a whole, which promotes economic growth and stability.

1.5.3 Goal 9: Industry, Innovation, and Infrastructure

Goal 9 has as an objective to build resilient infrastructure, promote inclusive and sustainable industrialisation, and foster innovation.

This study makes use of AI to stimulate innovation in the medical imaging sector, leading to technical breakthroughs that improve the efficacy and resilience of the healthcare system. Therefore it contributes to the creation of resilient healthcare systems that are better equipped to manage the growing number of patients and complicated medical conditions by enhancing medical imaging capabilities. Furthermore, especially in underserved or distant places, AI-generated medical images assist in closing gaps in access to high-quality medical diagnostics, fostering inclusivity in healthcare access and guaranteeing that technological improvements benefit a wider audience. The application of AI to medical imaging is also driving expansion in associated sectors, including data analysis, software development, and the production of medical equipment. This industrial expansion propels economic growth and fosters additional innovation in several industries, making the economy more robust and dynamic.

1.6 Document Structure

Besides this Introduction, this dissertation contains 4 more chapters. Chapter 2 provides an overview of what cancer is, specifically lung cancer. It also delves into the anatomy of the lungs, how they work, and the specifications of CT scans. Chapter 3 introduces the primary generative models. These models are ordered so that the more complex ones are the latest. They represent the most studied generative models and also some

new promising alternatives. Chapter 4 introduces the proposed implementations, and Chapter 5 the results of the models used to generate the 3D lung CT images. Lastly, chapter 6 outlines the key findings from the research conducted for this thesis and makes recommendations for more, future research.

Chapter 2

Medical Background

Contents

2.1	Overview about Cancer	8
2.2	Lung Anatomy	9
2.3	Lung Imaging	9
2.4	Lung Cancer	11
2.4.1	Types of Lung Cancer	11
2.4.2	Lung Cancer Statistics	12
2.4.3	Lung Cancer Survival Rate	12
2.5	Summary	13

2.1 Overview about Cancer

Cancer is a condition where specific cells in the body proliferate uncontrollably and extend to other areas of the body. It can originate in various parts of the human body, which comprises an immense number of cells [10].

Cell division is the process by which human cells typically divide and proliferate to create new cells in response to the needs of the body. Cells expire as they reach the end of their life cycle or are damaged, and new cells replace them [10].

This organised progression sometimes falters, leading to irregular or damaged cells multiplying and growing inappropriately [10]. These anomalous cells have the potential to aggregate and form clusters of tissue called tumours. Such growths may be either malignant, meaning they can infiltrate nearby tissues and migrate to distant sites in the body (a phenomenon termed metastasis), or benign, signifying they do not invade surrounding tissues [10].

There were an estimated 9.6 million deaths from cancer in 2018, making it the second most common cause of death globally. Men are most usually diagnosed with lung, prostate,

colorectal, stomach, and liver cancers, while women are most commonly diagnosed with breast, colorectal, lung, cervical, and thyroid cancers [7].

Globally, the toll from cancer is still rising, placing a great deal of physical, psychological, and financial strain on individuals, families, communities, and healthcare systems. A large percentage of cancer patients worldwide lack prompt access to high-quality diagnosis and treatment, and many healthcare systems in low- to middle-income countries are unprepared to handle this load. Strong healthcare systems in developed nations have seen an increase in the survival rates of many cancer types due to readily available early identification, first-rate treatment, and supportive services for survivors [7].

2.2 Lung Anatomy

Our lungs, vital for respiration, are two sponge-like organs in the chest. The heart and other structures in the mediastinum divide them from one another as they are positioned in the thoracic cavity. Each lung has three borders: anterior, posterior, and inferior. The base of each lung rests on the diaphragm, while the apex extends above the clavicle. The smaller mediastinal surface faces inward, whereas the coastal surface presses on the rib cage. The lungs' root is formed by the hilum, a slit that allows bronchi, blood vessels, lymphatic vessels, and nerves to enter [6, 11].

The left lung, which is smaller because of the presence of the heart, has two lobes, whereas the larger and heavier right lung is divided into three. Air enters the mouth or nose during inhalation and passes into the trachea, or windpipe, of the lungs. The trachea splits into bronchi, then split into bronchioles, which terminate in alveoli, which are tiny air sacs [6, 11].

The alveoli play a crucial role in respiration by absorbing oxygen into the blood and removing carbon dioxide during exhalation. These functions are the primary roles of the lungs. Lung cancers often originate in cells lining the bronchi, bronchioles, or alveoli. A protective layer surrounds the lungs called the pleura, allowing them to slide against the chest wall during breathing [6, 11].

Beneath the lungs, the diaphragm, a thin, dome-shaped muscle, separates the chest from the abdomen. As you breathe, the diaphragm moves up and down, facilitating the inhalation and exhalation of air. Together, these elements form the intricate respiratory system, emphasising the lungs' vital functions in oxygen absorption and carbon dioxide elimination [6, 11]. See Figure 2.1 for the anatomy of the lungs.

2.3 Lung Imaging

CT is a noninvasive imaging procedure that utilises special X-ray equipment to produce detailed cross-sectional images of the body. Also known as computerised tomography or

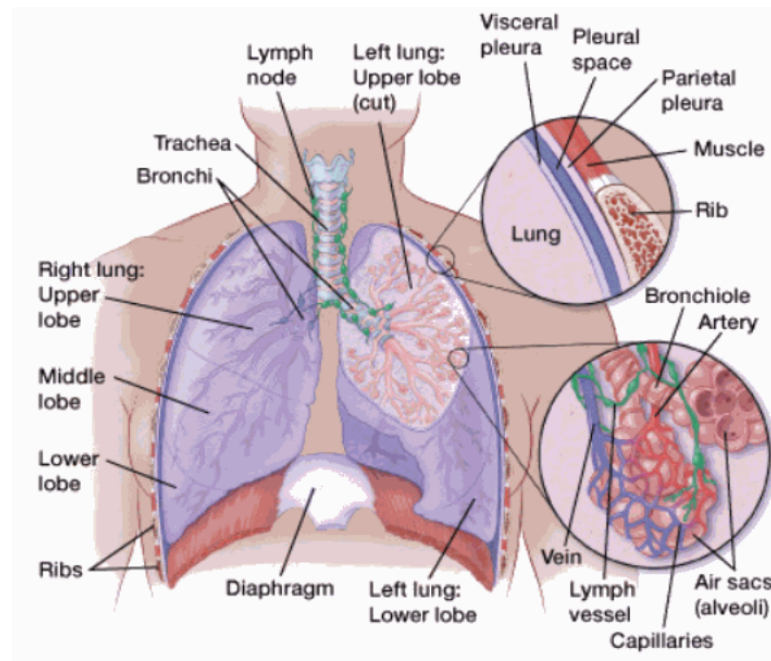


Figure 2.1: Anatomy of the lungs, (from [11]).

computerised axial tomography (CAT), it creates slices or sections of the body, akin to slices in a loaf of bread [3].

CT scans are advanced imaging tools that create detailed cross-sectional images of internal structures, aiding diagnostics, treatment guidance, and condition monitoring. They are used for various medical indications, including traumatic injuries, degenerative conditions, post-operative assessments, neoplastic conditions, and infectious processes. Additionally, CT scans assist in image-guided procedures and evaluation of bony alignment abnormalities and spinal cord-related issues. In essence, CT scans are integral for precise medical assessments and interventions [3]. See Figure 2.2 for the structure of a CT scanner.

Modern CT machines utilise helical (spiral) imaging, capturing continuous pictures instead of individual slices (see Figure 2.3). Helical CT offers advantages like speed, improved 3D imaging, and better detection of small abnormalities. Multi-slice or multi-detector CT scanners, the latest technology, allow quicker imaging of more slices [3].

Digital geometry processing generates 3D images from a series of 2D X-ray images. CT data can be manipulated through “windowing” to display various bodily structures. Contrast agents, such as iodine and barium, may enhance images by highlighting specific areas. The procedure involves a motorised table moving the patient through a circular opening while an X-ray source and detector assembly rotate around them. The detectors register X-rays passing through the patient, creating multiple snapshots reconstructed by a computer into cross-sectional images of internal organs and tissues [3].

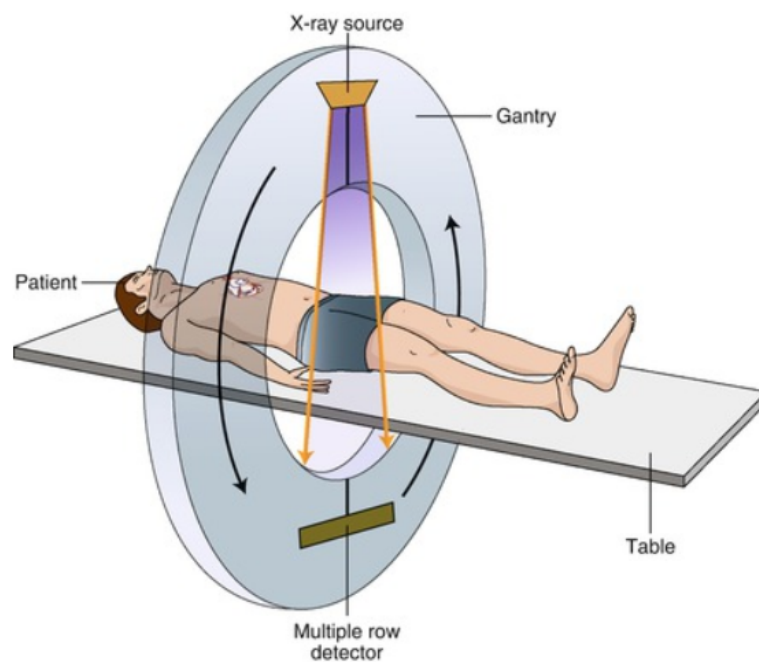


Figure 2.2: A cartoon depiction of a typical CT scanner, from [2].

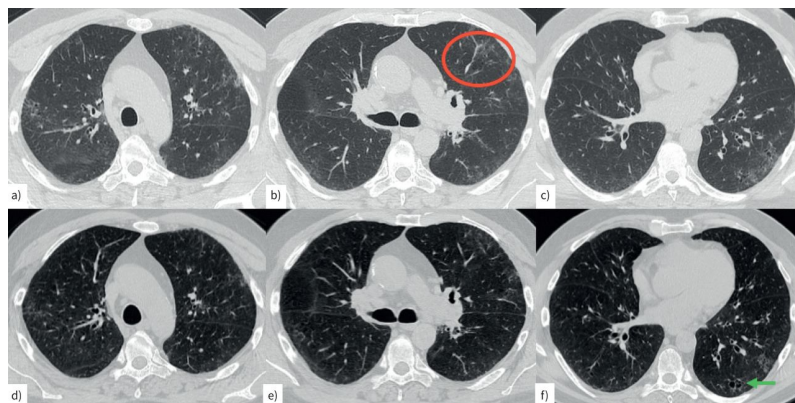


Figure 2.3: Example of CT scan slices of the lungs in a 60-year-old male smoker, (from [78]).

2.4 Lung Cancer

Lung cancer is a type of cancer that starts in the lungs. It is one of the most common types of cancer, especially among men.

2.4.1 Types of Lung Cancer

- **NSCLC:** This is responsible for 80% to 85% of lung cancers. Sub-types include:
 - Adenocarcinoma: frequent in smokers and nonsmokers, particularly women;

- Squamous cell carcinoma: associated with smoking and located in central lung areas;
- Big cell carcinoma: may develop anywhere.

Adenocarcinoma is frequently discovered early, particularly when it is in situ [11];

- **SCLC:** This accounts for approximately 10% to 15% of lung malignancies, also known as oat cell cancer. It grows and spreads quickly, frequently beyond the lungs at diagnosis. While chemotherapy and radiation therapy work well initially, recurrence is typical [11].

2.4.2 Lung Cancer Statistics

The following data is related to cases diagnosed in 2020 [13].

- **Both Sexes:** The most prevalent cancers globally were lung and breast cancers, accounting for 12.5% of all new cases (2.2 million new diagnoses), respectively [13];
- **Men:** Lung cancer accounted for 15.4% of all new cases (1.4 million new diagnoses), making it the most frequent cancer among men worldwide [13];
- **Women:** Lung cancer contributed to 8.8% to the total number of new cases (740 thousand new diagnoses) [13].

2.4.3 Lung Cancer Survival Rate

Lung cancer is sadly one of the deadliest types of the disease, taking more lives annually than breast, prostate, and colon cancers put together [12].

A frequent fallacy is that lung cancer's high mortality rate is exclusively due to its prevalence. However, the fatality rate is disproportionately high when compared to other common malignancies, such as breast cancer. Since 1989, advances in treatment and early detection have resulted in a considerable reduction in breast cancer fatalities [12].

Breast cancer can now be identified early, enhancing survival prospects significantly. In contrast, only 16% of lung cancers are detected early because symptoms appear after the cancer has progressed. Because lungs cannot be inspected externally, early detection is difficult [12].

When lung cancer spreads, it becomes metastatic, which makes therapy much more difficult. Chemotherapy is usually the primary treatment because most cases are diagnosed at this stage. However, it is not always effective. Some cells defy chemotherapy, adding to the high death rate of lung cancer. See Figures 2.4 and 2.5 for the survival rates after one year and 5 years, respectively.

Thankfully, there are a lot of researchers working hard all over the world to develop better treatment alternatives and early detection techniques. Low-dose CT screening has

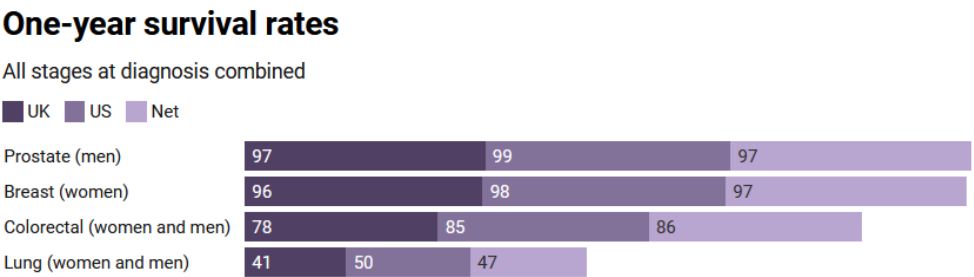


Figure 2.4: One-year cancer survival rates, (from [13]).

been demonstrated in recent trials to help in the early diagnosis of lung cancer. Those who are thought to be at high risk for lung cancer, such as those who smoke frequently or who smoked frequently in the past, can get this screening [12].

2.5 Summary

Since most cases of lung cancer are detected at a later stage, the disease progresses quickly and spreads to other parts of the body. Furthermore, the 5-year survival rate decreases dramatically as cancer spreads, making an early diagnosis crucial to the patient’s prognosis. Regarding disease markers, pulmonary nodules are an excellent place to start because research has shown that the existence and size of the nodules correlate with the chance of cancer. CT scans are typically used for lung cancer diagnosis and analysis because they are inexpensive, offer a 3D image of the affected area, and have faster scan times than other options [59].

Five-year survival rates %

All stages at diagnosis combined

■ UK ■ US ■ NL

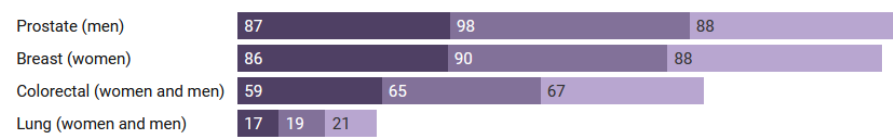


Figure 2.5: Five-year cancer survival rates, (from [13]).

Chapter 3

Literature Review

Contents

3.1	Introduction	15
3.2	Variational Auto-Encoders	16
3.3	Autoregressive Generative Models	17
3.4	Generative Adversarial Networks	19
3.5	Normalising Flows	41
3.6	Diffusion Models for 3D Medical Imaging	44
3.7	MedSyn	51
3.8	Evaluation Metrics	54
3.9	Summary	57

This chapter introduces some primary generative models: autoregressive models, VAEs, GANs, normalising flows. While the emphasis of this work is on synthesising 3D data, the majority of the models discussed here are trained on and generate 2D data. Nevertheless, the analysis focuses on the methods, which can often be modified to be applied to a 3D context. Other recent methods, such as Diffusion Models are also discussed.

3.1 Introduction

Generative models [36] are a type of machine learning algorithm that, given a sample of some training data, can obtain new synthetic data by mimicking the original dataset. They work by learning underlying patterns or distributions in the authentic images and using them to create newer ones.

The above is done by performing density estimation, which is the estimation of the probability distribution of a dataset. This estimation can be **explicit**, where the model directly parameterises the probability distribution of the data, or **implicit**, where the model focuses on generating samples from the distribution without parameterising it.

Figure 3.1 shows these distinguishing and succeeding types, being the ones mentioned above, the ones discussed in this thesis.

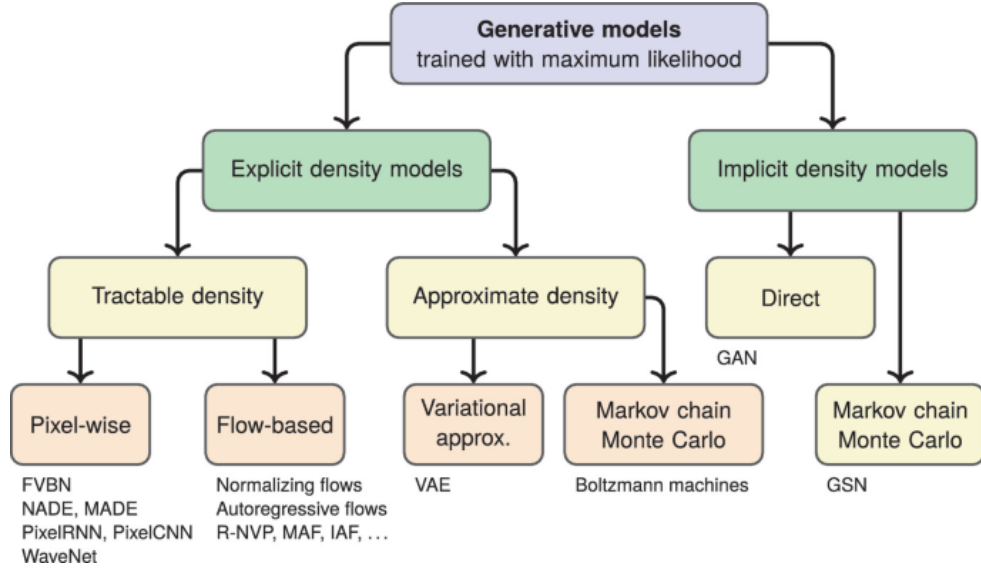


Figure 3.1: Types of generative models, (from [70]).

3.2 Variational Auto-Encoders

Autoencoders, which were first introduced in [82], are a type of unsupervised learning that reconstructs input data and learns representations. The objective of this type of model [19], is to learn and find the functions $A: \mathbb{R}^n \rightarrow \mathbb{R}^p$ and $B: \mathbb{R}^p \rightarrow \mathbb{R}^n$, which are the encoder and decoder, correspondingly [20], and satisfy Equation 3.1:

$$\operatorname{argmin}_{A,B} E[\Delta(x, B \circ A(x))], \quad (3.1)$$

, where Δ is the reconstruction loss function, which calculates the difference between the decoder's output and the input, and E is the expectation across the distribution of x , [20].

Figure 3.2 illustrates the architecture of an autoencoder.

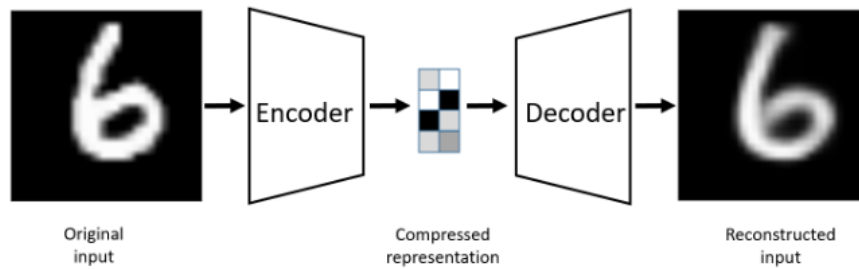


Figure 3.2: After being compressed into a representation, the input image is decoded, (from [20]).

An autoencoder can generate data by learning a fixed latent representation, yet the latent space lacks regularisation and continuity due to its focus on minimising loss. VAEs [57] significantly improved the representation capabilities of autoencoders. VAEs are generative models that aim to understand and generate data through a probabilistic distribution [57]. They work by assuming that an observed dataset X with N independent and identically distributed samples x_i follows a probability distribution. In this case, each data point x_i is generated based on an unobserved or hidden random variable z_i , and the parameters θ control how this data is generated, which is equivalent to a probabilistic decoder [20]. On the other side, a distribution over the latent variable z_i given an item x_i is assumed, which is equivalent to a probabilistic encoder and is controlled by the parameters ϕ [20] (see Figure 3.3).

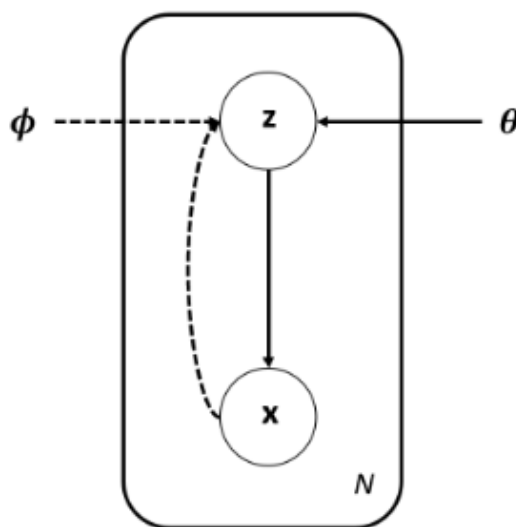


Figure 3.3: A Graphical Representation of VAE, (from [20]).

3.3 Autoregressive Generative Models

Autoregressive models are generative models that model a probability distribution of a sequence of data points. The core idea of this type of model is to predict the next data point in a sequence by learning about the previous ones, following the chain rule for conditional probability [22].

Oord et al. [72] proposed the PixelRNN model, which employs a recurrent neural network (RNN) that allows it to create images by progressively taking care of each pixel individually instead of creating the images at once. It assigns a probability to each pixel based on the previous pixels in the image grid. These are calculated considering the 3 RGB channels (i.e., Red, Green, and Blue).

The calculated probabilities are then used to attribute a probability to the image. The goal is to give each image I of size $N \times N$ pixels a probability $p(x)$. The total of all the conditional probabilities for every pixel adds up to this probability (see Equation 3.2):

$$p(x) = \prod_{i=1}^{n^2} p(x_i | x_1, \dots, x_{i-1}), \quad (3.2)$$

, where $p(x)$ is the probability of each image [72].

In [72], two-dimensional RNNs were applied to large-scale modelling of natural images. The resulting PixelRNNs comprise two Long Short-Term Memory (LSTM) types: Row LSTM and Diagonal BiLSTM. As a unidirectional layer, the Row LSTM computes features for a whole row at once by processing the image row by row from top to bottom. Figure 3.4 shows the layer which records a triangle context above each pixel x .

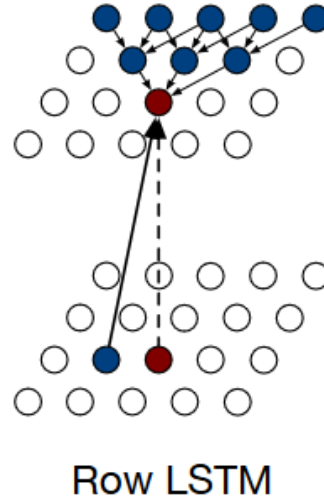


Figure 3.4: Visualisation of the input-to-state and state-to-state mapping of the Row LSTM architecture, (from [72]).

In the Diagonal Bidirectional LSTM (BiLSTM), instead of applying the convolution along each row, the convolution is applied along the diagonals of the images (see Figure 3.5). This way, Diagonal BiLSTM uses the whole context of the pixels before the selected pixel instead of the triangular context used in Row LSTM. In the test set, the Diagonal BiLSTM achieved better performance results.

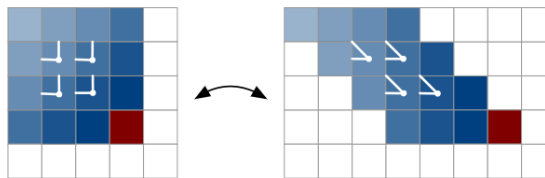


Figure 3.5: Diagonal BiLSTM, (from [72]).

Given the large number of pixels in large image datasets, PixelRNN and Diagonal BiLSTM tend to be disadvantageous regarding overall results because they generally achieve better performance and take a long time to train. This is because the models can use the context of all the other pixels in each checked pixel, making them computationally demanding [73].

PixelCNN, modelled with convolutional networks, solves this problem because convolutions are inherently easier to parallelise [73].

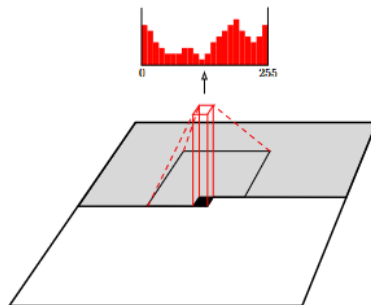


Figure 3.6: A visualisation of the PixelCNN that maps a neighborhood of pixels to prediction for the next pixel, (from [73]).

One other variant of autoregressive models is the PixelSNAIL [26], which combines causal convolutions [85, 73] with self-attention [94] and is specifically aimed at overcoming the problem of long-range dependencies that tend to be a bottleneck for RNN-based models. Applying convolutions across the sequence (masked or moved so that the current prediction is only influenced by the preceding element) is what these causal convolutions do. Even though the receptive field is still limited, this provides better bandwidth access to the earlier portions of the sequence, so attenuation is still apparent on elements that are far from the sequence [26]. The self-attention transforms sequences into unordered key-value stores, enabling content-based queries and maintaining an unbounded receptive field. This characteristic allows seamless access to information across distant sequence parts without degradation [26].

By leveraging the advantages of self-attention and convolution models, PixelSNAIL outperforms all of the previous models [26].

3.4 Generative Adversarial Networks

Using an adversarial process made up of two distinct models, generative modelling is accomplished by GANs [37]:

- **Generative model G** that represents the distribution of the data;
- **Discriminative model D** that calculates the likelihood that a sample originated from the training set instead of G .

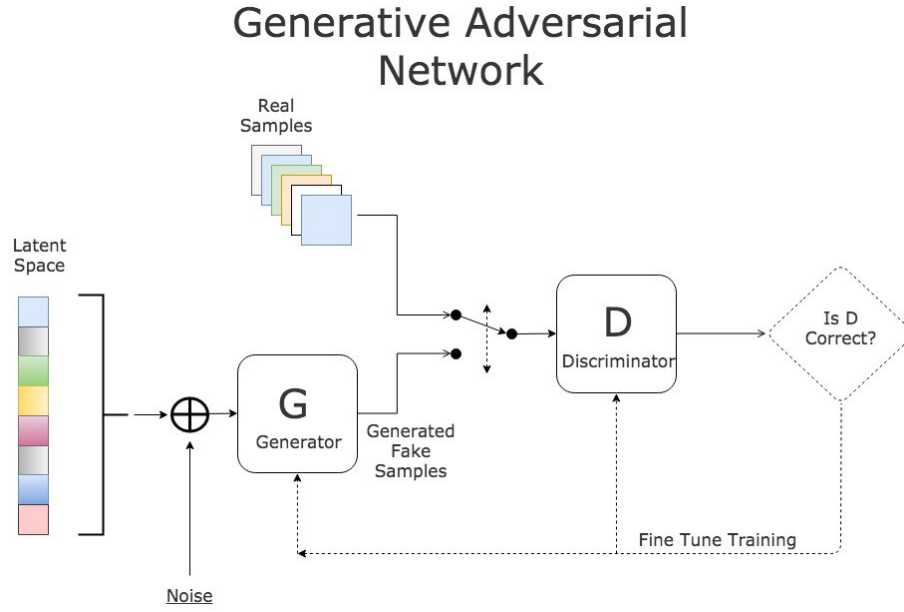


Figure 3.7: Generative Adversarial Networks, (from [37]).

In theory, the objective of the generator (G) is to maximise the probability of inducing a mistake in the discriminator (D) (minimising $\log(1 - D(G(z)))$ in Equation 3.3), which should minimise its error when distinguishing between real and fake samples. Therefore, since the G and D want to maximise probabilities and minimise errors, they continuously improve by playing what's called a minimax game where the G gets better at generating realistic data, and the D gets better at distinguishing real from fake data [37]. Equation 3.3 shows this application:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))], \quad (3.3)$$

, where $\min_G \max_D V(D, G)$ is the adversarial game between the G and D , the expected value which is denoted by $\mathbb{E}$ is $\mathbb{E}_{x \sim p_{data}(x)} [\log D(x)]$ which is taken over the real data distribution $p_{data}(x)$, and $\mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$ is the expected value over the G 's input noise distribution $p_z(z)$.

However, in practice, there are better approaches to the problem than this, especially in the first stages of training. This happens if the D becomes too accurate at distinguishing fake results from the actual data. In contrast, if the G is still poor at generating realistic data, the required signals updating the G would become weak or non-existent. This would mean the G 's learning process would be flawed [37].

To solve this early-stage problem, an alternative is used where we stop minimising the probability of the D being correct. Rather, what is maximised is the likelihood that the D would classify generated data incorrectly (therefore maximising $\log D(G(z))$) [37].

This training process ends when G can generate fake samples that cannot be distinguished by D .

3.4.1 Conditional GAN

Conditional GANs (CGANs) [66] (see Figure 3.8), by conditioning the G and D on some more knowledge y , can expand the basic generative adversarial networks. Any extra auxiliary data, such as class labels or labels from different modalities, can be represented by y . This additional information y is fed into the D and G as an additional input layer to accomplish the previously specified conditioning [66].

The input noise $p_z(z)$ is mixed with y in joint hidden representations in the G, and x and y are supplied as inputs to a discriminative function in the D [66].

Considering the Equation 3.3 presented before, for CGANs, there are a few changes to it regarding the addition of y (see Equation 3.4):

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x|y)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z|y)))], \quad (3.4)$$

where the terms of the equation are the same as in 3.3, apart from $D(x|y)$ which represents the addition of y to the D and $G(z|y)$ which represents the addition of y to the G.

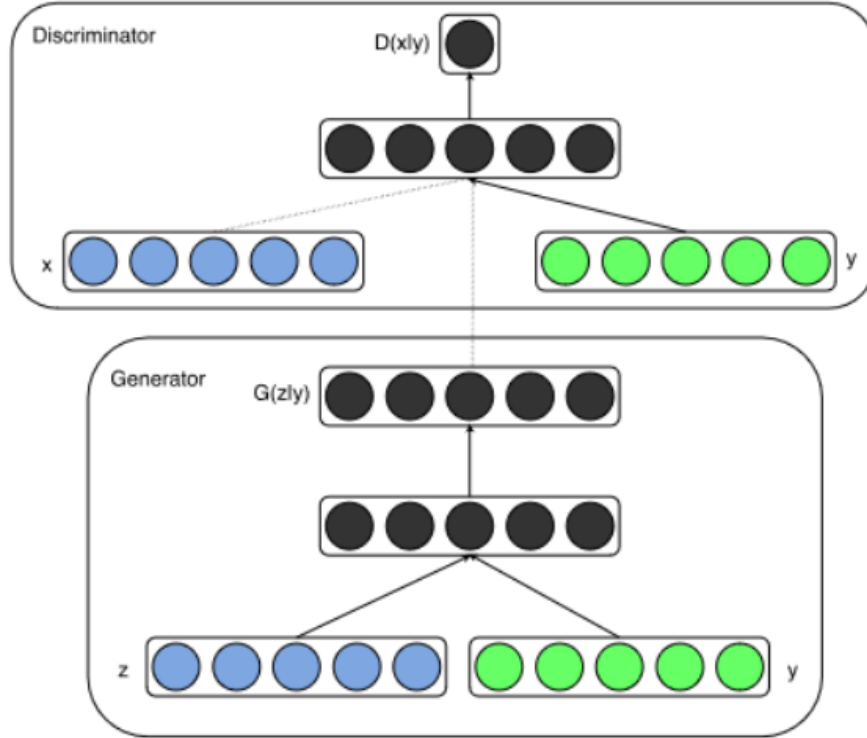


Figure 3.8: The architecture of CGAN, (from [66]).

Auxiliary Classifier GANs (AC-GAN) [71], are a type of CGAN architecture in which each generated sample has a class label associated with it, $c \sim p_c$ in addition to the noise z . The D classifies the label of the data instead of using them as input, showing improvements

to the original CGAN [71]. Equation 3.5 shows the modification implemented in the original loss function (Equation 3.3).

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] + \mathcal{L}_c, \quad (3.5)$$

where,

$$\mathcal{L}_c = E[\log P(C = c | X_{real})] + E[\log P(C = c | X_{fake})], \quad (3.6)$$

In Equation 3.6, $E[\log P(C = c | X_{real})]$ and $E[\log P(C = c | X_{fake})]$ measure how well the classifier C can correctly predict the class label c for real data samples X_{real} and generated data samples X_{fake} , respectively.

Figure 3.9 presents samples generated from AC-GAN and previous state-of-the-art model [84].



Figure 3.9: On the left there are examples of samples generated with the AC-GAN using the ImageNet dataset, and on the right examples of samples generated from the model in [84], (from [71]).

3.4.2 Image-to-image Translation

Introduced in [41], image-to-image translation is presented as a two-stage framework. Firstly, a pair of images, where one is a modified version of the other, is used as training data. Then, this learned filter is applied to a new target image, producing a similar modified result. In other words, the main objective is to find an “analogous” image B' that relates to B in “the same way” as A' relates to A [41].

3.4.2.1 Pix2Pix

A CGAN-based architecture called Pix2Pix [51] employs Convolutional PatchGAN as the D and U-Net [80] for the G . With the inclusion of L_1 distance as a loss term in G to promote reduced blurring in the generated outputs, the Pix2Pix training procedure is

identical to that of the CGAN. L_2 was also tested as a loss term, but the results were not satisfying, producing blurrier images. See Equation 3.7 for the final loss function [51]:

$$\min_G \max_D V(D, G) = \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G), \quad (3.7)$$

where,

$$\mathcal{L}_{cGAN}(G, D) = E_y[\log D(y)] + E_{x,z}[\log(1 - D(G(x, z)))], \quad (3.8)$$

$$\mathcal{L}_{L1}(G) = E_{x,y,z}[\|y - G(x, z)\|_1], \quad (3.9)$$

where, in Equation 3.8, $E_{x,z}[\log(1 - D(G(x, z)))]$ represents the expected G loss, and $E_y[\log D(y)]$ represents the expected D loss, and in Equation 3.9, $E_{x,y,z}[\|y - G(x, z)\|_1]$ represents the average pixel-wise absolute difference between y and $G(x, z)$.

The hyper-parameter λ regulates the significance of the L_1 distance term. Because z is not used for G in the actual implementation, the results are deterministic. Rather, only dropout is given to many G layers during training and testing to produce noise [51].

As said before, for the G, a U-Net [80] architecture was chosen. This architecture, paired with skip connections, allows the G to circumvent the absence of low-level features in conventional encoder-decoder architectures [51]. See Figure 3.10 for an example of skip-connections' impact.



Figure 3.10: Incorporating skip connections within an encoder-decoder architecture, forming a U-Net inspired architecture, significantly enhances the overall quality of the outcomes, (from [51]).

For the D, and since the GAN D is constrained to model high-frequency structures only, Convolutional PatchGAN was used, focusing on local image patches and penalising structure at the patch scale. The D attempts to determine if every $N \times N$ patch in a picture is authentic or phony. The final output of D is obtained by averaging all of the responses obtained from this convolution applied throughout the image. Figure 3.11 shows

a comparison between patch sizes on the PatchGAN.



Figure 3.11: Variation of patch sizes on PatchGAN. Larger patch sizes enforce more sharpness; we can see that 70×70 outputs are sharp and more realistic, and 286×286 is similar to 70×70 , (from [66]).

Figure 3.12 shows examples of the results of Pix2Pix on different scenarios.

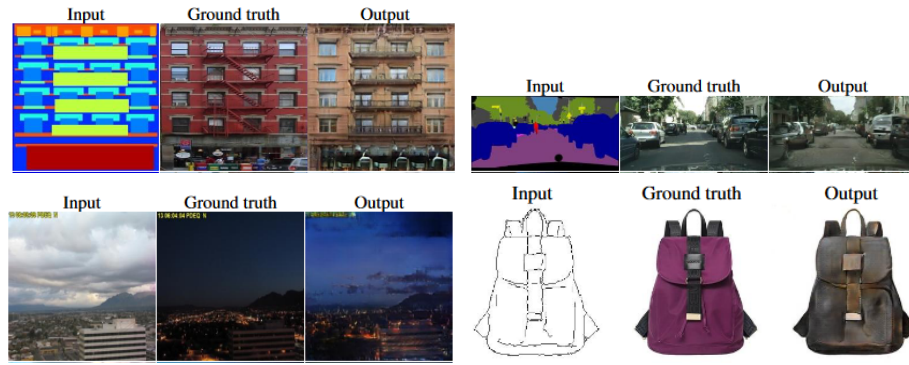


Figure 3.12: Examples of the results of Pix2Pix on different scenarios. Facades labels $\rightarrow$ photo (top-left), cityscapes labels $\rightarrow$ photo (top-right), day $\rightarrow$ night (bottom-left), and automatically detected edges $\rightarrow$ bags (bottom-right), (from [51]).

3.4.2.2 Cycle GAN

Cycle GANs [110] offer an alternative to image-to-image translations where paired training data is unavailable, mostly because it might be costly and challenging to obtain paired data. Using an adversarial loss, the goal is to train the mapping $G: X \rightarrow Y$ so that the distribution of images produced by $G(X)$ cannot be separated from the distribution Y . An inverse mapping is additionally obtained where $F: Y \rightarrow X$, as this mapping is not well-constrained. A cycle consistency loss is used to guarantee that $F(G(X)) \approx X$ [110] (see Figure 3.13).

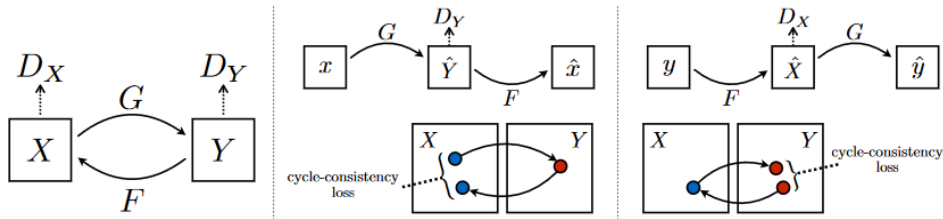


Figure 3.13: The two mapping functions G and F (left), forward cycle consistency loss (middle) and backward cycle consistency loss (right), (from [110]).

Two adversarial D's, D_x and D_y , are present in addition to the two mapping functions. D_x seeks to discriminate between pictures x and translated images $F(y)$, while D_y seeks to distinguish between images y and translated images $G(x)$. The objective loss consists of two main components, as mentioned earlier: cycle consistency losses, which ensure the mappings G and F remain consistent with each other [110], and adversarial losses [37], aimed at aligning the distribution of generated images with the target domain's data distribution.

The full objective function is given by Equation 3.10:

$$\mathcal{L}_{GAN}(G, F, D_x, D_y) = \mathcal{L}_{GAN}(G, D_y, X, Y) + \mathcal{L}_{GAN}(F, D_x, Y, X) + \lambda \mathcal{L}_{cyc}(G, F), \quad (3.10)$$

where λ controls the relative importance of the two objectives.

The first and second terms of the objective function are expressed in Equations 3.11 and 3.12, respectively:

$$\mathcal{L}_{GAN}(G, D_y, X, Y) = \mathbb{E}_{y \sim p_{data(y)}}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{data(x)}}[\log(1 - D_Y(G(x)))] \quad (3.11)$$

$$\mathcal{L}_{GAN}(F, D_x, Y, X) = \mathbb{E}_{x \sim p_{data(x)}}[\log D_X(x)] + \mathbb{E}_{y \sim p_{data(y)}}[\log(1 - D_X(G(y)))] \quad (3.12)$$

where in the first one, D_Y seeks to discern between translated samples $G(x)$ and genuine samples y , G attempts to produce pictures $G(x)$ that resemble images from domain Y . For Equation 3.12, the situation is the opposite [110].

The third term is expressed in Equation 3.13:

$$\mathcal{L}_{cyc}(G, F) = \mathbb{E}_{x \sim p_{data(x)}}[\log D_X(x)] + \mathbb{E}_{y \sim p_{data(y)}}[\log(1 - D_X(G(y)))] \quad (3.13)$$

Figure 3.14 presents the results obtained from applying cycle consistency loss may be examined, and it can be observed that the reconstructed images $F(G(x))$ closely resemble the input images x [110].

3.4.3 Deep Convolutional GAN

The Deep Convolutional GAN (DCGAN) [77] surges as a way to increase the performance of GANs in images by using Deep Convolutional Neural Networks (DCNNs). For this, the authors applied a series of transformations to original Convolutional Neural Network (CNN) architectures.

The first main change is the application of an all convolutional [88] net, which replaces the deterministic spatial pooling functions (e.g., max-pooling) with strided convolutions in D and fractionally-strided convolutions in G , which allows the network to learn the D and spatial upsampling on its own [77]. The second main change is eliminating fully-connected

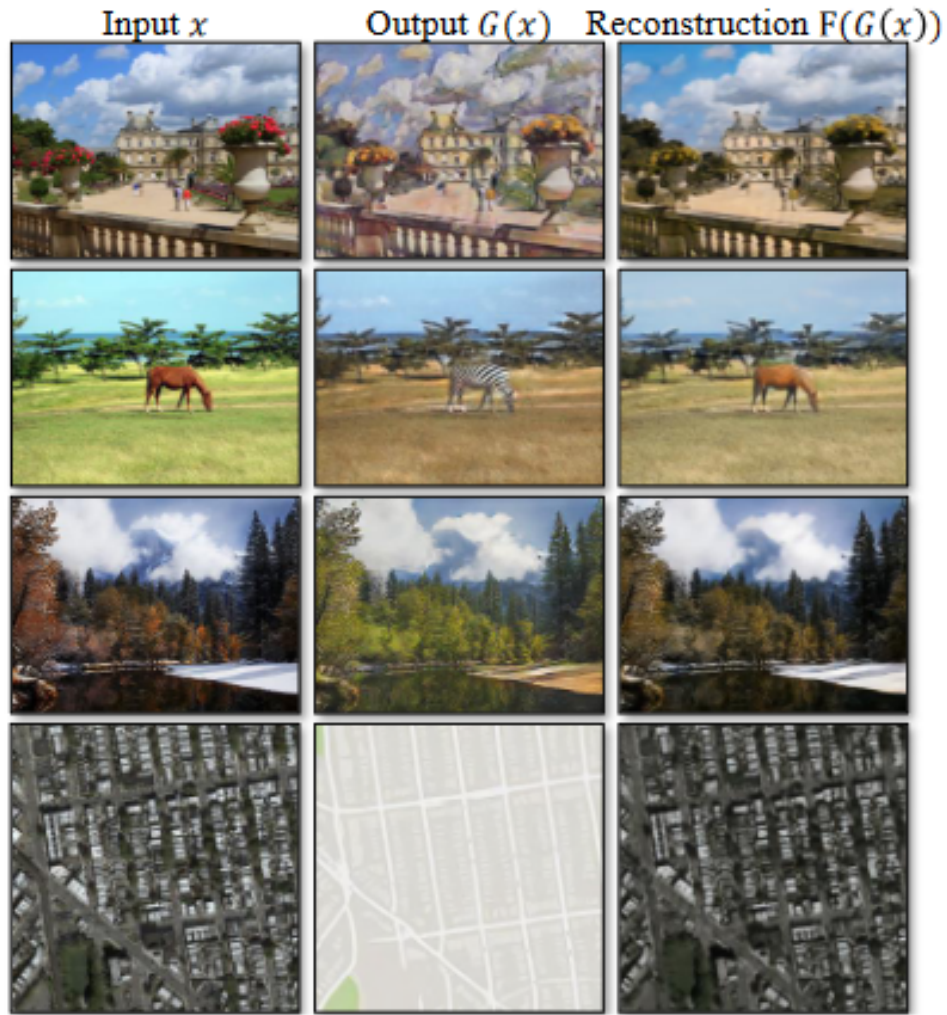


Figure 3.14: Cycle consistency loss behaviour. There is a high similarity between the original inputs and the reconstructed images, (from [110]).

layers on top of convolutional features (see Figure 3.15). The third main change is the batch normalisation [50] that maintains learning stability by standardising the input for each unit to achieve a zero mean and a variance of one.

The DCGANs G (see Figure 3.15) was used for Large-Scale Scene Understanding (LSUN) which is an important image dataset built using human-in-the-loop supervision and deep learning techniques, making it easier to train and assess machine learning models on a wide scale [106].

Some other changes that were applied in the DCGAN architecture are the use of Rectified Linear Unit (RELU) activation [68] in G for all layers except output, which uses hyperbolic tangent (Tanh), and the use of LeakyReLU activation [63, 103] in the D for all layers.

The study [77] shows that on CIFAR-10, DCGANs outperform K-means in generating features for supervised learning, and on StreetView House Numbers dataset [69], it

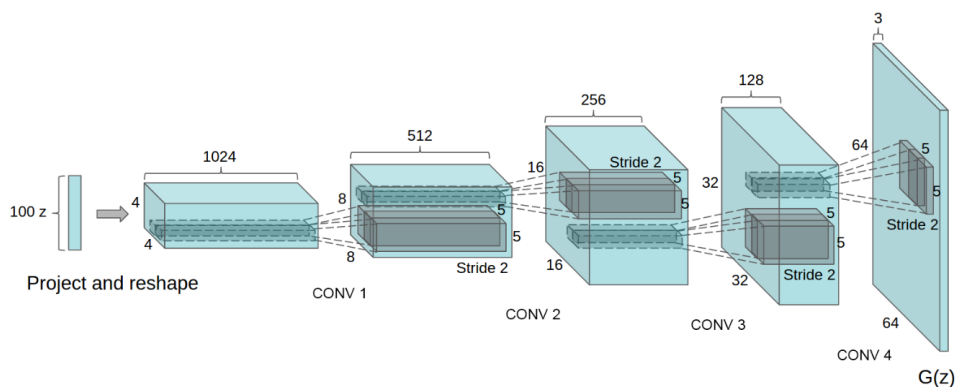


Figure 3.15: DCGAN G used for LSUN [106] scene modelling where we can see a series of four fractionally-strided convolutions which then convert a 100-dimensional uniform distribution Z into a 64×64 pixel image, (from [77]).

outperforms other CNN-based methods.

3.4.4 Progressive Growing GAN

Progressive Growing GANs (PGGANs) [53] are generative networks that begin with low-resolution images and gradually add layers to the networks to raise the resolution (see Figure 3.16).

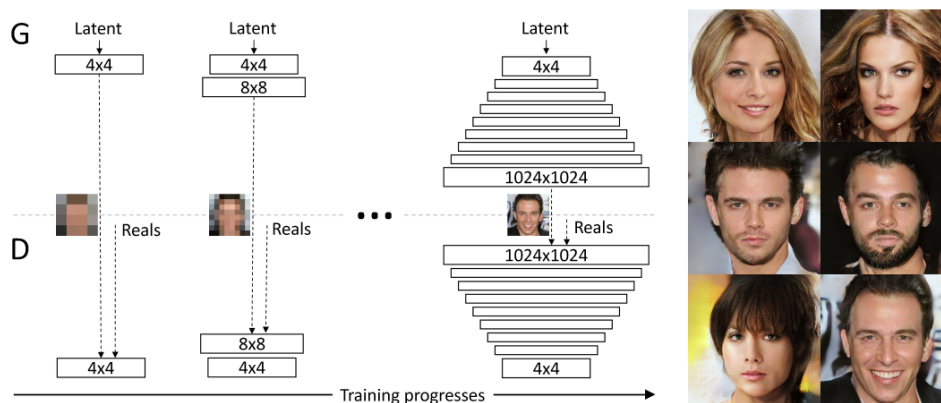


Figure 3.16: The training process starts with low spatial resolution, which is gradually increased by adding layers to both the G and D, (from [53]).

These extra layers are added such that there is no disruption to the already trained layers of the model. This progressive training allows the stabilisation of the generation process for high-resolution images and the stabilisation of training, especially in generating smaller images. Depending on the final output resolution, the training period is also shortened by two to six times because most iterations take place at lower resolutions [53].

GANs capture just a portion of the variation included in the training batch of data. They gather feature statistics from single images and across groups of images, encouraging consistency between batches of generated and actual training images. This involves incorporating a minibatch layer at the end of the D. Here, this layer learns to create a comprehensive set of statistics by projecting input activations into an array of statistical measures. Each example within a minibatch generates its own set of statistics, which is then combined with the layer's output. This allows the D to utilize these statistics internally [53].

In PGGANs, this process is significantly simplified while also enhancing the diversity within the generated outputs. The simplification works by computing standard deviations across batches without introducing new settings. This results in a single value replicated across spatial areas, enhancing variation. Despite trying other statistical methods, this approach remained effective [53].

Figure 3.17 shows some examples of generated images.



Figure 3.17: 1024 x 1024 photos produced with the CELEBA-HQ [53] collection, (from [53]).

Signal magnitude escalation is a common problem with GANs due to unhealthy competition between the two networks, so to address this problem, two different solutions were introduced [53].

- **Equalised Learning Rate:** A standard initialisation and dynamically scaling weights at runtime are employed. This ensures a consistent dynamic range and learning speed for all weights, avoiding issues arising from varying parameter scales [53];
- **Pixelwise Feature Vector Normalisation in G:** To fight uncontrolled signal amplification in the G, the feature vectors within each pixel are normalised. This normalisation effectively controls signal amplification without negatively impacting the performance of the G [53].

3.4.5 Wasserstein GAN

Wasserstein GAN (WGAN) [14] is a generative model that minimises a reasonable and efficient approximation of the Earth Mover (EM) distance, which is also known as the Wasserstein Distance. There are a few probability distances used to measure similarities between distributions, such as Total Variation (TV), Kullback-Leibler (KL) divergence, Jensen-Shannon (JS) divergence, and Earth-Mover (EM) distance. However, some distances, like JS and KL, fail to capture convergence for certain sequences of distributions. This highlights that when learning distributions are supported by low-dimensional manifolds, the KL, JS, and TV distances are not reasonable cost functions. But in that configuration, the EM distance makes sense, which is why the WGAN model uses it [14].

The paper also introduces Kantorovich-Rubinstein duality [95], which equates the EM distance to a supremum involving 1-Lipschitz functions. By training a neural network to approximate this supremum and solve for the EM distance, using weight clamping to enforce Lipschitz constraints [14], the method becomes practically advantageous. This trick simplifies the training process, ensuring stable and reliable convergence, which is crucial for effective neural network performance.

The authors then try to find a D that maximises the difference between two distributions, allowing for calculating the EM distance through neural network optimisation. They propose the Wasserstein Generative Adversarial Network (WGAN) algorithm to approximate the EM distance, which involves training a D to optimality, emphasising the use of weight clipping to enforce Lipschitz constraints, despite acknowledged limitations in its effectiveness. The benefit of the EM distance being continuous and differentiable in most areas, which enables training the D until optimality, contrasting this advantage with the limitations is also emphasised. This approach of the algorithm also prevents mode collapse [14].

Image creation using the LSUN-Bedrooms dataset as the target distribution for learning [106] was employed for the experiments, and the results were compared with the baseline DCGAN.

Figures 3.18 and 3.19 show the comparison between WGAN and the standard GAN procedure outputs and more outputs of the model.



Figure 3.18: A G without batch normalisation and an equal number of filters at each layer were used to train both algorithms. On the left is the WGAN algorithm, and on the right is a conventional GAN formulation. As we can see, the WGAN was still able to generate samples but the normal GAN was unable to learn, (from [14]).



Figure 3.19: WGAN algorithm where the G and D are DCGANs, (from [14]).

3.4.6 α -WGAN

To improve the generative modelling process, α -WGAN combines the benefits of Wasserstein based metrics with autoencoder structures. The fundamental breakthrough of α -WGAN is the integration of adversarial training with an autoencoder-derived reconstruction loss. The two main issues with GANs are mode collapse and the absence of clear ways to ensure data fidelity, which are addressed by this dual strategy [81, 59], which takes advantage of 2 different loss functions:

- **Adversarial Loss:** uses the Wasserstein distance with gradient penalty, same as WGAN-GP, to guarantee steady training and significant gradient flows. By reducing the Wasserstein distance between the generated and real data distributions, this adversarial component aids the G in producing diverse and realistic data samples [59];
- **Reconstruction Loss:** a reconstruction loss is incorporated into α -WGAN, directing the G to produce data that can be precisely reconstructed from a latent representation. By maintaining the generated sample's structural integrity, this loss term helps to ensure that they are both realistic and true to the underlying data distribution [59].

Regarding the architecture, it consists of 4 main components:

- **G:** Produces data samples from latent vectors;
- **Encoder (E):** Maps real data to the latent space;
- **D** Differentiates between real and generated data;
- **Code Discriminator (C):** Differentiates between encoded real data and randomly sampled latent vectors.

As for the training procedure, it is trained by periodically updating the G, E, D, and code D. In particular, the E and G are updated twice for every iteration for every update

of the D and code D. The G and E can promptly adjust to the D’s feedback thanks to this training regimen [59].

Figure 3.20 shows an example of generated slices of the brain.

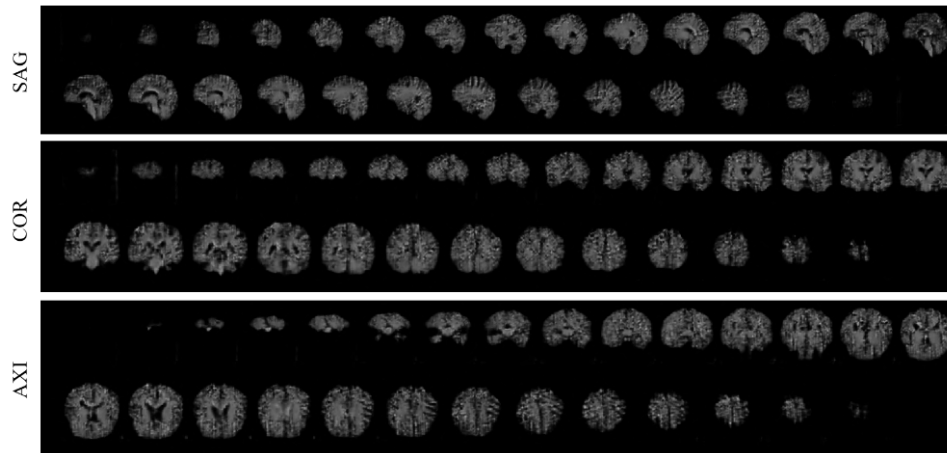


Figure 3.20: Slices of normal brain 3D samples from α -WGAN, (from [59]).

3.4.7 Style GAN

Karras et al. [54] introduce a novel G architecture for image synthesis that allows precise control over image features at different scales by adjusting the “style” based on a latent code at each convolution layer. This leads to automatically separating high-level attributes and stochastic variations in generated images, enabling intuitive scale-specific mixing and interpolation operations. The G’s design doesn’t modify the D or loss function, focusing on image synthesis control [54].

Usually, an input layer provides the latent code to the G. The authors deviate from this method by removing the input layer entirely and starting with a learned constant. They also use a mapping network to generate a style vector that controls adaptive instance normalisation (AdaIN) [48, 30, 35, 75] operations in the synthesis network. AdaIN applies normalisation, scaling, and biasing to feature maps using features from the style vector derived from the latent code [54].

This approach creates spatially invariant styles from the latent vector, emphasising its efficiency and suitability for their purposes compared to other feature transforms. Explicit noise inputs are also introduced to generate stochastic detail, feeding noise images to each layer of the synthesis network [54]. Figure 3.21 shows the G of the model presented.

The authors experimented with different G architectures, starting with a baseline and progressively enhancing it with techniques like style controls, noise inputs (see Figure 3.22), and mixing regularisation. Their style-based approach involves a mapping network that

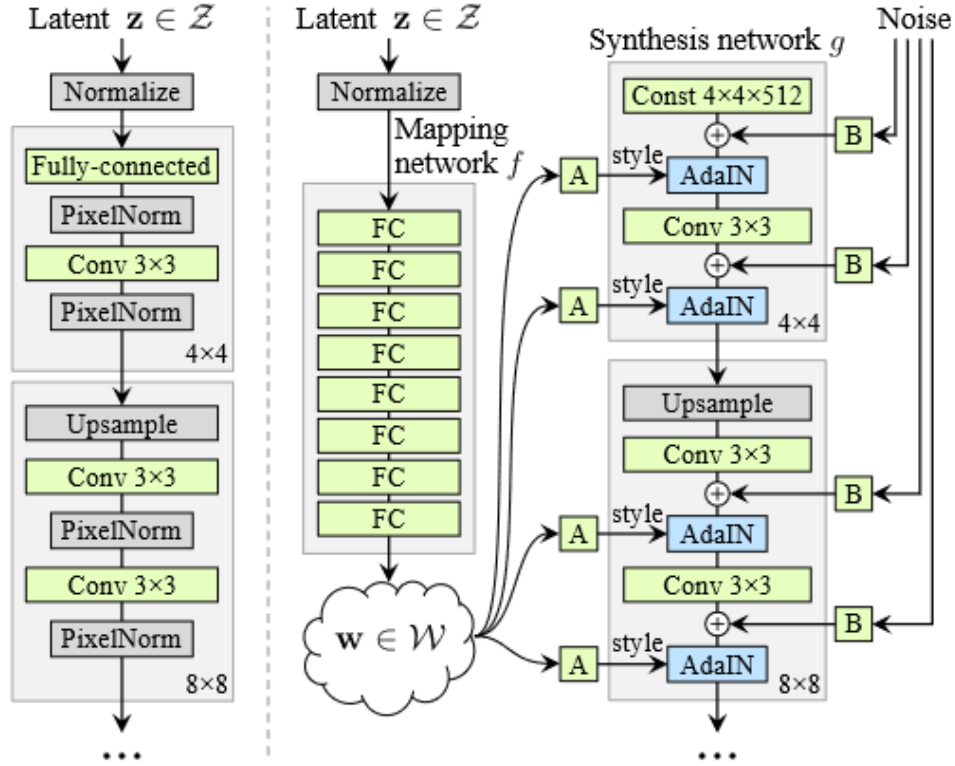


Figure 3.21: In contrast to standard G's [53], the model translates the input to an intermediate latent space W , which then influences the G through AdaIN at each convolution layer. Gaussian noise is applied after each convolution before measuring the nonlinearity, (from [54]).

generates style vectors to control adaptive instance normalisation operations in the synthesis network. The architecture allows scale-specific modifications to the styles, influencing different aspects of the generated images [54].

They also introduce noise inputs to control stochastic detail in the generated images and illustrate how noise affects different layers, highlighting its role in adding stochastic variation while maintaining overall composition (see Figure 3.23).

Finally, the authors mention separating global effects (such as pose and identity) from stochasticity in the images.

3.4.8 3D-GANs

Wu et al. [101] introduced 3D Generative Adversarial Networks (3D-GANs), a system that creates 3D objects from a probabilistic space using volumetric convolutional networks and generative adversarial nets.

A 200-dimensional latent vector z , chosen at random from a probabilistic latent space, is mapped by the G in the proposed 3D-GAN to a $64 \times 64 \times 64$ cube, which represents an object $G(z)$ in 3D voxel space. The D generates a confidence value $D(x)$ that indicates the accuracy of a 3D object input x [101].

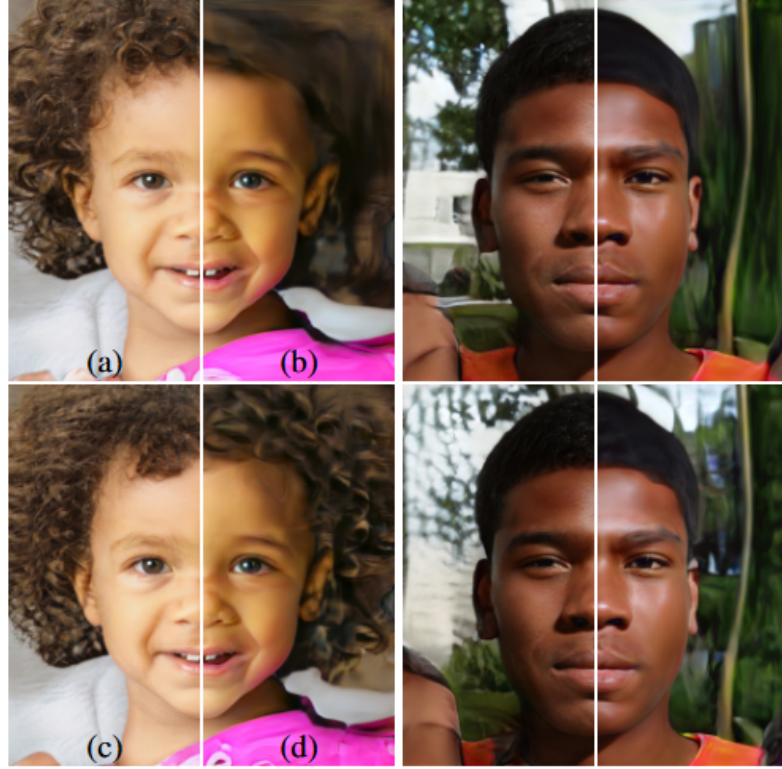


Figure 3.22: In image a) noise has been applied in every layer. In (b) there is no noise and in c) and d) there's noise only in fine layers and coarse layers, respectively, (from [54]).

Following [37], this model uses binary cross entropy as classification loss and the following function as the overall adversarial loss function (see Equation 3.14).

$$\mathcal{L}_{3D-GAN} = \log D(x) + \log(1 - D(G(z))), \quad (3.14)$$

where z is a noise vector randomly sampled from a distribution $p(z)$, and x is a real object in a $64 \times 64 \times 64$ space.

The network structure was inspired by [77], being an all-convolutional neural network. The G is composed of five volumetric convolutional layers. These convolutional layers have kernels of size $4 \times 4 \times 4$ with strides of 2. In between them, there are batch normalisation and ReLU layers, and at the end a Sigmoid layer [101] (see Figure 3.24). The D copies the G but uses Leaky ReLU [63] instead of ReLU layers [101].

In addition, the authors used an adaptive training technique that differs from the simpler one in which patches are applied to update G and D. This is because, in general, D learns significantly faster than the G because distinguishing between genuine and synthetic items is easier than producing objects in a 3D voxel space. It would, therefore, eventually become difficult for the G [101] to extract signals for improvement from the D.

The D is only updated for each batch if its accuracy in the previous batch was no more than 80%. This is how the approach operates. It was discovered that doing this improved



Figure 3.23: The first column shows two generated images. In the second column, we can see zoom-ins of the result of applying different realisations of input noise. Finally, the last column shows the standard deviation of each pixel over 100 different realisations, (from [54]).

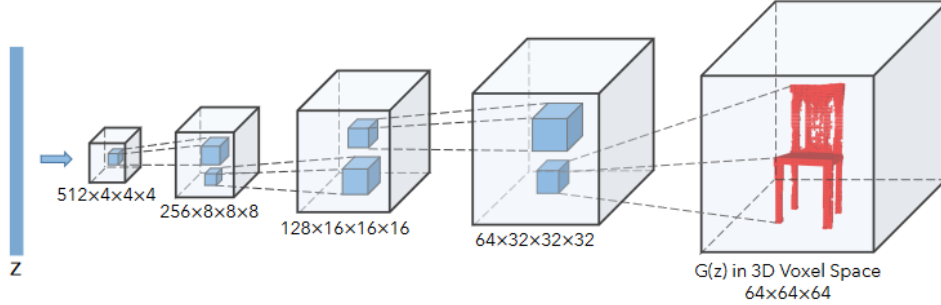


Figure 3.24: The model initially transforms the input into an intermediate latent space called W . This space subsequently influences the G via AdaIN at each convolutional layer. Gaussian noise is then incorporated after every convolution and before applying the non-linearity, (from [101]).

training results and helped stabilise it.

Figure 3.25 shows an example of some outputs of the model.

3.4.8.1 3D-VAE-GANs

A new image E , 3D-VAE-GAN, was introduced as an expansion of 3D-GAN, accepting 2D pictures x as input and outputting the latent representation vector z [61].

3D-VAE-GAN is composed of 3 components. A D , a G , and the image E which is composed of five spatial convolution layers.

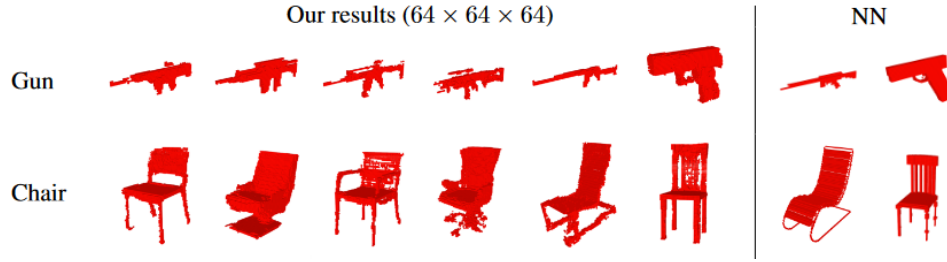


Figure 3.25: Objects created from vectors by 3D-GANs without the need for a reference object or image, (from [101]).

Like in [61], the loss function is composed of three components: a KL divergence loss $\mathcal{L}_{KL}$ to limit the distribution of the E's output, an object reconstruction loss $\mathcal{L}_{recon}$, and a cross-entropy loss $\mathcal{L}_{3D-GAN}$ (see Equation 3.15).

$$\mathcal{L} = \mathcal{L}_{3D-GAN} + \alpha_1 \mathcal{L}_{KL} + \alpha_2 \mathcal{L}_{recon}, \quad (3.15)$$

where α_1 correspond to the KL divergence loss and α_2 to the reconstruction loss.

Figure 3.26 shows an example of the reconstruction of images.



Figure 3.26: Examples of single image 3D results using the IKEA dataset [62], (from [101]).

The model was able to reconstruct volumes through images of the IKEA dataset (see Figure 3.26).

3.4.9 3D-GANs for Medical Imaging

Due to their ability to generate, translate, and reconstruct new data, GANs show immense promise in medical imaging, being currently one of the most extensively studied methods. This section examines cutting-edge GAN techniques for synthesising volumetric data and new approaches using other alternatives, such as text-guided synthesis.

3.4.9.1 3D Brain MRI Generation

Kwon et al. [59] proposed a 3D-GAN model for generating accurate 3D brain MRI scans using random vectors. This model extends α -GANs [81] by incorporating additional auto-encoder and critic-discriminator networks to mitigate issues like mode collapse and image

blurriness. To ensure stable training, it employs Wasserstein GAN with Gradient Penalty (WGAN-GP) loss functions [39].

In 3D generation, the mode collapse issue where GANs tend to generate a restricted range of outputs escalates significantly due to the increased complexity of the task. One way to fix this is using VAEs, which are free from mode collapse but present blurriness in the outputs [59].

The work discusses using α -GAN, a hybrid of GANs and VAEs, to overcome their limitations. α -GA introduces a CD network, transforming VAEs inference process into a GAN-like adversarial game, benefiting from both models to improve image generation quality in 3D tasks. The adaptation involves specialised 3D convolution layers, Batch-Norm, LeakyReLU [63, 103], and resize-convolution techniques in the networks, along with larger latent vectors for better image representation [59] (see Figure 3.27).

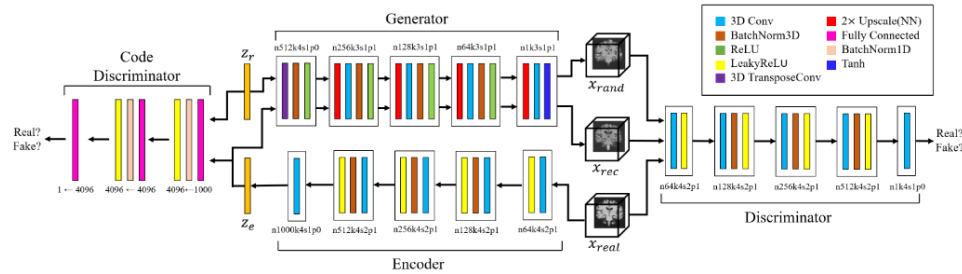


Figure 3.27: Detailed design of the suggested model, (from [59]).

Initially, the model faced instability with the basic loss function originating from GANs, leading to substantial differences between generated and real data distributions. It adopted the WGAN loss to address this, utilising Wasserstein distance and a gradient penalty for stability [59].

The comprehensive loss function involves four parts for each network: D, CD, G, and E. These modifications are pivotal for stabilising training and improving the accuracy of the generated data [59].

During training, the E and G operate as a single network. Their combined loss functions are summed up and optimised sequentially in the E-G, D, and CD order. To compensate for the G's slower optimisation speed, it undergoes updates twice within a single step [59].

The ADNI [76] dataset contains 991 T1 structural images that have been classified as control normal (CN) for the datasets that were utilised. Additionally, the model is trained using the ATLAS dataset for stroke magnetic resonance imaging (MRI) production and the BRATS [18, 65] 2018 dataset for brain tumor MRI generation (see Figure 3.28).

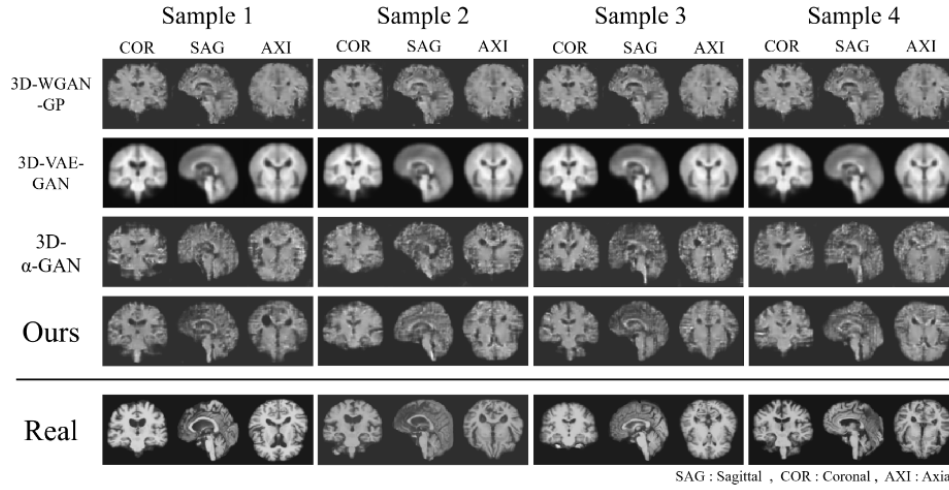


Figure 3.28: Samples of normal brain MRI, both real and created, (from [59]).

3.4.9.2 Cycle Consistent Embedding 3D Brain MRI Generation

This study [102] focuses on improving the previous models in two ways: using Wasserstein Auto-Encoding GAN to elevate generation quality via Wasserstein distance instead of CD and employing Cycle Consistent Embedding GAN (CCE-GAN) to refine the mapping with cycle consistency loss in the Z-space.

The objective is to improve the GAN framework for better image generation. GANs struggle with mode collapse in random image generation due to limited Z-space coverage. As presented before, VAE-GAN and 3D- α -WGAN [59] addressed this but suffered from blurry images or inaccurate Z-space constructions for 3D brains [102] (see Figure 3.29).

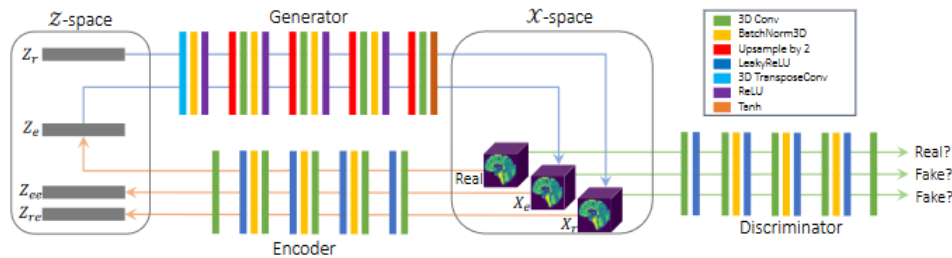


Figure 3.29: CCE-GAN Architecture, (from [102]).

With WAE-GAN, Wasserstein loss is employed in the Z-space for better mappings, and cycle-consistent embeddings are introduced. The resulting model, CCE-GAN, enhances image quality and Z-space accuracy without mode collapse, prioritising the improvement of E for better image generation [102] (see Figure 3.30).

3.4.9.3 Hierarchical Amortized GAN

HA-GAN [90] was presented as a solution to the problems associated with manufacturing effective high-resolution medical volumes. Typically, during training, insufficient memory

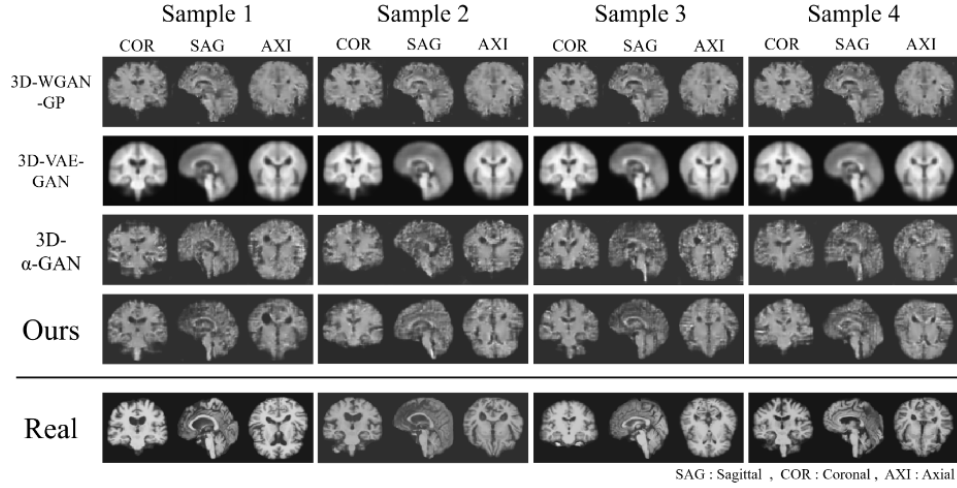


Figure 3.30: Examples of Real (Left), VAE-GAN generated (Middle), and CCE-GAN generated (Right) brains, (from [102]).

limits the generated images to $128 \times 128 \times 128$ pixels or smaller (see Figure 3.31). To surpass this HA-GAN uses a hierarchical framework to create high-resolution sub-volumes and low-resolution images during training, which distributes the memory load and maintains both the global structure and local details.

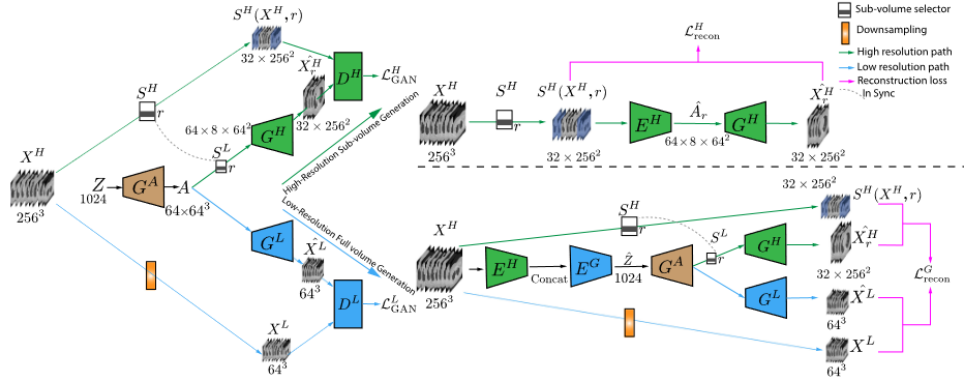


Figure 3.31: HA-GAN architecture. The G has two branches for generating low-resolution full volume and high-resolution sub-volume during training, instead of producing high-resolution full volume directly, (from [90]).

The model is comprised of a G, a D, and an E. A image with low-resolution G^L and a randomly chosen sub-volume of the image with high-resolution G^H are produced by the G's two branches. In the same way, the D is split into two branches: one D^H separates the true high-resolution sub-volume, and the other D^L separates the low-resolution image from the fake ones. In the high-resolution sub-volume, D^H guarantees the realism of the local details, while D^L guarantees the preservation of the correct global structure [90]. The author proposed two different GAN loss functions: $\mathcal{L}_{GAN}^H(G^A, G^H, D^H)$ and $\mathcal{L}_{GAN}^L(G^L, G^A, D^L)$ as shown on Equations 3.16 and 3.17.

$$\begin{aligned} \mathcal{L}_{GAN}^H(G^A, G^H, D^H) = & \min_{G^H, G^A} \max_{D^H} \mathbb{E}_{r \sim U} \left[\mathbb{E}_{X \sim P_X} [\log D^H(S^H(X^H; r), r)] \right. \\ & \left. + \mathbb{E}_{Z \sim P_Z} [\log(1 - D^H(\hat{X}_r^H, r))] \right] \end{aligned} \quad (3.16)$$

where the real high-resolution image is X^H , the high-resolution sub-volume selector is S^H , the underlying data distribution is P_X , the random noise distribution is P_Z , and the generated high-resolution sub-volume is $\hat{X}_r^H$, beginning at slice r [90].

$$\mathcal{L}_{GAN}^L(G^L, G^A, D^L) = \min_{G^L, G^A} \max_{D^L} \mathbb{E}_{X \sim P_X} [\log D^L(X^L)] + \mathbb{E}_{Z \sim P_Z} [\log(1 - D^L(\hat{X}^L))], \quad (3.17)$$

where X^L is real low-resolution image, and $\hat{X}^L$ is the generated low-resolution image.

The E also adopts a hierarchical structure with an E^H high-resolution sub-volume and E^G for the entire image. This process begins by breaking a high-resolution image into smaller, non-overlapping sub-volumes. These parts generate feature maps using an E, forming an overall image representation by combining these features [90]. To train this system, two objectives: $\mathcal{L}_{recon}^H(E^H)$ and $\mathcal{L}_{recon}^G(E^G)$ are used (see Equations 3.18 and 3.19).

$$\mathcal{L}_{recon}^H(E^H) = \min_{E^H} \mathbb{E}_{X \sim P_X, r \in U} \|S^H(X^H; r) - G^H(\hat{A}_r)\|_1, \quad (3.18)$$

$$\begin{aligned} \mathcal{L}_{recon}^G(E^G) = & \min_{E^G} \mathbb{E}_{X \sim P_X} \left[\|X^L - G^L(G^A(\hat{Z}))\|_1 \right. \\ & \left. + \mathbb{E}_{r \sim U} \left[\|S^H(X^H; r) - G^H(S^L(G^A(\hat{Z}); r))\|_1 \right] \right] \end{aligned} \quad (3.19)$$

Reconstructing a randomly chosen high-resolution sub-volume $S^H(X^H; r)$ is ensured by Equation 3.18. The low-resolution picture X^L and a randomly chosen $S^H(X^H; r)$ can be reconstructed given Z [90], according to Equation 3.19.

There is also the inference phase, where high-resolution images are directly generated by feeding Z into G^A and G^H sequentially [90] (see Figure 3.32).

We can then conclude that the overall model can be represented by the following loss function (see Equation 3.20).

$$\mathcal{L} = \mathcal{L}_{GAN}^H(G^A, G^H, D^H) + \mathcal{L}_{GAN}^L(G^L, G^A, D^L) + \lambda_1 \mathcal{L}_{recon}^H(E^H) + \lambda_2 \mathcal{L}_{recon}^G(E^G) \quad (3.20)$$

where λ_1 and λ_2 manage the balance between the GANs and reconstruction losses.

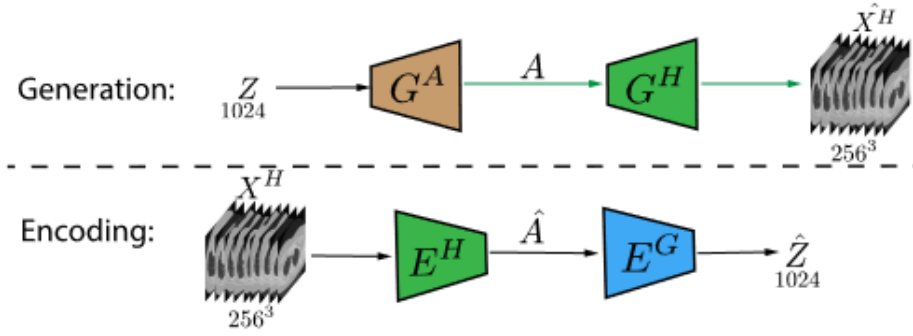


Figure 3.32: Inference with the hierarchical G and E, (from [90]).

The experiments used two datasets: the COPDGene [79] and the GSP [47]. See Figure 3.33 for the qualitative results.

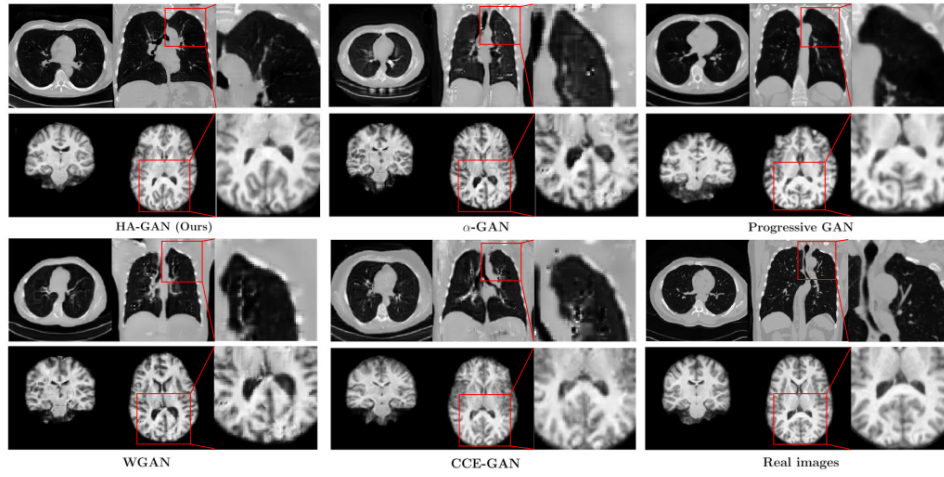


Figure 3.33: Real photos and random images produced by various models, (from [90]).

3.4.9.4 Lung Tissue Modelling in Pulmonary CT

The work [31] explains how GAN models from the literature have been modified to provide 3D output. The models studied are three GAN models: DCGAN [77], styleGAN [54], and bigGAN [24] for 3D healthy pulmonary CT image patch modelling. While 3D styleGAN generates realistic images with 3D structure, it faces mode collapse issues that need specific training attention. In contrast, 3D DCGAN offers more image variability at the expense of image quality. The 3D bigGAN strikes a balance, providing moderate image quality while accurately representing specific features [31].

The DCGAN architecture [77] was a pivotal step in integrating convolutions into GANs. In this study, the DCGAN3D G, an adaptation from a PyTorch implementation contains approximately 24 million parameters [31]. It starts by expanding the latent code through convolutions into a $4 \times 4 \times 4$ volume and progressively upsamples it using transposed convolution layers until reaching a $32 \times 64 \times 64$ output size. Each transposed

convolution operation includes 3D batch normalisation [50] and ReLU activation, except for the last layer, which uses tanh activation to restrict the range of the resulting image to -1 and 1 [31].

StyleGAN [54], revolutionised GAN techniques through style-based image generation, combining fully-connected layers and a multi-resolution convolutional G. It employs adaptive instance normalisation [48] for comprehensive alteration of overall image appearance, outperforming solely convolutional approaches. This model includes a mapping network transforming z into a refined space, P_w , based on semantic image features [31]. This study focuses on a 3D-style patch G with 18 million trainable parameters, omitting noise injection channels to control precise image detail for tasks like anomaly detection and image reconstruction (see Figure 3.34) [31].

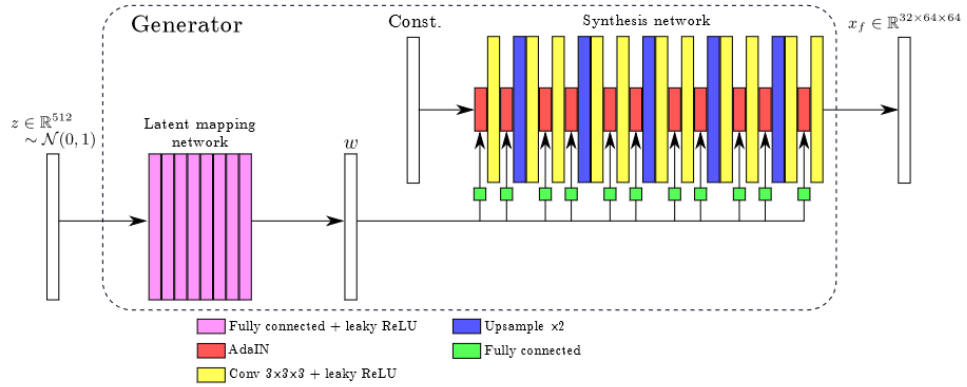


Figure 3.34: Architecture of the styleGAN3D G, (from [31]).

bigGAN [24] offers stable training for high-resolution images by integrating effective architectural components from previous GAN models. It inherits spectral normalisation [67] and self-attention layers [107]. This study implemented a 3D adaptation of bigGAN with around 4 million trainable parameters. This number of parameters was limited due to memory constraints.

3.5 Normalising Flows

3.5.1 Context

Flow-based methods are part of likelihood-based methods (similar to GANs and VAEs). Even though GANs and VAEs have shown pretty good results in many demanding tasks. Because none enables precise assessment of the probability density of new points, they have some drawbacks [58]. This can cause difficulties in the training process due to phenomena like mode collapse (G maps different values to the same output, resulting in similar generated images), posterior collapse (the latent space representation in VAEs becomes unresponsive to changes in the input data), and vanishing gradients (gradients in

neural networks become small to the point where they have no effects on weight updates anymore) [23].

3.5.2 Introduction

Normalising Flows (NFs) are a family of generative models where both sampling and density evaluation can be precise and exact because of their tractable distributions [58].

An NF transforms a simple probability distribution into a more complex one by a sequence of invertible and differentiable mappings. Each step changes the data in an easy way to undo. These small changes allow us to calculate how much and in what direction to go each step to get closer to the desired outcome [58].

The basic concept of this method is that it is possible to explore the probability distribution of the function of a random variable when we have access to the distribution function of the original variable.

After transforming the sample to the original simple distribution, the density of a sample can be calculated by multiplying the density of the inverse-transformed sample under this distribution and the associated change in volume induced by the sequence of inverse transformations. By obtaining the absolute values of the Jacobian determinants for each transformation that is carried out and computing their product, the volume change can be computed [58].

Following these procedures and changes, we are left with a technique to build new distribution families.

3.5.2.1 Formal Application

The random variable $Z \in \mathbb{R}^D$ has the probability density function $p_Z : \mathbb{R}^D \rightarrow \mathbb{R}$. Let $Y = g(Z)$ and g be the AN invertible function.

With this, we can calculate the probability density function of the random variable Y using the change of variables formula (see Equation 3.21):

$$p_Y(y) = p_Z(f(y)) |det Df(y)| = p_Z(f(y)) |det Dg(f(y))|^{-1}, \quad (3.21)$$

, where g 's inverse is denoted by f , f 's Jacobian is $Df(y) = \frac{\delta f}{\delta y}$ and g 's Jacobian is $Dg(z) = \frac{\delta g}{\delta z}$.

The function g called the G , takes a simple random noise distribution (p_Z) and converts it into a more complex one. This is the generative direction of the model.

The function f , on the contrary, acts as an inverse function that converts a complex and irregular data distribution into a simpler one, p_Z . That is why the function works in the opposite direction of g . This process gives the name to “normalising flows” because the main objective of f is to normalise the distribution of data.

Given fair assumptions about these distributions, the hypothesis is that if the transformation function g is complex enough, from any base distribution p_Z , it can create any distribution p_Y .

However, for it to work, we need to find completely invertible functions, which is one of the difficulties of this type of approach.

3.5.3 Generative Flows - Glow

Glow is a generative flow [56], built upon Non-linear Independent Components Estimation (NICE) and Real-valued Non-Volume Preserving (RealNVP) that were proposed in [28, 29], and it consists of a series of flow steps built in a multi-scale architecture (see Figure 3.35).

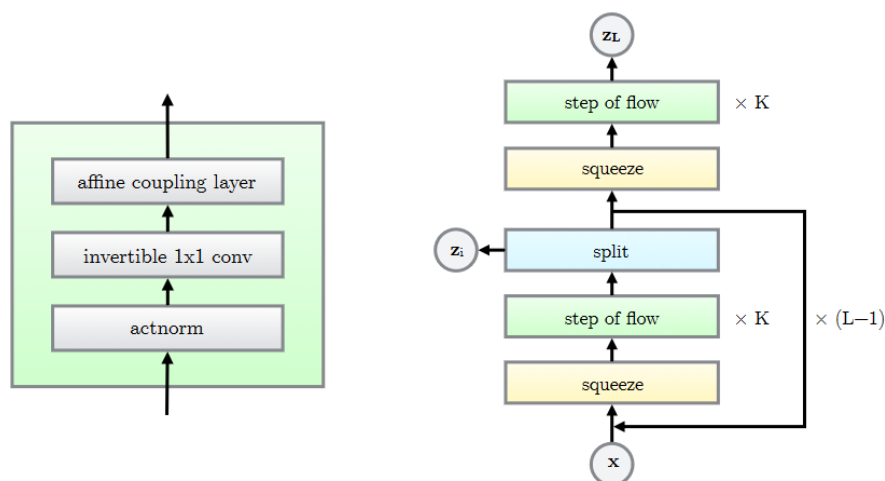


Figure 3.35: Each step of the generative flow (left) consists of a *actnorm* step, followed by an invertible 1×1 convolution and an affine coupling layer [28], which is then combined with a multi-scale architecture (right) [29], (from [56]).

3.5.3.1 Actnorm

In the work [29], batch normalisation was suggested to tackle issues found in the training of the deep models. However, the variability introduced in activations becomes higher the smaller the size of the data batches processed by each computing unit. This then leads to performance issues.

To fix this problem, they proposed an *actnorm* layer for activation normalisation, which uses a scale and a bias parameter for each channel to regulate the activations. Scale and bias are also set up initially to ensure that the post-*actnorm* activations have a mean of zero and a variance of one compared to the initial batch of data.

3.5.3.2 Invertible 1x1 convolution

Dinh et al. [28, 29] proposed a fixed method that rearranged the order of the channels. Kingma et al. [56] propose a more flexible method with learned 1×1 convolutions where the weight matrix is two as a random rotation matrix.

This new proposition allows the rearrangement of channels in a flexible but mathematically controlled way.

Kingma et al. [56] also found a way to make the calculations computationally less costly, especially when presented with many channels.

3.5.3.3 Affine layer

The affine layer is introduced as a powerful and reversible transformation, with computationally efficient forward, reverse, and log-determinant functions [28].

In [56], the last convolution of each neural network was initialised with zeros, initially making each affine coupling layer perform an identity matrix. This was found to help the training of very deep neural networks. The architecture of the split() functions proposed by Dinh et al. [28, 29] was also simplified by performing splits only along the channel dimension.

Before each flow step, it is essential to reorder the variables such that each dimension can effectively impact the other dimensions. Dinh et al. [28, 29] introduced a method that reverses channels' order before applying the additive coupling layer. Kingma et al. [56], instead, introduced an invertible 1×1 convolution as generalisation of those permutations.

3.6 Diffusion Models for 3D Medical Imaging

Probabilistic models usually suffer a trade-off between two conflicting objectives: tractability and flexibility.

To tackle these problems, Sohl-Dickstein et al. [86] propose an innovative method for defining probabilistic models, offering exceptional flexibility in model structure, precise sampling, straightforward combination with other distributions, and efficient evaluation of the model's log-likelihood and the probability of individual states.

The model aims to establish a forward diffusion process, transforming a complex data distribution into a simpler, manageable one, followed by learning a reverse diffusion process that defines the generative model distribution. The forward process gradually converts the data distribution to a more analytically tractable distribution using a diffusion kernel. The reverse process is trained to describe the same trajectory in reverse [86].

The learning process involves estimating mean and covariance for Gaussian diffusion kernels and bit flip probability for binomial kernels. The generative model's probability is evaluated by averaging forward and reverse trajectories. The training objective is to

maximise log-likelihood by estimating reverse Markov transitions. The method allows easy multiplication of distributions, which is crucial for image denoising or inpainting tasks [86].

The approach establishes a method for converting complex data distributions into simpler ones using diffusion processes and then learning the reverse process to define generative models, offering insights into probability estimation and distribution manipulation [86].

3.6.1 DPMs

Diffusion Probabilistic Models (DPMs) [43, 86] can create MRIs from random noise, by repeatedly translating data gradually to noise (forward diffusion process) and then, doing the reverse (also known as the reverse diffusion process).

The forward diffusion process transforms real data into Gaussian noise over multiple steps, gradually introducing noise through a Markov chain. This process enables sampling at any specific step without iterating through the entire chain [43, 86]. The reverse diffusion process, on the other hand, attempts to create realistic data from random noise. By going through the complete chain backward and estimating the distribution's mean using a neural network that minimises the difference between estimated and true values by applying the L2 norm [43, 86], it approximates the posterior distribution. Figure 3.36 shows an example of the application of DPMs.



Figure 3.36: LSUN [106] Church and Bedroom samples, (from [43]).

For the quantitative results, cDPM ended up performing the best.

3.6.2 Conditional Diffusion Probabilistic Model

To overcome the need of standard DPMs for extensive computational resources models, the Conditional Diffusion Probabilistic Model (cDPM) was proposed [74]. It works by training a 2D cDPM to produce sub-volumes of the brain MRI. The model then learns the relationships between those slices, regardless of distance. This can be done with computational efficiency by using an attention network. These learned relationships are then used to generate a new, anatomically accurate 3D brain MRI by iteratively applying the cDPM [74]. These cDPMs, like in [43], use a U-Net (see Figure 3.37).

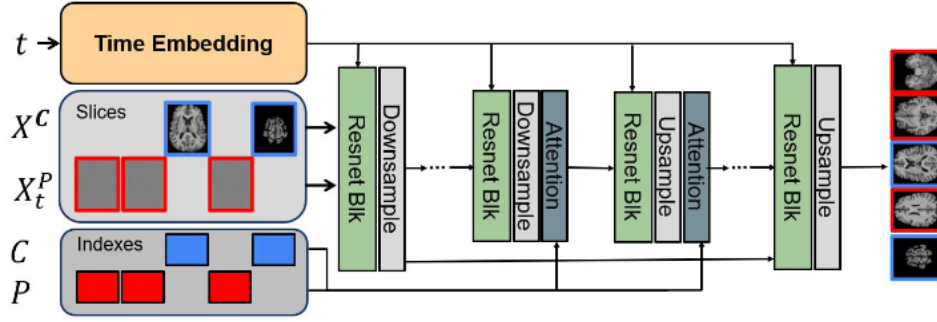


Figure 3.37: The architecture of cDPM is composed by a U-shape NN with skip-connections. t , C, P , X^C , and X_t^P represent, respectively, the input at step t , the slice indexes, the condition sub-volume, and the target sub-volume, (from [74]).

A multi-head self-attention mechanism [94] was also added to model the relationships between the slices. After training the cDPM, the 3D MRI is generated by performing N stages. The initial set of slices is produced from random noise. Then, conditioned on those synthetic slices, the cDPM runs again to produce a new set of slices. This process of generating slices based on other slices generated in prior stages is repeated until an entire 3D MRI is created [74].

1262 t1-weighted brain MRIs of the participants were used in the tests. These were derived from three distinct datasets: SRI International [108], UCSF (PI: V. Valcour), and the Alzheimer’s Disease Neuroimaging Initiative (ADNI-1) [52].

Figure 3.38 shows the qualitative results of the cDPM model compared with other baselines.

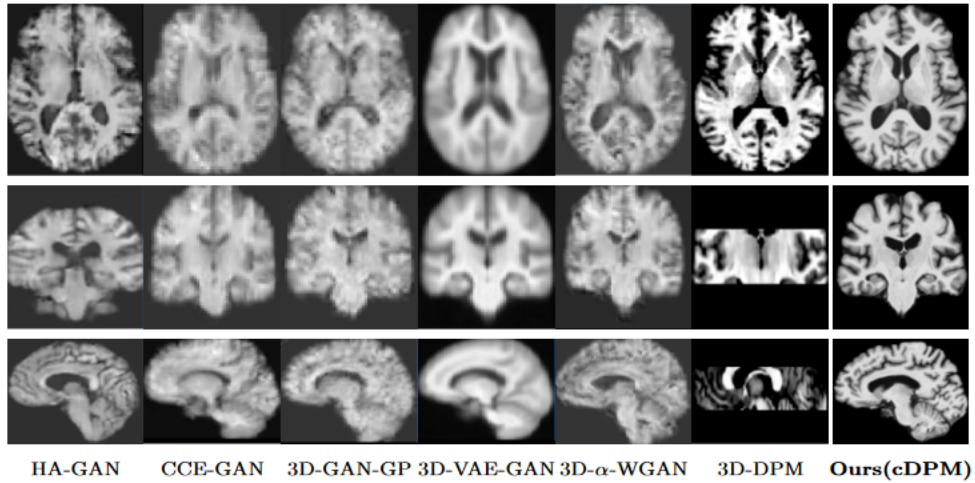


Figure 3.38: Three MRI images produced by seven models. As we can see, cDPM generates the most realistic results, (from [74]).

3.6.3 Sequence-Aware Diffusion Model

Sequence-Aware Diffusion Model (SADM) [105] is a model that tackles issues like high dimensionality, missing data, and fluctuations in sequence length while creating longitudinal medical images. SADM conditions a diffusion model using a transformer-based attention module. The attention module is designed to generate the conditioning signals for the diffusion model [105] (see Figure 3.39).

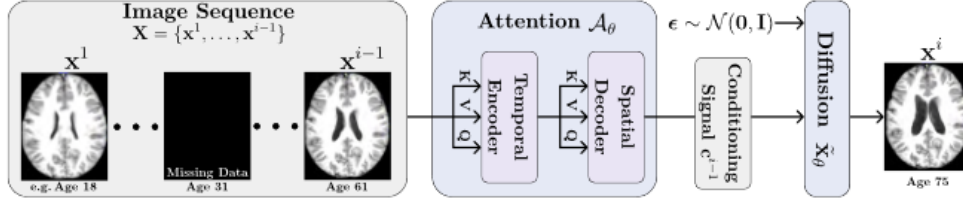


Figure 3.39: An overview of the SADM model, (from [105]).

SADM's architecture consists of an attention module A_θ composed of a token embedding, a temporal encoder, and a spatial decoder. The token embedding utilises a 4D convolution window for image tokenisation. As the temporal resolution of medical images tends to be lower than spatial resolution, it employs a non-overlapping linear projection window and sets temporal stride = 1 [17] for computational efficiency. The temporal E uses factorised transformers [17] to calculate self-attention over tokens that share identical spatial indices [105]. An MLP then reduces token dimensionality for each temporal transformer block. The spatial decoder reshapes temporal E outputs into spatial tokens, facilitating self-attention across spatial dimensions within the same temporal index.

It also incorporates autoregressive sampling to handle temporal dependencies better during inference. Training involves unobserved indices for target images and prior indices for conditioning, greatly enhancing generative performance in modelling medical image sequences over time [105].

For the experiments, a cardiac MRI dataset from the Automated Cardiac Diagnosis Challenge (ACDC) organisers [21] was used as well as a brain dataset [33] composed of 2851 subject scans evenly distributed in age [105].

For the implementation, a classifier-free diffusion model [44] architecture was chosen as well as its hyperparameters.

Figure-3.40 shows the qualitative comparisons with the baseline models.

For the quantitative comparisons, SADM outperformed other GAN and Diffusion-based models [105].

3.6.4 MedGen3D

MedGen3D [40] is a pioneering deep generative framework for producing paired 3D medical images and masks. It treats medical data as sequential slices using an autoregressive

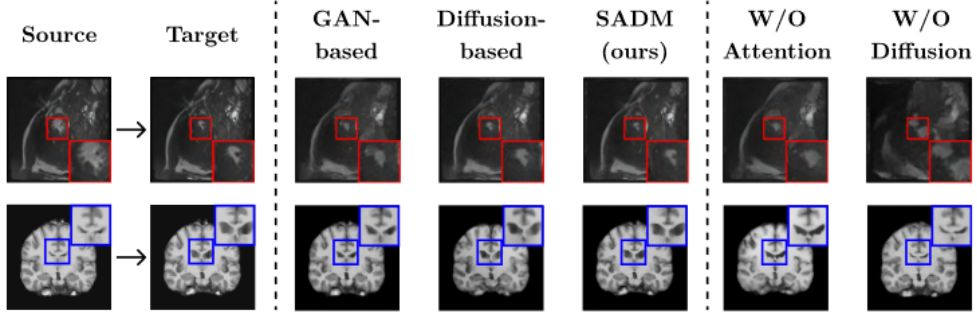


Figure 3.40: Qualitative comparison between baselines, (from [105]).

process. Using slice indices to support anatomical structure consistency, a Multi-Condition Diffusion Probabilistic Model (MC-DPM) combines conditional and unconditional generation processes to create mask sequences in the first stage. To create realistic medical images aligned with masks, the second stage introduces a conditional image G that uses a semantic diffusion refiner and a seq-to-seq model from [97]. This model's primary contributions include solving the issue of synthesising full 3D volumetric medical images along with their matching masks, introducing a multi-condition diffusion probabilistic model for producing 3D anatomical masks that are diverse and highly accurate, and using masks to generate accurate images [40].



Figure 3.41: An overview of the proposed MedGen3D is provided, which consists of a 3D mask G that will generate the mask sequence autoregressively from a random position and a conditional image G that will construct 3D images conditioned on created masks. z , (from [40]).

The model's training involves generating mask sequences and conditioning slices, minimising a specific loss function (see Equation 3.22).

$$Loss(\theta) = E_{X_0 \sim q(X_0), \epsilon \sim N(0, I), p^C, z, t} [\|\epsilon - \epsilon_\theta(X_t^P, X^C, z, t)\|^2] \quad (3.22)$$

where, θ represents the parameters of the model that are being optimised during training, $E_{X_0 \sim q(X_0), \epsilon \sim N(0, I), p^C, z, t}$ denotes the expected value over the distributions of the initial data X_0 , the noise ϵ , and other variables p^C, z, t . Then $\epsilon_\theta(X_t^P, X^C, z, t)$ represents the model's prediction of the noise ϵ , given the inputs X_t^P, X^C, z , and t . Finally, $\|\epsilon - \epsilon_\theta(X_t^P, X^C, z, t)\|^2$ is the squared error between the true noise ϵ and the model's predicted noise, which is what the model is trying to minimise.

During inference, MC-DPM generates mask sequences from noise and expands them based on existing slices. In the second step, an image sequence G produces initial images based on masks using Vid2Vid [97]. Further refinement is achieved using a Semantic

Diffusion Refiner (SDM) based on diffusion probabilistic models, addressing issues like blocking and blurriness in generated images. The SDM refines images aligned with input masks, improving image quality [40].

Three thoracic CT datasets and two brain MRI datasets at the brain location were used in the trials. These were divided into the generative models and the downstream segmentation tasks [40].

- Generative models:
 - SegTHOR [60]: 3D CT scans of the thorax;
 - OASIS [64]: 3D MRI T1 of the brain.
- Downstream segmentation tasks:
 - 3D CT datasets of the thorax StructSeg-Thorax [9] and Public-Thor [27];
 - 3D MRI T1 dataset of the brain, from ADNI [1],

Figure 3.42 shows the qualitative results.

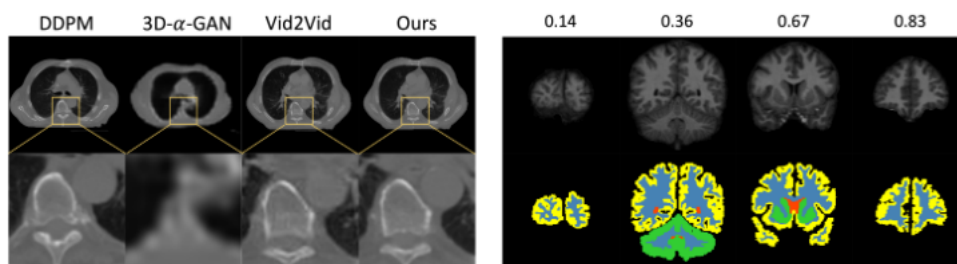


Figure 3.42: Left: Comparing various generative models qualitatively. Right: An illustration of a 3D brain MRI slice in synthetic form with varying relative placements, (from [40]).

3.6.5 3D medical image generation with DPMs

Medical diffusion [55] investigates how diffusion models might be used to produce three-dimensional medical data. We provide a novel architecture for a diffusion model that functions on the three-dimensional latent space of medical images. Three distinct datasets containing CT and MRI data of several anatomical locations, including the knee, chest, and brain, are used to train the model. The study investigates whether segmentation algorithms are improved especially in situations with little data by pre-training on these artificial images.

The study demonstrates the robustness of medical diffusion model training. The diffusion models were trained, as previously mentioned, using publicly available datasets from four distinct anatomic regions: brain MRI studies from the Alzheimer’s Disease Neuroimaging Initiative (ADNI) [1], chest CT studies from the Cancer Imaging Archive

(Lung Image Database Consortium (LIDC) [16]), breast MRI studies from Duke University (DUKE) [83], and knee MRI studies from Stanford University (MRNet) [55]. These datasets were specifically selected to investigate our method’s capacity to produce synthetic data in low-data environments. The models produced realistic 3D medical pictures despite the magnitude of the dataset, without experiencing mode collapse [55] or needing to alter hyperparameters for different datasets (see Figure 3.43).

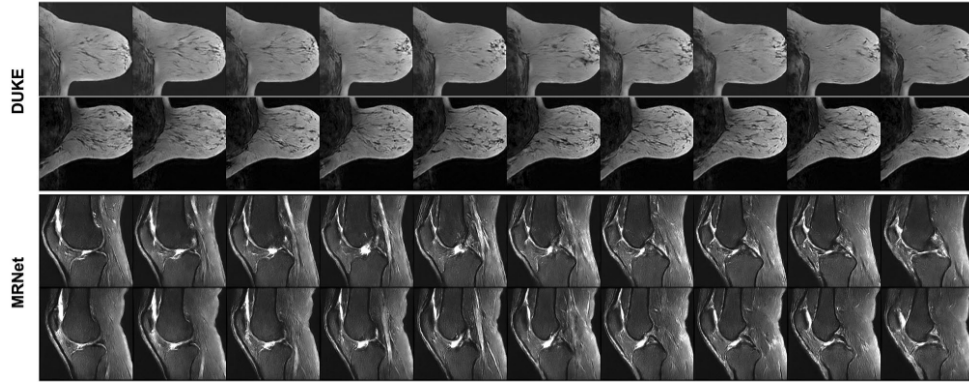


Figure 3.43: Generated data samples from two public datasets that were utilised in the training of the diffusion model, (from [55]).

Finding better metrics for assessing the image quality of synthetic images is still an open problem, especially on images such as these, where the smallest of details is very important. To overcome this, human experts rated the synthetic images as largely realistic, consistent between slices, and exhibiting minor anatomic inconsistencies [55].

A variation of the Vector Quantised Variational Autoencoder (VQVAE), VQ-GANs [32] improves image production by combining perceptual loss, GAN-based adversarial training, and vector quantisation to preserve high perceptual quality at high compression rates.

This study also examines how varying those compression factors impact image quality in VQ-GANs [32] which rely on methods that leverage the power of a D and a vector quantised latent space, producing higher-quality image reconstructions, in particular less blurry images, compared to conventional VAEs. A compression factor of 8 led to the loss of relevant anatomical features, while a compression factor of 4 resulted in a more accurate reconstruction of these features. Across all four datasets, a maximum compression factor of 4 retained precise anatomical details [55] (see Figure 3.44).

The study also compared a diffusion model against a Wasserstein GAN with gradient penalty (WGAN-GP) in generating brain MR images of resolution $64 \times 64 \times 64$ [55].

Finally, the study demonstrates the use of synthetic data in breast MRI segmentation, generating 2000 synthetic breast MR images and pre-training a Swin UNETR [93] network. The network was finetuned with the available segmentation data resulting in improved performance and a substantial enhancement in the Dice similarity coefficient [55].

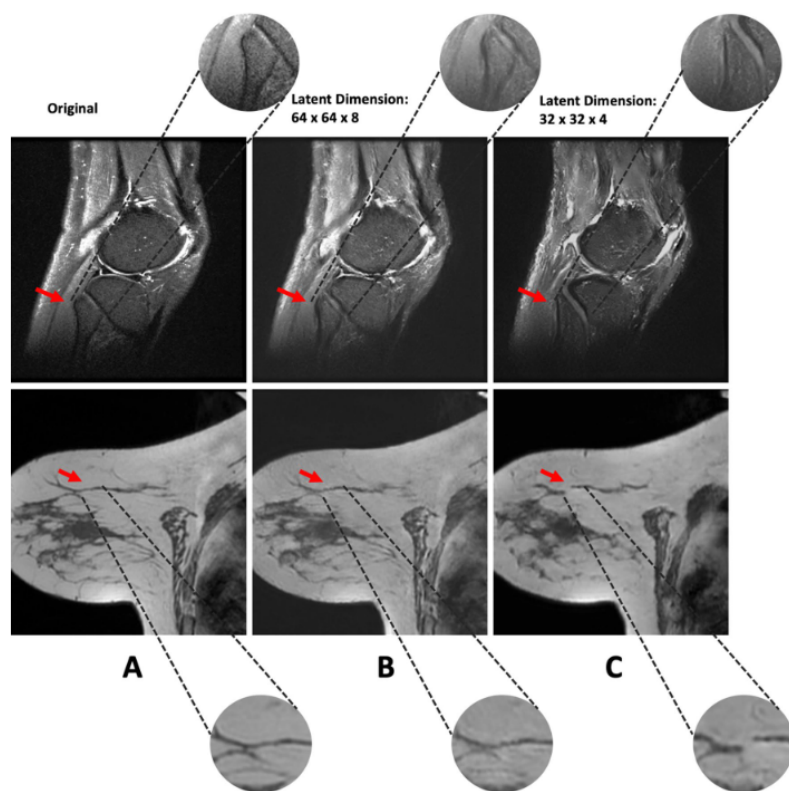


Figure 3.44: The effects of different compression factors in the reconstruction quality for two distinct samples using the VQ-GAN autoencoder, (from [55]).

3.7 MedSyn

MedSyn [104] introduces an innovative approach to the generation of 3D lung CT images with high-quality guided by textual guidance. One of the main problems with medical image generation is related to the outputs that produce low-resolution images and under-utilise information in radiology reports, even in the latest state-of-the-art diffusion models. This work uses a hierarchical scheme with a modified U-Net [80] architecture to fix this. It uses text to create low-resolution visuals that serve as a basis for later G's to finish volumetric data. Additionally, it produces segmentation masks for the main anatomical structures to ensure anatomical accuracy (see Figure 3.45).

The proposed model is composed of three components:

- **Text-encoder:** Which is pre-trained and used to extract language features from radiology reports;
- **Low-Resolution 3D diffusion model:** Which is text-guided, used for the generation of CT volumes and its anatomical structure volumes;
- **Super-Resolution 3D diffusion model:** To supplement the missing anatomical information and scale up the low-resolution produced volumes [104].

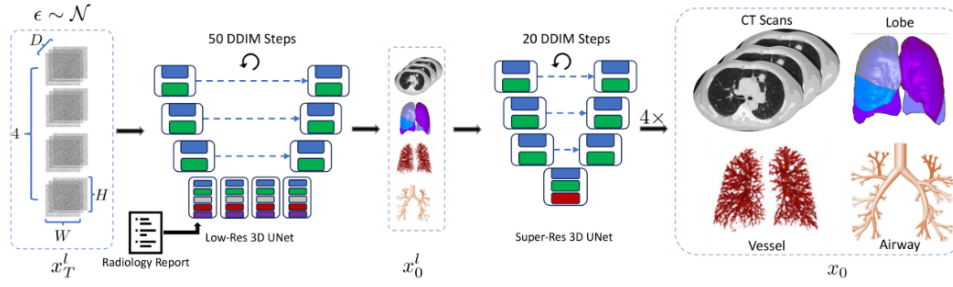


Figure 3.45: Overview of the generative model. Using Gaussian noise, represented by ϵ and a radiology report, a $64 \times 64 \times 64$ volume with low-resolution and its anatomical structures are generated to start the procedure. These volumes give way to a more detailed resolution of $256 \times 256 \times 256$, (from [104]).

3.7.1 Medical BERT

The authors trained a text encoder called Medical BERT, which translates information from CT radiology reports into high-dimensional vectors (f_{text}). This model was fine-tuned by concentrating on essential sections like “Impression” and “Findings” of CT reports because they contain the most distinctive specifications of the symptoms of each disease and anatomical structures; this was done using tasks such as mask language modelling and paired section prediction [104]. These tasks involve masking tokens and shuffling sections to create paired and unpaired samples. Leveraging unique embeddings, a classifier was trained to identify whether the sections mentioned before are paired or unpaired. See Figure 3.46 for an overview of the low-res generative model and the application of the Medical BERT encoder [104].

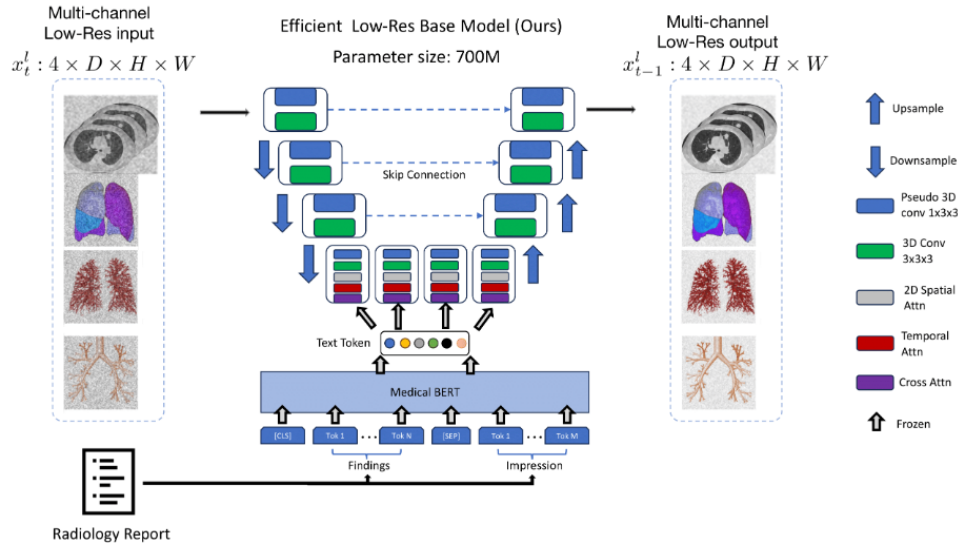


Figure 3.46: Effective low-resolution generative model with input of clinical tokens, (from [104]).

3.7.2 Text-Conditioned Volume Generation

One problem of directly training the model on $256 \times 256 \times 256$ is that it is very memory-costly and therefore not the best approach. Yanwu Xu et al. [104] proposed a model with a two-phase hierarchical process to fix this [104]. The first phase includes generating a $64 \times 64 \times 64$ low-resolution volume conditioned by radiology reports, and the second one the upscaling to $256 \times 256 \times 256$. The first process involves a text-conditioned diffusion model for low-res images, and the second involves a lightweight U-Net based super-resolution module for upscaling. The super-resolution process focuses on denoising without incorporating text information to save computation [104].

3.7.3 Joint Generative Modelling

CT scans of the lungs present way more important details than all the other anatomical structures of the body [104]. These anatomical details, often vaguely described in clinical reports, are crucial but overlooked in generative modelling. Yanwu Xu et al. [104] proposed a joint model that generates volumes and anatomical structures conditioned on radiology reports to address this issue. Firstly, the shape information for every component is extracted using segmentation tools: lungmask [46], TotalSegmentator [100], and Navi-Airway [96]. Additional channels are added to the input and output layers to incorporate this into the existing model. These channels are concatenated, forming a diffusion process on the resulting 3D volumes [104].

3.7.4 Anatomically Controllable Synthesis

The suggested model differs considerably from existing conditional models, such as Control-Net [109], as it models volume generation alongside structure information, whereas Control-Net is conditional on structure. This allows MedSyn to generate data even when specific structures are not available. However, by discriminating one or more elements in the joint denoising process, we can get the same results as Control-Net [104].

3.7.5 Efficient 3D Attention U-Net

A new base neural architecture was proposed to fix the memory issues of generating such high-resolution images. Compared to the standard 3D attention U-Net [45], an encoder-decoder with convolutions and the shift of all attention mechanisms to the bottom of the U-Net was proposed [104].

This allows for a reduced computational load demanded from the attention mechanisms, improving significantly the computational efficiency, even though it also highly increases the number of parameters. Almost all attention modules are deleted in the super-resolution network, except for the temporal attention module at the U-Net's, which allows it to handle $256 \times 256 \times 256$ volume inputs [104].

3.7.6 Datasets and Results

The experiment made use of a large-scale 3D dataset of 8,752 participants' 3D thorax CT images and related radiology reports. The dataset also comprises 209,683 reports without associated images. It was acquired from the University of Pittsburgh Medical Centre [104].

Figure 3.47 shows the qualitative results.

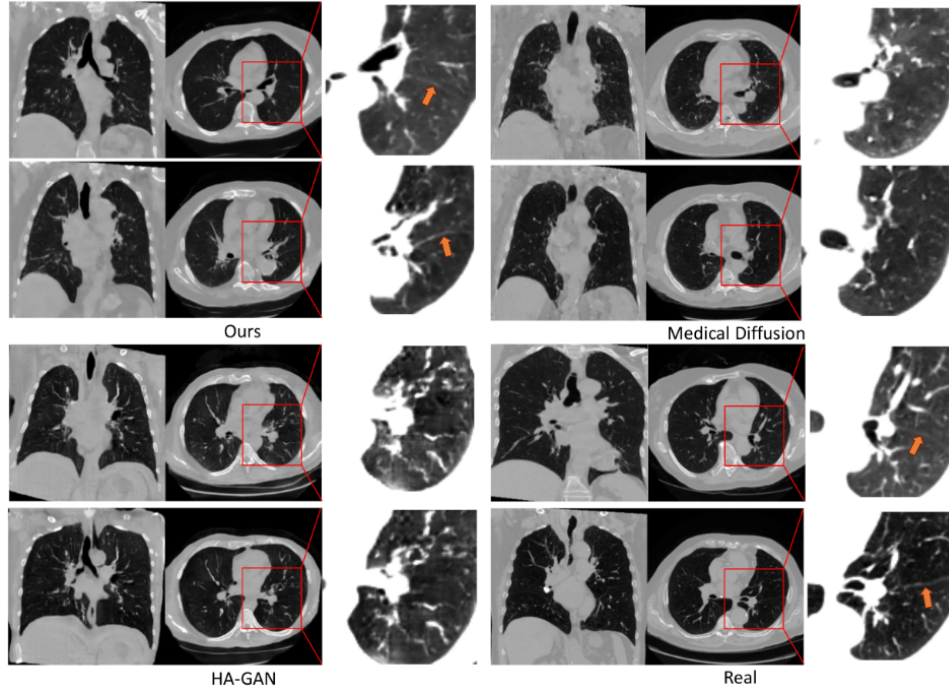


Figure 3.47: Real and generated lung images from HA-GAN [90], MedSyn, Medical Diffusion. Only the MedSyn model could preserve the smallest anatomical characteristics, including fissures, as shown by the arrows, (from [104]).

3.8 Evaluation Metrics

Since many approaches are available for GAN evaluation, none of them can be considered optimal, making it an open problem. These fall into two categories: objective measurements, which include similarity scores and distances, and perceptual investigations, which call for human observers to evaluate and distinguish between synthetic and genuine samples. This section addresses perceptual research in addition to the most used metrics.

3.8.1 Inception Score

Introduced in [84], every produced sample is subjected to the Inception Model [91] by the IS to obtain the conditional label distribution $p(y|x)$. This label distribution with low entropy should be present in samples that contain meaningful items. Since the model should have a wide range of outputs, the marginal $\int p(y|x = G(z))dz$ should have a large

entropy. Combining both, we get the IS 3.23.

$$\exp(\mathbb{E}_x KL(p(y | x) \parallel p(y))) \quad (3.23)$$

where higher values denote higher quality and more varied samples.

3.8.2 Mode Score

Mode score (MS) [25] is an improvement to the Inception Score (IS) metric, where the MS measures the difference between the produced distribution and the true distribution (see Equation 3.24).

$$\exp(\mathbb{E}_x KL(p(y | x) \parallel p(y)) - KL(p(y) \parallel p(y^*))) \quad (3.24)$$

3.8.3 Fréchet Inception Distance

Fréchet Inception Distance (FID) [42] also stands as an improvement to IS. The model samples distribution is computed by calculating the Fréchet distance between two multi-variate Gaussians with mean and covariance (m, C) , and for the real dataset distribution, (m_w, C_w) (see Figure 3.25).

$$\| m - m_w \|^2_2 + \text{Tr}(C + C_w - 2(CC_w)^{1/2}) \quad (3.25)$$

3.8.4 Structural Similarity

Structural Similarity (SSIM) [99] is used to measure the similarity between a pair of images, and it is measured based on three different factors, which are: luminance (l) (see Equation 3.27), contrast (c) (see Equation 3.28), and structure (s) (see Equation 3.29). Considering two images x and y the SSIM can be calculated using Equation 3.26:

$$[l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma \quad (3.26)$$

where the exponents α , β , and γ are parameters used to adjust the importance of each component and

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (3.27)$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (3.28)$$

$$s(x, y) = \frac{2\sigma_{xy} + C_3}{\sigma_x + \sigma_y + C_3} \quad (3.29)$$

where μ_x and μ_y are the means, σ_x and σ_y are standard deviations, σ_{xy} is the correlation coefficient and C_1 , C_2 and C_3 are constants.

3.8.5 Multi-Scale Structural Similarity

Multi-Scale Structural Similarity (MS-SSIM) [98] is an improvement on the SSIM metric by analysing picture information at many resolutions as opposed to only one. The metric is as follows: two photos are used as input, a low-pass filter is applied interactively, and the input is downsampled by a factor of two. Luminance (l), contrast (c), and structure (s) of the pictures are used by the MS-SSIM in the same way as the SSIM; however, luminance is only computed after $M - 1$ repetitions, where M is the highest scale and $M = 1$ is the original image. The final score is given by Equation 3.30:

$$[l_M(x, y)]^{\alpha_M} \cdot \prod_{j=1}^M [c_j(x, y)]^{\beta_j} [s_j(x, y)]^{\gamma_j} \quad (3.30)$$

where the exponents α , β and γ are used to adjust the importance of the components.

3.8.6 Maximum-Mean Discrepancy Score

Maximum-Mean Discrepancy Score (MMD) [38] objective is to create a statistical test to determine if two random variables, x and y , characterised by probability measures p and q respectively, are not equal. This test is based on observations X and Y , drawn independently and identically distributed from p and q . The key question is whether these probability measures are unequal. Using an unspecified function class F , the MMD measures the difference between p and q . It evaluates the difference between the expectations of functions from F when applied to x and y (see Equation 3.31):

$$MMD[F, p, q] := \sup_{f \in F} (E_x[f(x)] - E_y[f(y)]) \quad (3.31)$$

where F is a class of functions such that $f : X \rightarrow \mathbb{R}$ and X and Y be observations of x and y , respectively.

3.8.7 Peak Signal-To-Noise Ratio

Peak Signal-To-Noise Ratio (PSNR) [8] is a formula for the relationship between a signal's maximum value and the amount of noise that distorts it and lowers the representational quality of the signal. The PSNR's mathematical representation is depicted by Equation 3.32:

$$PSNR = 20 \log_{10} \left(\frac{MAX_f}{\sqrt{MSE}} \right) \quad (3.32)$$

where MSE is given by Equation 3.33:

$$MSE = \frac{1}{(m * n)} \sum_0^{m-1} \sum_0^{n-1} \| f(i, j) - g(i, j) \|^2 \quad (3.33)$$

and f represents the matrix data of our original image, g represents the matrix data of our degraded image in question, m represents the number of rows of pixels of the images, and i represents the index of that row, n represents the number of columns of pixels of the image, and j represents the index of that column, and MAX_f is the maximum signal value that exists in our original “known to be good” image [8].

3.8.8 Normalised Root Mean Square Error

Normalised Root Mean Square Error (NMRSE) [89] is a measure used to assess the accuracy or discrepancy between observed and predicted data after a normalisation process. It’s calculated as the Root Mean Square Error (RMSE) normalised by the average of the observed data. The formula for NMRSE is given by Equation 3.34:

$$NMRSE = \frac{RMSE}{\text{Average of observed data}} \quad (3.34)$$

where RMSE is given by Equation 3.35:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (O_i - P_i)^2} \quad (3.35)$$

This measure helps assess the normalised error, allowing for comparison and evaluation of the accuracy of predictions relative to the observed data. A lower NMRSE indicates a better fit between observed and predicted values after normalisation [89].

3.8.9 Learned Perceptual Image Patch Similarity

Learned Perceptual Image Patch Similarity (LPIPS) [5] calculates the perceptual similarity between two images. For a given network, LPIPS basically calculates how similar two image patches activation’s are to one another. It has been demonstrated that this measure closely resembles human perception. Perceptually comparable image patches get a low LPIPS score.

3.8.10 Visual Turing Test

The Visual Turing Test (VTT) [34] consists of posing binary questions to humans to gauge how well the system can identify items, characteristics, and connections in pictures. In the context of this paper, the test is considered successful if the humans in question cannot distinguish between the system’s outputs and reality.

3.9 Summary

This chapter explores and discusses the main types of generative models. Although autoregressive models are mostly employed for picture completeness or frame interpolation, they

are inefficient to train and can provide high-quality images. VAEs provide hazy, unreal visuals but are effective at training and sampling. Currently, GANs are one of the most popular methods in terms of output quality, and due to their popularity, they are the most studied framework. However, the main drawbacks of GANs are their challenging training requirements and the absence of a consistent and trustworthy evaluation criterion, which might result in subjective visual assessment.

Normalising Flows have become highly effective instruments in creating medical images, providing benefits in modelling intricate data distributions and permitting manageable probability estimation. Their capacity to produce adjustable latent regions in high-fidelity medical images suggests intriguing uses in treatment planning and diagnostic research. However, large computational resources and complex structures are frequently needed for these models to be trained and inferred effectively.

The evolution of 3D GANs such as the ones discussed in this chapter has significantly contributed to advancing medical image synthesis in three dimensions. These models offer varying strengths, from preserving anatomical accuracy to generating high-resolution and diverse medical images. One drawback of this type of models is the trade-off between image resolution and training stability which remains a concern.

Diffusion models have the innate capacity to represent intricate data distributions, facilitating the creation of highly realistic high-resolution medical images that support treatment planning and diagnostic studies. These models do have certain limits, though. Similarly to normalising flows, diffusion models present implementation and data acquisition issues since they frequently require large computational resources and large training datasets.

There is also enormous potential for synthesising high-quality 3D lung CT images under the guidance of textual prompts. Personalised medical imaging could benefit from the textual direction in image production, enabling pathology simulation and accurate anatomical detail. This development may simplify medical diagnosis and treatment plans, allowing for customised patient interventions.

However, there are drawbacks to this strategy as well. For example, reliable natural language processing models must comprehend and correctly translate textual instructions. Maintaining anatomical correctness while maintaining coherence between textual advice and image production is a major difficulty.

Many of the measures used to assess picture generation models and their outputs discussed in this chapter are limited to 2D data and cannot be extended to 3D space. On the other hand, metrics such as the SSIM and the MS-SSIM that assess the data directly can be adjusted to accept volumes as input. Other metrics, like LPIPS, MMD, MSRE, and PSNR, have also been used for quality assessment, which gauges various aspects of image fidelity.

Chapter 4

3D Lung Image Generation

Contents

4.1	Dataset	59
4.2	Hardware	62
4.3	Architectures	62

The creation of the deep learning model for creating lung CT scans is covered in length in this chapter. First, the pre-processing processes and the dataset used to train the model are introduced. A description of the model, its composition, and specifics on its training process come next.

4.1 Dataset

4.1.1 LIDC-IDRI

The LIDC/IDRI Database [15] comprises 1018 helical thoracic CT scans gathered retrospectively from seven academic institutions with appropriate IRB approval. In compliance with Health Insurance Portability and Accountability Act (HIPAA) regulations, anonymisation software was utilised to eliminate protected health information. Scans were not conducted specifically for the database to ensure heterogeneity in scanner type and technical parameters. Inclusion criteria verified relevance to the development of CAD systems. Each included scan's collimation and reconstruction intervals were less than 3 mm.

Radiologists also annotated the images by identifying nodules and non-nodules. Several software programs were employed to record the features and shapes of nodules. The XML format was used to exchange data, and retrospective data on pathology was gathered for 268 scans. This annotation process was done in two different phases [15] :

- **Blind-read phase:** every radiologist examined every scan independently and noted any lesions;

- **Unblinded-read phase:** where the radiologists individually assessed the scan using each other's anonymised marks before forming a final judgment.

The annotation process was also done autonomously and discreetly to identify every potential lesion in each sample without requiring doctors to come to an imposed consensus. It resulted in the following identified nodules [15] :

- 7371 lesions marked as a nodule by at least one radiologist;
- 2669 marked as a nodule $\geq 3\text{mm}$ by at least one radiologist;
- 928 marked as a nodule $\geq 3\text{mm}$ by all four radiologists;
- 1940 marked as a nodule $\geq 3\text{mm}$ or $< 3\text{mm}$ by all four radiologists;
- 2562 marked as either a nodule $\geq 3\text{mm}$ or $< 3\text{mm}$ by all four radiologists.

Figure 4.1 shows examples of lesions that satisfy the LIDC/IDRI definition.

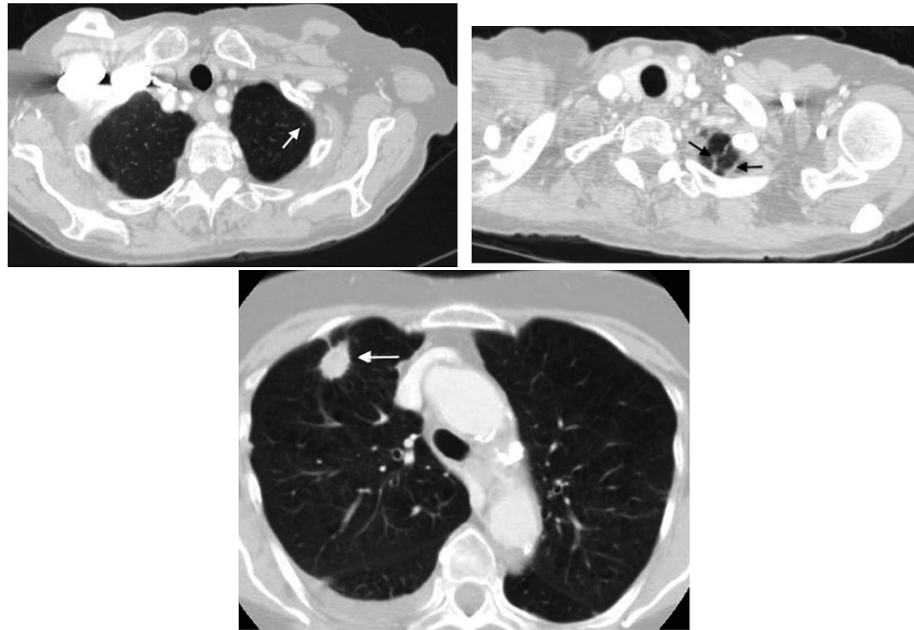


Figure 4.1: Lesions that meet the LIDC/IDRI criteria include those that are categorised as nodules $< 3\text{ mm}$ (top-left), non-nodules $\geq 3\text{mm}$ (top-rigth), and nodules $\geq 3\text{mm}$ (bottom), (from [15]).

4.1.1.1 Pre-processing

To ensure that the model would be trained in the most efficient way, a segmentation of the lungs was performed, to guarantee that all slices used in the training were useful for the model.

Before this segmentation, the Digital Imaging and Communications in Medicine (DICOM) files of each patient were converted to NumPy files where a JSON was also created with all the information available in each of the slices.

The lung segmentation process [87] involved several key steps and considerations:

- **Normalisation and Resizing:** A min-max normalisation technique was initially used to normalise each CT scan. The pixel values, which originally ranged from -1000 to 1000 in Hounsfield units (HU), were scaled to a range of 0 to 1 using -1000 and 400 HU as the lower and upper limits, respectively. The images were then scaled using bilinear interpolation to 128 by 128 pixels to minimise the computational burden while preserving the spatial resolution required for precise segmentation [87];
- **Segmentation Model:** Utilising a hybrid neural network that included the architectures of ResNet34 and U-Net, lung segmentation was performed. This network was selected due to its demonstrated ability to produce precise lung masks. The model's decoder incorporated U-Net blocks for feature concatenation and upsampling, while the encoder portion was made up of residual blocks configured according to the ResNet34 architecture. The last layer created the probability map for lung segmentation using a 1×1 convolution and a sigmoid activation function [87].

With those masks generated, it was easy to see which slices were useful or not by checking if the binary values of the masks of each slice were all zero or not. If they were, then we know that the slice has no relevant information about the lungs and therefore is not useful, being discarded and not used in the training of the models (see Figure 4.2).

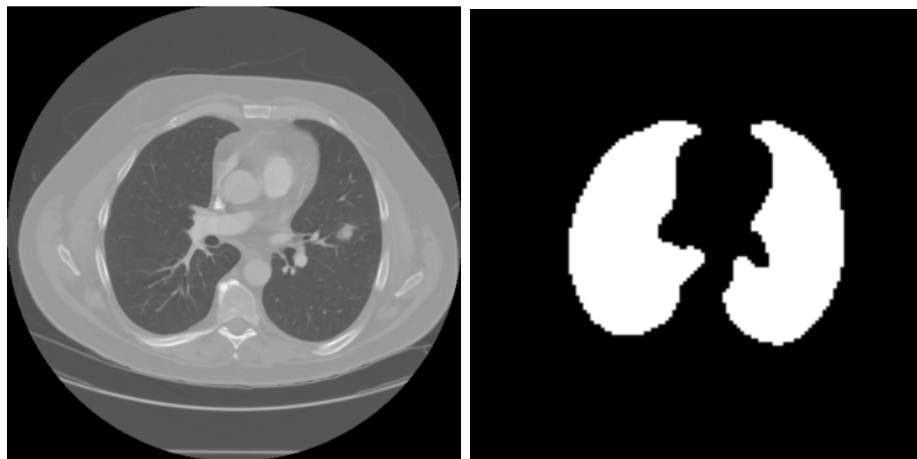


Figure 4.2: In the left image we have the slice without any processing and in the right one the mask generated, as we can see this image is useful for the training of the model.

Min	Max	Average
64	764	241

Table 4.1: Minimum, maximum, and average number of slices per patient scan.

4.1.1.2 Data Preparation and Augmentation

When it comes to the data preparation and augmentation a few key steps were done in order to ensure that the training of the model would be as optimal as possible for the dataset used.

To have a better idea of how the dataset was distributed an analysis of the number of slices of each patient was performed (see Table 4.1).

With this information, and to balance the best resolution possible with reasonable computational power, the number of slices used per patient was set to 64. For patients with more than 64 slices, a uniform sampling is done, to ensure a balanced and unbiased selection.

To further enhance the robustness of the models trained, several data augmentation techniques were used. These included random horizontal and vertical flips, which help the model generalise better by exposing it to different orientations of the same anatomical structures. All images were also resized to 64x64 pixels to ensure uniform input size, and finally, were converted to PyTorch tensors [4] for efficient processing.

4.2 Hardware

The SLURM platform, which is an open-source workload manager for effectively scheduling jobs and managing resources in high-performance computing clusters, handled all of the training and processing needed for this project. Training jobs were executed on 32 GB (NVIDIA Tesla V100), 64 GB (NVIDIA Quadro RTX 6000), and occasionally 80 GB (NVIDIA A100) partitions.

4.3 Architectures

In this chapter, we delve into the implementation details of various generative models tested for 3D medical image generation of the lungs. Our evaluation includes the WGAN [59], its enhanced version α -WGAN [59], and the HA-GAN [90]. Each section explores the model’s architecture, training process, and results, aiming to identify the most effective approach for producing realistic and high-quality synthetic lung images.

4.3.1 WGAN

The WGAN implementation was based on the approach detailed in [59], where the training procedure is simplified and enhanced by incorporating the Wasserstein distance into the original GAN loss function.

4.3.1.1 Structure

We describe the architecture of two distinct models: one designed for $64 \times 64 \times 64$ images and the other for $128 \times 128 \times 128$ images. Both models utilise 3D CNNs and are structured as GANs, consisting of a G and a D.

The G begins with a fully connected layer, transforming a 1000-dimensional noise vector into a 512-channel feature map of dimensions $4 \times 4 \times 4$. This is followed by a series of 3D convolutions with ReLU activations and Batch Normalisation. The feature maps are progressively upsampled using transposed convolutions across four layers, decrementing 128 channels per layer until a single-channel output is achieved. The final layer employs a tanh activation function to ensure the output voxel values are within the range $[-1, 1]$.

The D processes the $64 \times 64 \times 64$ voxel image through a series of four 3D convolutional layers, incrementing from 1 to 512 channels by adding 128 channels at each layer. The last layer reduces the 512 channels to a single-channel output, yielding the final classification score. Each convolutional layer is followed by a leaky ReLU activation with a negative slope of 0.2 and batch normalisation, except for the first layer.

The architecture for the $128 \times 128 \times 128$ model mirrors that of the $64 \times 64 \times 64$ model, with additional layers to achieve a higher resolution.

4.3.1.2 Training

The model was trained with the following parameters:

- **Latent Dimension:** The size of the input latent variable was set to 1000;
- **Batch Size:** A batch size of 4 was chosen to balance training stability and memory constraints;
- **Learning Rate:** Both G and D were optimised using the Adam optimiser with a learning rate of 0.0002;
- **Training Iterations:** The model was trained for a total of 200,001 iterations.

In the training cycle, the D receives a batch of real CT slices while the G creates a batch of fake CT slices from random noise. The D's loss is calculated using the mean scores of real and fake images, incorporating a gradient penalty to enforce the Lipschitz constraint. This constraint ensures that the D's function is bounded to prevent instability

by limiting the rate at which the D can change its output concerning changes in its input. The computed loss updates the D's parameters through backpropagation and optimisation.

The G produces a new set of CT slices, which the D evaluates. The G's loss, calculated as the negative mean score of the fake images, also updates the G's parameters through backpropagation and optimisation.

Model checkpoints were saved every 10,000 iterations to monitor progress.

Two experiments were also conducted:

- **Experiment 1:** Both G and D were updated once per iteration to maintain a balanced training rate;
- **Experiment 2:** G was updated five times per iteration while D's update rate remained unchanged. This was to investigate if allowing the G more updates would enhance its ability to produce better fake samples and prevent the D from overfitting.

4.3.1.3 Summary

As previously mentioned, two experiments were conducted. For qualitative analysis, sequences of slices from random 3D images were selected (see Figure 4.3). For $64 \times 64 \times 64$ images, the visual quality remained consistent with slightly more texture detail in the first experiment. The anatomic structures of the lungs were clearly outlined, although minor airways and pulmonary vessels were not well-defined.

For $128 \times 128 \times 128$ images, there was a noticeable increase in detail. In this resolution, we can clearly outline some of the respiratory airways and vessels mentioned earlier. Both experiments yielded similar visual results, with the second experiment showing slightly more internal lung structure detail.

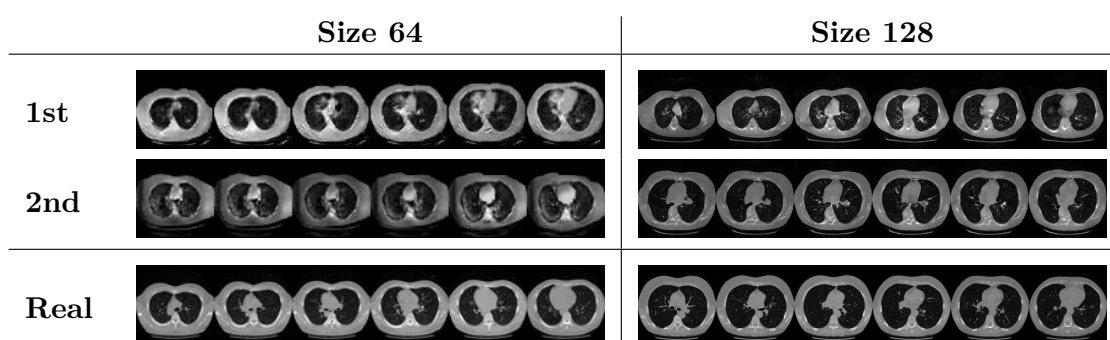


Figure 4.3: Sequence of generated slices using the WGAN architecture, for both the experiments and for both sizes. On the bottom, there is a sequence of real slices for comparison.

Regarding bad, unrealistic results, we also provide examples in Figure 4.4.

For the quantitative analysis, three different metrics were used (see Table 4.2). The MMD metric assesses how the distributions of the generated and real images differ. Better



Figure 4.4: Sequence of unrealistic slices generated by the WGAN model.

similarity to real images is indicated by lower MMD values. The MS-SSIM evaluates the structural similarity between generated images where a higher MS-SSIM value denotes better structural similarity. Lastly, the FID score calculates the distance between the generated and real feature vectors, where lower scores indicate more resemblance to real images and higher image quality.

For $64 \times 64 \times 64$ images, the second experiment showed a slight increase in MMD, with FID and MS-SSIM values remaining similar. For $128 \times 128 \times 128$ images, the metric changes were insignificant, reflecting similar visual outcomes. When comparing the different resolutions, although the $128 \times 128 \times 128$ images were visually more accurate, the MMD values were higher than the $64 \times 64 \times 64$ images. This can happen because larger images with higher resolutions contain more detailed features which can lead to higher MMD values even if the images are visually more accurate. On the other hand, the FID values are lower, complying with the visual improvement.

Table 4.2: Average quantitative results for WGAN.

Experiment	MMD $\times 10^{-4}$ ↓	3D-MS-SSIM ↑	FID ↓
64 - 1st	3.48	0.997	190.8
64 - 2nd	4.08	0.997	189.4
128 - 1st	16.85	0.436	178.0
128 - 2nd	16.84	0.431	178.6

When we look at the values of the MS-SSIM 4.2, we can also draw some conclusions. Given the very high MS-SSIM for the generated $64 \times 64 \times 64$ images, we can infer that mode collapse is happening. Mode collapse is a common issue associated with the training of GANs where the G, rather than producing a wide range of samples that fully represent the variety in the training data, consistently generates a narrow range of outputs that are very similar. This can be caused by unstable training dynamics, overfitting, and other possible reasons. The second experiment also originated as an effort to fix this issue, however this was not achieved. This hypothesis was then confirmed by visual analysis, as seen in Figure 4.5. Even though mode collapse is happening, it does not exactly mean that the generated images have poor quality and unwanted noise, which is typical in the presence of mode collapse. In this case, the G simply had variability problems and not quality problems, as confirmed by visual analysis.

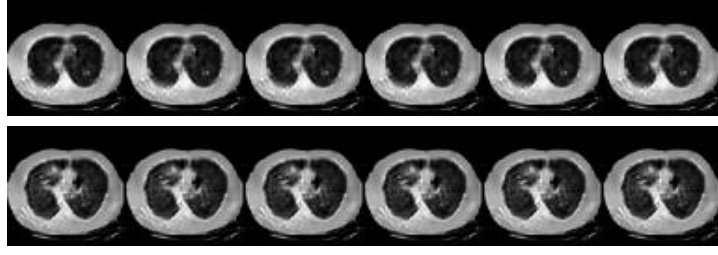


Figure 4.5: Each sequence of slices represents the same generated slice for 6 different patients, and as seen, they are all almost completely identical.

4.3.2 α -WGAN

The α -WGAN architecture, also based on [59], features specific modifications to the G and D structures.

4.3.2.1 Structure

The α -WGAN model retains similarities to WGAN but includes notable enhancements. The architectures for $64 \times 64 \times 64$ and $128 \times 128 \times 128$ images are detailed below.

The G starts with a fully connected layer that transforms a 100-dimensional noise vector into a feature map through a series of 3D transposed convolutions and batch normalisation. Feature maps are upsampled after each batch normalisation and ReLU activation layer. The final layer uses a tanh activation function to ensure output voxel values are within the range $[-1, 1]$.

The D processes the $64 \times 64 \times 64$ voxel image through a series of 3D convolutional layers, incrementing channels until the final layer, which produces the classification score. Lastly, the E encodes real images into latent space, and the Code Discriminator (CD) evaluates the latent space representation to make sure that the G produces realistic and coherent 3D images. It has three linear layers, batch normalisation, and leaky ReLU activations.

The $128 \times 128 \times 128$ model follows the same structure with additional layers and up-sampling for higher resolution.

4.3.2.2 Training

To facilitate comparison, the α -WGAN was trained with parameters similar to those used for WGAN.

In the training cycle, firstly the G and E parameters are updated while those of the D and CD remain unchanged temporarily. Real images are encoded into latent space, and fake images are generated by sampling random noise. The combined loss for E-G training includes the CD loss, D loss, and L1 loss.

Then the D's parameters are updated using real images from the training dataset and fake images. The loss included real and fake image losses and gradient penalties to enforce the Lipschitz constraint.

Lastly, the CD is trained using the CD loss and a gradient penalty.

To track progress, model checkpoints were saved every 10,000 iterations.

4.3.2.3 Summary

As with the WGAN model, two different experiments were made. However, the training of the model was the same, where the only thing different was the checkpoint of the model used for the slice generation. In experiment 1 the checkpoint of the model is an earlier iteration of the full training.

For the qualitative analysis of the generated images, a sequence of slices from random 3D images was also selected (see Figure 4.6). The α -WGAN generated images with clearer outlines and more detail than WGAN, particularly in the anatomical structures.

For the $64 \times 64 \times 64$ images, between the two experiments, the results were very similar with just some differences in the saturation of the lungs, wherein in the first experiment there is a stronger black colour characteristic of lung CT exams. This colour is because the lungs contain a large amount of air, which results in the lung fields appearing black on the scan. There is also a little bit more detail regarding the respiratory airways and vessels in the second experiment.

The $128 \times 128 \times 128$ images, as before, have more quality and detail than the $64 \times 64 \times 64$ ones. The outlines of the lungs as well as the outlines and details of the respiratory airways and vessels are more sharp and visible. It is also easier to see the rib structure. However, there are also some noticeable unwanted artifacts present in the images as well as a slight uncharacteristic grey colour that is visible in the slices. This can be due to the overfitting of the training data by the model.

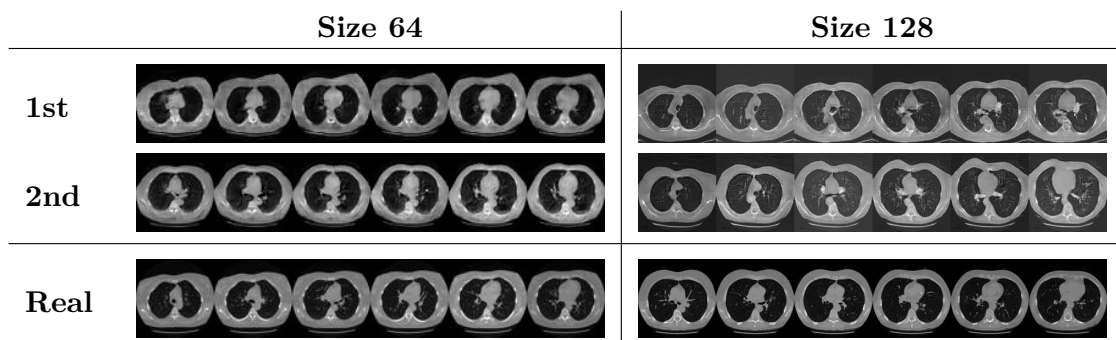


Figure 4.6: Sequence of generated slices using the α -WGAN architecture, for both the experiments and for both sizes. On the bottom, there is a sequence of real slices for comparison.

Similarly to WGAN, Figure 4.7 shows some generated, unrealistic samples.

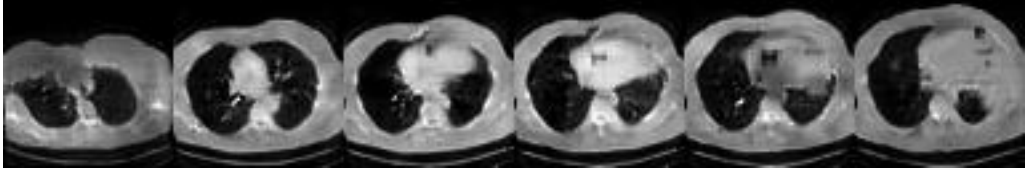


Figure 4.7: Sequence of unrealistic slices generated by the α -WGAN model.

For the quantitative analysis (see Table 4.3), identically, the same three different metrics were also used in α -WGAN. For the $64 \times 64 \times 64$ images, both experiments present very similar results with a slight increase of MS-SSIM and MMD and a slight decrease in FID in the second experiment. When compared to WGAN, the only two major differences were in the FID value which is lower in both experiments, which confirms the improvement in the quality of the images, and in the values of MS-SSIM, which are much lower than in the previous model. With this, we can confirm that the mode collapse problem of the previous model was fixed. For the $128 \times 128 \times 128$ images, both experiments also represent very similar results. However, when we compare with the values from WGAN, we can see that the values of MMD and FID are much higher and this is most likely caused by the artifacts and the unwanted grey colour mentioned in the qualitative analysis. Artifacts and color overlays represent deviations from the expected distribution which causes an increase in MMD. Since FID compares the statistics of features extracted from an Inception network for real and generated images, artifacts and colour issues will lead to significant differences in these statistics, therefore increasing FID values.

Table 4.3: Average quantitative results for α -WGAN.

Experiment	MMD $\times 10^{-4}$ ↓	3D-MS-SSIM ↑	FID ↓
64 - 1st	3.62	0.437	152.4
64 - 2nd	3.72	0.469	149.2
128 - 1st	87.65	0.339	294.9
128 - 2nd	87.00	0.305	296.8

4.3.3 HA-GAN

The HA-GAN architecture was based on the implementation [90], and uses hierarchical attention mechanisms to focus on different levels of image features, enhancing the generation of high-quality, realistic images.

4.3.3.1 Structure

For the HA-GAN, apart from the two previous models, only 1 model was used for images of $128 \times 128 \times 128$. This is because HA-GAN is primarily intended to manage the development of high-resolution images, which necessitates substantial memory resources, particularly

for 3D medical images. The architecture uses a hierarchical framework to create high-resolution sub-volumes and low-resolution images during training, which distributes the memory load and maintains both the global structure and local details. For high-resolution outputs, such as $128 \times 128 \times 128$, where memory efficiency is critical, this architecture is especially beneficial.

The architecture consists of five main components. The G, D, and E are responsible for extracting high-level features from the input 3D CT images, a Sub-Encoder (SE), and a CD.

Firstly, the G takes a latent vector as input and upsamples it through a series of transpose convolutional layers. It uses group normalisation and ReLU activation functions to ensure stable training and effective gradient propagation. It also has a SG that specialises in refining smaller regions (crops) of the generated images, providing higher detail where necessary. Then the D consists of multiple 3D convolutional layers with spectral normalisation and leaky ReLU activations to enforce non-linearity.

Similarly to the G, it features two Sub-Discriminators where one operates on high-resolution patches, and the other on low ones. The final output is a classification score indicating the likelihood that the input is real or generated. The E takes as input single-channel 3D CT images and has a 3D convolutional layer, along with a GroupNorm for normalisation, which normalises feature channels in groups, enhancing training stability and performance, a ReLU activation for non-linearity, and multiple convolutional layers.

To complement it, the SE is composed of additional layers that further reduce the spatial dimensions and increase the depth of the feature maps, ending in a latent space representation suitable for image generation.

Lastly, the CD evaluates the latent space representation to make sure that the G produces realistic and coherent 3D images. It has three linear layers, batch normalisation, and leaky ReLU activations.

4.3.3.2 Training

The model was trained with the following parameters:

- **Latent Dimension:** The size of the input latent variable was set to 1024;
- **Batch Size:** A batch size of 4 was chosen to balance training stability and memory constraints;
- **Learning Rate:** The G, D, E, and SE were optimised using the Adam optimiser with a learning rate of 0.0001, 0.0004, 0.0001, 0.0001, respectively;
- **Training Iterations:** The model was trained for a total of 80,000 iterations.

In the training cycle, the E training initiates with encoding real images into latent space, followed by generating reconstructions from the latent representations. The E loss

is computed using L1 loss, reflecting the fidelity of the reconstructions to the original images. The E parameters are updated based on this computed loss to improve the encoding process.

Then during D training, real images are first loaded and processed to extract high-resolution sub-volumes. Random noise is then utilised to generate fake images. The training process involves computing the loss for both real and fake images, incorporating binary cross-entropy (BCE) loss. Subsequently, the D parameters are updated through backpropagation based on the computed loss.

The G training begins by freezing the parameters of the D to disable gradient updates. This allows the G to generate fake images using random noise. The training proceeds with computing and backpropagating the G loss, which is then used to update the G parameters.

Lastly, the SE training involves freezing both E and G parameters. Latent representations are concatenated to facilitate the generation of reconstructed images by the SE. The loss for the SE is then computed using L1 loss, assessing the similarity between the generated high-resolution and low-resolution images.

Losses and visualisations are logged every 20 iterations and model checkpoints are saved every 10,000 iterations to track progress.

4.3.3.3 Summary

As with the α -WGAN model, two experiments were conducted using two different checkpoints of the same training (see Figure 4.8). The first experiment, which corresponds to an earlier checkpoint of the model’s training, ends up showing a little bit less detail than the second experiment, especially in the internal lung structure detail. The second experiment shows, visually, the best results out of the two previous models, where all the anatomic structures are correctly outlined as well as the rib structure and the respiratory airways and vessels.

Figure 4.9 shows examples of some generated unrealistic samples.

Table 4.4 shows the same metrics as the ones used by the two previous models. Between the two experiments, the results were very similar, with a small decrease in MMD and a little increase in FID. When compared to the other models, there is an improvement regarding the FID values. With the MMD values, it showed results higher than WGAN and lower than α -WGAN.

Table 4.4: Average quantitative results for HA-GAN.

Experiment	MMD $\times 10^{-4}$ ↓	3D-MS-SSIM ↑	FID ↓
128 - 1st	30.86	0.337	156.06
128 - 2nd	28.00	0.346	157.03

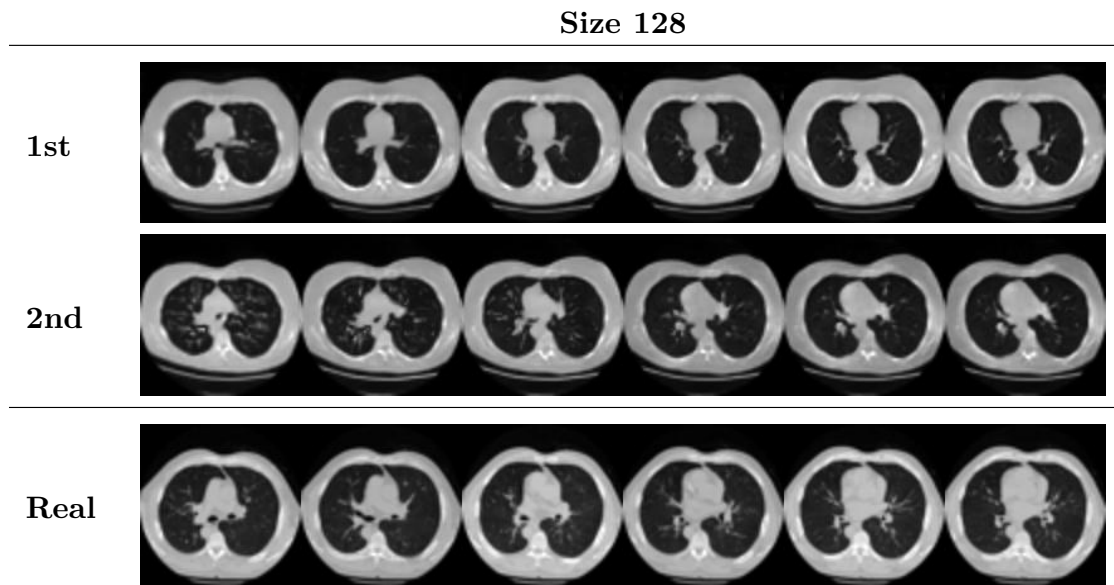


Figure 4.8: Sequence of generated slices using the HA-WGAN architecture, for both the experiments and for size 128. On the bottom, there is a sequence of real slices for comparison.

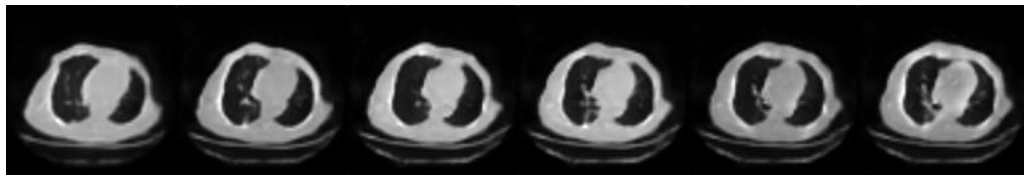


Figure 4.9: Sequence of unrealistic slices generated by the HA-GAN model.

Chapter 5

Results

Contents

5.1 Overview	73
5.2 Visual Turing Test	74

5.1 Overview

In this chapter, the best results of all models are evaluated by comparing the MMD, MS-SSIM, and FID values.

For $64 \times 64 \times 64$ images, the 2 first models will be compared and for $128 \times 128 \times 128$ images all 3 models will be compared.

When it comes to the MMD metric, the WGAN model ended up achieving the lowest score of 3.48 a little bit lower than the α -WGAN model, which means that the α -WGAN model has the largest discrepancy between the real and generated image distributions (see Table 5.1). For the MS-SSIM, the MS-SSIM of the real dataset (0.3254) will be used as a method of comparison. As mentioned previously, the WGAN model suffered from mode collapse, therefore presenting values of structure similarity between the generated images very close to 1, achieving very poor variety in the generated dataset. Ignoring the mode collapse, the α -WGAN had the lowest score of MS-SSIM which indicates that for $64 \times 64 \times 64$ images, it achieved the best result. For the FID values, as predicted, the α -WGAN achieves the lowest results, meaning it produced the most realistic images (see Table 5.1).

Table 5.1: Metric comparison for $64 \times 64 \times 64$ images.

Model	MMD $\times 10^{-4}$ ↓	3D-MS-SSIM ↑	FID ↓
WGAN	3.48	0.997	189.4
α -WGAN	3.62	0.437	149.2

On the $128 \times 128 \times 128$ images, the WGAN achieved the best results for the MMD metric, followed closely by the HA-GAN and then by the α -WGAN with the highest value. For the MS-SSIM, the α -WGAN presents the best results, followed closely by the HA-GAN, where both of these models present similar results to the ones from the real dataset. However, this can be justified by the fact that in both models there were unrealistic generated examples. Finally, for the FID scores, the HA-GAN shows the lowest score out of the 3 models, followed by the WGAN, and lastly by the α -WGAN (see Table 5.2).

Table 5.2: Metric comparison for $128 \times 128 \times 128$ images.

Model	MMD $\times 10^{-4}$ ↓	3D-MS-SSIM ↑	FID ↓
WGAN	16.84	0.431	178
α -WGAN	87.00	0.305	294.9
HA-GAN	28	0.337	156.06

In conclusion, for $64 \times 64 \times 64$ images, α -WGAN, demonstrated the best performance out of the 2 models, and for $128 \times 128 \times 128$ images, the HA-GAN proved to have the best overall results out of the 3 models, generating, visually, the most realistic and structurally similar images to the real ones.

5.2 Visual Turing Test

The VTT test was done by asking a radiologist to distinguish 45 different lung CT scans and classify them as real or generated. The set of images used for the VTT was obtained by firstly taking 25,400 real images and 25,400 generated images from the HA-GAN model and putting them in a pool. This prevented the introduction of bias from choosing the images by hand, and most likely selecting the best ones. From this pool, and to make sure the number of real images was not the same as the number of generated images, there was a random selection of between 40% and 60% of the total images. This was done to ensure that there would be no bias from the radiologist when evaluating the images.

Then, all the selected images were shuffled to ensure randomness and presented to the radiologist. For the calculations, the generated scans were regarded as positive and the real scans as negative in the confusion matrix calculations (see Table 5.3). In addition, we also report the results of five metrics that were computed using the matrices: accuracy, precision, recall, False Omission Rate (FOR), and Negative Predicted Value (NPV).

For a VTT, the optimal outcomes would be 0.5 for accuracy, 0 for precision, 0 for recall, 0.5 for FOR, and 0.5 for NPV, (i.e., exclusively negative predictions).

Based on the results in Table 5.3, we observe that the radiologist demonstrated an accuracy of 0.67, a precision of 0.76, a recall of 0.62, a FOR of 0.42, and an NPV of 0.58. The accuracy tells us that, overall, the radiologist managed to successfully predict 67%

Table 5.3: Confusion Matrix and Performance Metrics for the Radiologist

Radiologist	Confusion Matrix				Performance Metrics				
	TP	FP	TN	FN	Accuracy	Precision	Recall	FOR	NPV
R1	16	5	14	10	0.67	0.76	0.62	0.42	0.58

of the images and the precision indicates that 76% of the images identified as generated were generated. Regarding the recall values, the radiologist correctly identifies 62% of the generated scans.

As for the FOR value, it means that 42% of the scans identified as real are generated. Lastly, the NPV values indicate that 58% of the scans identified as real are indeed real.

During the VTT, the radiologist stated that the resolution of the images under evaluation was very limiting. This constraint could substantially affect the accuracy of visual assessments, as high-resolution images are essential for detailed medical analysis and are integral to the work of medical professionals in that area.

Therefore the radiologist noted that he relied on intuition and years of experience to perform the examination. This subjective approach may have introduced variability in the assessment which can be confirmed by the unclear metrics.

Chapter 6

Conclusion and Future Work

Contents

6.1 Overview	76
6.2 Future Work	77

6.1 Overview

The goal of this work was to utilise generative models that could simulate 3D lung CT scans in a way that would be realistic and spatially coherent, making it impossible for medical professionals to tell them apart from actual scans. To do this, research was done on the most prevalent pathology's, the structure of the lungs, and the technology underlying CT imaging. Following that, the literature on generative models was examined, with a particular emphasis on GANs, which are not only the most studied in conventional 2D applications but also rule the biomedical 3D realm.

The choice of 3 models was made according to 2 main factors. Firstly, for the WGAN was its improved training stability, secondly for the α -WGAN was the additional flexibility, which presented as an upgrade to the original WGAN model at the cost of added complexity in training, and for the HA-GAN was its excellence in generating detailed high-resolution images by using an hierarchical approach. The second, and most important factor, is that these models had already been successfully used in the generation of 3D medical images. This work adapts the architecture of those models to the generation of 3D lung CT exams using the publicly available LIDC/IDRI dataset for the training of the models.

Despite having some unrealistic scans in between, the three models were able to synthesise spatially coherent and realistic scans. These scans ended up achieving similar variability to the training dataset, apart from the WGAN model for $64 \times 64 \times 64$ images which suffered

from mode collapse. Out of the 3 models, the one that resulted in the highest quality generated images was the HA-GAN which also had the best overall results in terms of metrics.

In addition, a radiology doctor from Hospital São João was asked to administer a Visual Turing Test (VTT), where he was asked to identify 45 scans as artificial or real. The inconclusive results were primarily caused by the restricted resolution (128) which turned out to be an obstacle for the radiologist.

Transforming into a 3D space allows radiologists working in a medical setting to analyse data, which opens up a world of possibilities for the field of CT lung scan creation. However, the primary flaw in this approach is that this shift comes at a high-resolution cost.

6.2 Future Work

Apart from the limitations regarding resolution, there are also some other recommendations to improve this work:

- Reduce the degree of similarity in the generated samples by using a larger dataset with greater data diversity, or even do cross-training with multiple datasets;
- Incorporate additional quantitative metrics to provide a more comprehensive evaluation of image quality;
- Involve more radiologists in the VTT to reduce subjective biases and improve the robustness of the evaluation;
- Implement advanced regularisation techniques to reduce artifacts and unwanted colours and try other training techniques.

References

- [1] Adni. <https://adni.loni.usc.edu/>.
- [2] Basic principles in computed tomography (CT).
- [3] CT scans - Physiopedia.
- [4] Introduction to PyTorch tensors —PyTorch tutorials 2.3.0+cu121 documentation.
- [5] Learned perceptual image patch similarity (LPIPS) —PyTorch-metrics 1.2.1 documentation.
- [6] Lung anatomy - Physiopedia.
- [7] Overview about cancer - World Health Organisation.
- [8] Peak signal-to-noise ratio as an image quality metric - NI.
- [9] Structseg-thora. <https://structseg2019.grand-challenge.org/dataset/>.
- [10] What is cancer? - National Cancer Institute.
- [11] What is lung cancer? Types of lung cancer - American Cancer Society.
- [12] Why is lung cancer so deadly? - NCFR.
- [13] Worldwide cancer data - World Cancer Research Fund International.
- [14] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv*, 2017.
- [15] Samuel G Armato, Geoffrey McLennan, Luc Bidaut, Michael F McNitt-Gray, Charles R Meyer, Anthony P Reeves, Binsheng Zhao, Denise R Aberle, Claudia I Henschke, Eric A Hoffman, Ella A Kazerooni, Heber MacMahon, Edwin J R Van Beeke, David Yankelevitz, Alberto M Biancardi, Peyton H Bland, Matthew S Brown, Roger M Engelmann, Gary E Laderach, Daniel Max, Richard C Pais, David P Y Qing, Rachael Y Roberts, Amanda R Smith, Adam Starkey, Poonam Batrah, Philip Caligiuri, Ali Farooqi, Gregory W Gladish, C Matilda Jude, Reginald F Munden, Iva Petkovska, Leslie E Quint, Lawrence H Schwartz, Baskaran Sundaram, Lori E Dodd, Charles Fenimore, David Gur, Nicholas Petrick, John Freymann, Justin Kirby, Brian Hughes, Alessi Vande Casteele, Sangeeta Gupte, Maha Sallamm, Michael D Heath, Michael H Kuhn, Ekta Dharaia, Richard Burns, David S Fryd, Marcos Salganicoff, Vikram Anand, Uri Shreter, Stephen Vastagh, and Barbara Y Croft. The lung image database consortium (LIDC) and image database resource initiative (IDRI): a completed reference database of lung nodules on CT scans. *Medical Physics*, 38(2):915–931, feb 2011.

- [16] Samuel G. Armato III, Geoffrey McLennan, Luc Bidaut, Michael F. McNitt-Gray, Charles R. Meyer, Anthony P. Reeves, Binsheng Zhao, Denise R. Aberle, Claudia I. Henschke, Eric A. Hoffman, Ella A. Kazerooni, Heber MacMahon, Edwin J.R. Van Beek, David Yankelevitz, Alberto M. Biancardi, Peyton H. Bland, Matthew S. Brown, Roger M. Engelmann, Gary E. Laderach, Daniel Max, Richard C. Pais, David P.Y. Qing, Rachael Y. Roberts, Amanda R. Smith, Adam Starkey, Poonam Batra, Philip Caligiuri, Ali Farooqi, Gregory W. Gladish, C. Matilda Jude, Reginald F. Munden, Iva Petkovska, Leslie E. Quint, Lawrence H. Schwartz, Baskaran Sundaram, Lori E. Dodd, Charles Fenimore, David Gur, Nicholas Petrick, John Freymann, Justin Kirby, Brian Hughes, Alessi Vande Castele, Sangeeta Gupte, Maha Sallam, Michael D. Heath, Michael H. Kuhn, Ekta Dharaiya, Richard Burns, David S. Fryd, Marcos Salganicoff, Vikram Anand, Uri Shreter, Stephen Vastagh, Barbara Y. Croft, and Laurence P. Clarke. Data from LIDC-IDRI. *The Cancer Imaging Archive*, 2015.
- [17] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. ViViT: A video vision transformer. *arXiv*, 2021.
- [18] Spyridon Bakas, Hamed Akbari, Aristeidis Sotiras, Michel Bilello, Martin Rozycki, Justin S Kirby, John B Freymann, Keyvan Farahani, and Christos Davatzikos. Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features. *Scientific data*, 4:170117, sep 2017.
- [19] Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. In Isabelle Guyon, Gideon Dror, Vincent Lemaire, Graham Taylor, and Daniel Silver, editors, *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, volume 27 of *Proceedings of Machine Learning Research*, pages 37–49, Bellevue, Washington, USA, 02 Jul 2012. PMLR.
- [20] Dor Bank, Noam Koenigstein, and Raja Giryes. Autoencoders. *arXiv*, 2020.
- [21] Olivier Bernard, Alain Lalande, Clement Zotti, Frederick Cervenansky, Xin Yang, Pheng-Ann Heng, Irem Cetin, Karim Lekadir, Oscar Camara, Miguel Angel Gonzalez Ballester, Gerard Sanroma, Sandy Napel, Steffen Petersen, Georgios Tziritas, Elias Grinias, Mahendra Khened, Varghese Alex Kollerathu, Ganapathy Krishnamurthi, Marc-Michel Rohé, Xavier Pennec, Maxime Sermesant, Fabian Isensee, Paul Jäger, Klaus H. Maier-Hein, Peter M. Full, Ivo Wolf, Sandy Engelhardt, Christian F. Baumgartner, Lisa M. Koch, Jelmer M. Wolterink, Ivana Išgum, Yeonggul Jang, Yoonmi Hong, Jay Patravali, Shubham Jain, Olivier Humbert, and Pierre-Marc Jodoin. Deep learning techniques for automatic mri cardiac multi-structures segmentation and diagnosis: Is the problem solved? *IEEE Transactions on Medical Imaging*, 37(11):2514–2525, 2018.
- [22] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.
- [23] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *arXiv*, 2015.
- [24] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. *arXiv*, 2018.

- [25] Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. Mode regularized generative adversarial networks. *arXiv*, 2016.
- [26] Xi Chen, Nikhil Mishra, Mostafa Rohaninejad, and Pieter Abbeel. PixelSNAIL: An improved autoregressive generative model. *arXiv*, 2017.
- [27] Xuming Chen, Shanlin Sun, Narisu Bai, Kun Han, Qianqian Liu, Shengyu Yao, Hao Tang, Chupeng Zhang, Zhipeng Lu, Qian Huang, Guoqi Zhao, Yi Xu, Tingfeng Chen, Xiaohui Xie, and Yong Liu. A deep learning-based auto-segmentation system for organs-at-risk on whole-body computed tomography images for radiation therapy. *Radiotherapy and Oncology*, 160:175–184, jul 2021.
- [28] Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: Non-linear independent components estimation. *arXiv*, 2014.
- [29] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. *arXiv*, 2016.
- [30] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *arXiv*, 2016.
- [31] Sam Ellis, Octavio E. Martinez Manzanera, Vasileios Baltatzis, Ibrahim Nawaz, Arjun Nair, Loïc Le Folgoc, Sujal Desai, Ben Glocker, and Julia A. Schnabel. Evaluation of 3D GANs for lung tissue modelling in pulmonary CT. *arXiv*, 2022.
- [32] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. *arXiv*, 2020.
- [33] Xinyang Feng, Zachary C. Lipton, Jie Yang, Scott A. Small, and Frank A. Provenzano. Estimating brain age based on a healthy population with deep learning and structural MRI. *arXiv*, 2019.
- [34] Donald Geman, Stuart Geman, Neil Hallonquist, and Laurent Younes. Visual turing test for computer vision systems. *Proceedings of the National Academy of Sciences of the United States of America*, 112(12):3618–3623, mar 2015.
- [35] Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. Exploring the structure of a real-time, arbitrary neural artistic stylization network. *arXiv*, 2017.
- [36] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [37] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv*, 2014.
- [38] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *J. Mach. Learn. Res.*, 13(null):723–773, mar 2012.
- [39] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein GANs. *arXiv*, 2017.

- [40] Kun Han, Yifeng Xiong, Chenyu You, Pooya Khosravi, Shanlin Sun, Xiangyi Yan, James Duncan, and Xiaohui Xie. MedGen3D: A deep generative framework for paired 3D image and mask generation. *arXiv*, 2023.
- [41] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and D. Salesin. Image analogies. *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001.
- [42] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. *arXiv*, 2017.
- [43] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv*, 2020.
- [44] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv*, 2022.
- [45] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. *arXiv*, 2022.
- [46] Johannes Hofmanninger, Florian Prayer, Jeanny Pan, Sebastian Rohrich, Helmut Prosch, and Georg Langs. Automatic lung segmentation in routine imaging is primarily a data diversity problem, not a methodology problem. *arXiv*, 2020.
- [47] Avram J Holmes, Marisa O Hollinshead, Timothy M O’Keefe, Victor I Petrov, Gabriele R Fariello, Lawrence L Wald, Bruce Fischl, Bruce R Rosen, Ross W Mair, Joshua L Roffman, Jordan W Smoller, and Randy L Buckner. Brain genomics superstruct project initial data release with structural, functional, and behavioral measures. *Scientific data*, 2:150031, jul 2015.
- [48] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. *arXiv*, 2017.
- [49] Sarfaraz Hussein, Kunlin Cao, Qi Song, and Ulas Bagci. Risk stratification of lung nodules using 3D CNN-based multi-task learning. *arXiv*, 2017.
- [50] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv*, 2015.
- [51] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv*, 2016.
- [52] Clifford R Jack, Matt A Bernstein, Nick C Fox, Paul Thompson, Gene Alexander, Danielle Harvey, Bret Borowski, Paula J Britson, Jennifer L Whitwell, Chadwick Ward, Anders M Dale, Joel P Felmlee, Jeffrey L Gunter, Derek L G Hill, Ron Killiany, Norbert Schuff, Sabrina Fox-Bosetti, Chen Lin, Colin Studholme, Charles S DeCarli, Gunnar Krueger, Heidi A Ward, Gregory J Metzger, Katherine T Scott, Richard Mallozzi, Daniel Blezek, Joshua Levy, Josef P Debbins, Adam S Fleisher, Marilyn Albert, Robert Green, George Bartzokis, Gary Glover, John Mugler, and Michael W Weiner. The alzheimer’s disease neuroimaging initiative (ADNI): MRI methods. *Journal of Magnetic Resonance Imaging*, 27(4):685–691, apr 2008.
- [53] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. *arXiv*, 2017.

- [54] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *arXiv*, 2018.
- [55] Firas Khader, Gustav Müller-Franzes, Soroosh Tayebi Arasteh, Tianyu Han, Christoph Haarbuerger, Maximilian Schulze-Hagen, Philipp Schad, Sandy Engelhardt, Bettina Baeßler, Sebastian Foersch, Johannes Stegmaier, Christiane Kuhl, Sven Nebelung, Jakob Nikolas Kather, and Daniel Truhn. Denoising diffusion probabilistic models for 3D medical image generation. *Scientific Reports*, 13(1):7303, may 2023.
- [56] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *arXiv*, 2018.
- [57] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv*, 2013.
- [58] Ivan Kobyzev, Simon J. D. Prince, and Marcus A. Brubaker. Normalizing flows: An introduction and review of current methods. *arXiv*, 2019.
- [59] Gihyun Kwon, Chihye Han, and Dae-shik Kim. Generation of 3D brain MRI using auto-encoding generative adversarial networks. *arXiv*, 2019.
- [60] Z. Lambert, C. Petitjean, B. Dubray, and S. Ruan. SegTHOR: Segmentation of thoracic organs at risk in CT images. *arXiv*, 2019.
- [61] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv*, 2015.
- [62] Joseph J. Lim, Hamed Pirsiavash, and Antonio Torralba. Parsing ikea objects: Fine pose estimation. In *2013 IEEE International Conference on Computer Vision*, pages 2992–2999, 2013.
- [63] Andrew L. Maas. Rectifier nonlinearities improve neural network acoustic models. 2013.
- [64] Daniel S Marcus, Tracy H Wang, Jamie Parker, John G Csernansky, John C Morris, and Randy L Buckner. Open access series of imaging studies (OASIS): cross-sectional MRI data in young, middle aged, nondemented, and demented older adults. *Journal of Cognitive Neuroscience*, 19(9):1498–1507, sep 2007.
- [65] Bjoern H. Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, Levente Lenczi, Elizabeth Gerstner, Marc-André Weber, Tal Arbel, Brian B. Avants, Nicholas Ayache, Patricia Buendia, D. Louis Collins, Nicolas Cordier, Jason J. Corso, Antonio Criminisi, Tilak Das, Hervé Delingette, Çağatay Demiralp, Christopher R. Durst, Michel Dojat, Senan Doyle, Joana Festa, Florence Forbes, Ezequiel Geremia, Ben Glocker, Polina Golland, Xiaotao Guo, Andac Hamamci, Khan M. Iftikharuddin, Raj Jena, Nigel M. John, Ender Konukoglu, Danial Lashkari, José António Mariz, Raphael Meier, Sérgio Pereira, Doina Precup, Stephen J. Price, Tammy Riklin Raviv, Syed M. S. Reza, Michael Ryan, Duygu Sarikaya, Lawrence Schwartz, Hoo-Chang Shin, Jamie Shotton, Carlos A. Silva, Nuno Sousa, Nagesh K. Subbanna, Gabor Szekely, Thomas J. Taylor,

- Owen M. Thomas, Nicholas J. Tustison, Gozde Unal, Flor Vasseur, Max Wintermark, Dong Hye Ye, Liang Zhao, Binsheng Zhao, Darko Zikic, Marcel Prastawa, Mauricio Reyes, and Koen Van Leemput. The multimodal brain tumor image segmentation benchmark (brats). *IEEE Transactions on Medical Imaging*, 34(10):1993–2024, 2015.
- [66] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *ArXiv*, abs/1411.1784, 2014.
- [67] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv*, 2018.
- [68] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning*, 2010.
- [69] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Ng. Reading digits in natural images with unsupervised feature learning. *NIPS*, 01 2011.
- [70] Sergey I. Nikolenko. Generative models in deep learning. In *Synthetic data for deep learning*, volume 174 of *Springer optimization and its applications*, pages 97–137. Springer International Publishing, Cham, 2021.
- [71] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier GANs. *arXiv*, 2016.
- [72] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv*, 2016.
- [73] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with PixelCNN decoders. *arXiv*, 2016.
- [74] Wei Peng, Ehsan Adeli, Qingyu Zhao, and Kilian M. Pohl. Generating realistic 3D brain MRIs using a conditional diffusion probabilistic model. *arXiv*, 2022.
- [75] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. *arXiv*, 2017.
- [76] R C Petersen, P S Aisen, L A Beckett, M C Donohue, A C Gamst, D J Harvey, C R Jack, W J Jagust, L M Shaw, A W Toga, J Q Trojanowski, and M W Weiner. Alzheimer’s disease neuroimaging initiative (ADNI): clinical characterization. *Neurology*, 74(3):201–209, jan 2010.
- [77] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv*, 2015.
- [78] Claudia Ravaglia, Claudio Doglioni, Marco Chilosi, Sara Piciocchi, Alessandra Dubini, Giulio Rossi, Federica Pedica, Silvia Puglisi, Luca Donati, Sara Tomassetti, and Venerino Poletti. Clinical, radiological and pathological findings in patients with persistent lung disease following SARS-CoV-2 infection. *The European Respiratory Journal*, 60(4), oct 2022.

- [79] Elizabeth A Regan, John E Hokanson, James R Murphy, Barry Make, David A Lynch, Terri H Beaty, Douglas Curran-Everett, Edwin K Silverman, and James D Crapo. Genetic epidemiology of COPD (COPDGene) study design. *Copd*, 7(1):32–43, feb 2010.
- [80] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *arXiv*, 2015.
- [81] Mihaela Rosca, Balaji Lakshminarayanan, David Warde-Farley, and Shakir Mohamed. Variational approaches for auto-encoding generative adversarial networks. *arXiv*, 2017.
- [82] David E. Rumelhart and James L. McClelland. *Learning Internal Representations by Error Propagation*, pages 318–362. 1987.
- [83] Ashirbani Saha, Michael R. Harowicz, Lars J. Grimm, Jingxi Weng, E. H. Cain, C. E. Kim, S. V. Ghate, R. Walsh, and Maciej A. Mazurowski. Dynamic contrast-enhanced magnetic resonance images of breast cancer patients with tumor locations. *The Cancer Imaging Archive*, 2021.
- [84] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. *arXiv*, 2016.
- [85] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. PixelCNN++: Improving the PixelCNN with discretized logistic mixture likelihood and other modifications. *arXiv*, 2017.
- [86] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *arXiv*, 2015.
- [87] Joana Sousa, Tania Pereira, Inês Neves, Francisco Silva, and Helder P Oliveira. The influence of a coherent annotation and synthetic addition of lung nodules for lung segmentation in CT scans. *Sensors (Basel, Switzerland)*, 22(9), apr 2022.
- [88] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv*, 2014.
- [89] Karl D. Stephen and Alireza Kazemi. Improved normalization of time-lapse seismic data using normalized root mean square repeatability data to improve automatic production and seismic history matching in the nelson field. *Geophysical Prospecting*, 62(5):1009–1027, sep 2014.
- [90] Li Sun, Junxiang Chen, Yanwu Xu, Mingming Gong, Ke Yu, and Kayhan Batmanghelich. Hierarchical amortized GAN for 3D high resolution medical image synthesis. *IEEE journal of biomedical and health informatics*, PP, may 2022.
- [91] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *arXiv*, 2015.
- [92] Fabio Henrique Kiyoyiti dos Santos Tanaka and Claus Aranha. Data augmentation using GANs. *arXiv*, 2019.

- [93] Yucheng Tang, Dong Yang, Wenqi Li, Holger Roth, Bennett Landman, Daguang Xu, Vishwesh Nath, and Ali Hatamizadeh. Self-supervised pre-training of swin transformers for 3D medical image analysis. *arXiv*, 2021.
- [94] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv*, 2017.
- [95] Cédric Villani. Convergence of metric-measure spaces. In M. Berger, B. Eckmann, P. de la Harpe, F. Hirzebruch, N. Hitchin, L. Hörmander, A. Kupiainen, G. Lebeau, M. Ratner, D. Serre, Ya. G. Sinai, N. J. A. Sloane, A. M. Vershik, and M. Waldschmidt, editors, *Optimal Transport*, volume 338 of *Grundlehren der mathematischen Wissenschaften*, pages 743–772. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [96] Andong Wang, Terence Chi Chun Tam, Ho Ming Poon, Kun-Chang Yu, and Wei-Ning Lee. NaviAirway: a bronchiole-sensitive deep learning-based airway segmentation pipeline. *arXiv*, 2022.
- [97] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. *arXiv*, 2018.
- [98] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402 Vol.2, 2003.
- [99] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.
- [100] Jakob Wasserthal, Manfred Meyer, Hanns-Christian Breit, Joshy Cyriac, Shan Yang, and Martin Segeroth. TotalSegmentator: robust segmentation of 104 anatomical structures in CT images. *arXiv*, 2022.
- [101] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T. Freeman, and Joshua B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. *arXiv*, 2016.
- [102] Shibo Xing and Seong Jae Hwang. Cycle consistent embedding of 3d brains with auto-encoding generative adversarial networks. 2021.
- [103] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv*, 2015.
- [104] Yanwu Xu, Li Sun, Wei Peng, Shyam Visweswaran, and K. Batmanghelich. Medsyn: Text-guided anatomy-aware synthesis of high-fidelity 3d ct images. *ArXiv*, abs/2310.03559, 2023.
- [105] Jee Seok Yoon, Chenghao Zhang, Heung-Il Suk, Jia Guo, and Xiaoxiao Li. SADM: Sequence-aware diffusion model for longitudinal medical image generation. *arXiv*, 2022.

- [106] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv*, 2015.
- [107] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv*, 2018.
- [108] Jiequan Zhang, Qingyu Zhao, Ehsan Adeli, Adolf Pfefferbaum, Edith V Sullivan, Robert Paul, Victor Valcour, and Kilian M Pohl. Multi-label, multi-domain learning identifies compounding effects of HIV and cognitive impairment. *Medical Image Analysis*, 75:102246, jan 2022.
- [109] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv*, 2023.
- [110] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv*, 2017.