h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbmn
FACHBEREICH MATHEMATIK
UND NATURWISSENSCHAFTEN

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Hochschule Darmstadt

Fachbereiche Mathematik und
Naturwissenschaften & Informatik

Synthetic MRI Image Generation Using a Combination of VQ-GAN and Flow Matching

Thesis for the Award of the Academic Degree

Master of Science (M. Sc.)
in the Study Program Data Science

submitted by:

Nora Schaba

First supervisor	:	Prof. Dr. Elke Hergenröther
Second supervisor	:	Prof. Dr. Andreas Weinmann
Institut supervisor	:	Dr.-Ing. Cristina Oyarzun Laura

Issue date	:	10 Juni 2024
Submission date	:	09. December 2024

DECLARATION

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht.

Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Darmstadt, 09. December 2024

Nora Schaba

ABSTRACT

The application of deep learning in medical imaging has shown great potential but remains constrained by the scarcity and imbalance of available datasets. Challenges such as data privacy concerns, variability in medical pathologies, and limited access to comprehensive datasets limit the development of robust and generalizable models. Synthetic data generation offers a promising solution to these limitations by enabling the creation of diverse and realistic medical datasets that address data scarcity and imbalance.

This thesis investigates the potential of Flow Matching (FM) models for generating three-dimensional (3D) synthetic medical images, with a focus on brain MRI scans. Although FM models have demonstrated great performance and efficiency in non-medical contexts, their application in medical imaging, particularly for 3D data, remains largely unexplored. To address this gap, a novel architecture is proposed that combines FM with a Variational Autoencoder-based Generative Adversarial Network (Vector Quantized Generative Adversarial Network (VQ-GAN)). The (VQ-GAN) encodes high-dimensional data into a compact latent space, where the FM model operates to generate synthetic data efficiently.

The proposed framework is evaluated using publicly available brain MRI datasets (The Alzheimer's Disease Neuroimaging Initiative (ADNI)) through a series of quantitative and qualitative experiments. The results highlight both the strengths and limitations of Flow Matching (FM) models in generating high-quality medical images, emphasizing the importance of preprocessing, dataset characteristics, and model design. In comparison to a denoising diffusion model, the FM model demonstrated significant advantages, achieving good quality results while reducing training time from 10 days to just 13 hours. However, challenges such as the loss of fine anatomical details due to computational constraints were observed. Despite these limitations, the framework demonstrates significant potential for advancing synthetic data generation in medical imaging. To the best of our knowledge, this work represents the first systematic exploration of FM models for generating 3D medical images.

This study offers a foundation for future research aimed at optimizing Flow Matching models further and explore their application to other medical imaging modalities, such as CT or PET scans. By reducing training times and enabling the generation of diverse datasets, the proposed framework could significantly impact real-world healthcare applications, including diagnostic tools, training datasets, and privacy-compliant synthetic data augmentation.

ZUSAMMENFASSUNG

Die Anwendung von Deep Learning in der medizinischen Bildgebung birgt ein enormes Potenzial, wird jedoch durch den Mangel an Daten und die Ungleichheit verfügbarer Datensätze erheblich eingeschränkt. Herausforderungen wie Datenschutzbedenken, die Variabilität medizinischer Pathologien und der begrenzte Zugang zu umfassenden Datensätzen erschweren die Entwicklung robuster und generalisierbarer Modelle. Die Generierung synthetischer Daten bietet eine vielversprechende Lösung für diese Herausforderungen, da sie die Erstellung diverser und realistischer medizinischer Datensätze ermöglicht, die sowohl den Datenmangel als auch die Ungleichheit in der Datensammlung berücksichtigt.

Die vorliegende Arbeit untersucht das Potenzial von Flow Matching (FM)-Modellen zur Generierung dreidimensionaler (3D) synthetischer medizinischer Bilder, mit einem besonderen Fokus auf MRT-Aufnahmen des Gehirns. Während FM-Modelle in nicht-medizinischen Anwendungsbereichen durch ihre überlegene Leistung und Effizienz überzeugen konnten, ist ihre Nutzung in der medizinischen Bildgebung, insbesondere im Zusammenhang mit 3D-Daten, bislang kaum erforscht. Zur Schließung dieser Forschungslücke wird in dieser Arbeit eine neuartige Architektur vorgestellt, die FM mit einem Variational Autoencoder-basierten Generative Adversarial Network (VQ-GAN) kombiniert. (VQ-GAN) dient dazu, hochdimensionale Daten in einen kompakten latenten Raum zu kodieren, in dem das FM-Modell effizient synthetische medizinische Daten generieren kann.

Das vorgeschlagene Framework wird anhand öffentlich verfügbarer MRT-Datensätze (ADNI) in einer Reihe von quantitativen und qualitativen Experimenten evaluiert. Die Ergebnisse zeigen sowohl die Stärken als auch die Schwächen von FM-Modellen bei der Generierung hochqualitativer medizinischer Bilder und heben die zentrale Bedeutung von Vorverarbeitung, Datensatzcharakteristika und Modelldesign hervor. Im Vergleich zu einem Denoising Diffusion Model erzielte das FM-Modell signifikante Vorteile: Es erreichte eine gute Bildqualität, während die Trainingszeit drastisch von 10 Tagen auf lediglich 13 Stunden reduziert werden konnte. Allerdings wurden Herausforderungen wie der Verlust feiner anatomischer Details, bedingt durch Einschränkungen in der Rechenkapazität, beobachtet. Trotz dieser Limitationen demonstriert das Framework ein erhebliches Potenzial für die Weiterentwicklung der Generierung synthetischer Daten in der medizinischen Bildgebung. Nach unserem Kenntnisstand handelt es sich hierbei um die erste systematische Untersuchung von FM-Modellen zur Generierung dreidimensionaler medizinischer Bilder.

Die gewonnenen Erkenntnisse bilden eine solide Grundlage für zukünftige Forschungen zur Weiterentwicklung und Optimierung von FM-Modellen sowie deren Anwendung auf andere medizinische Bildgebungsmodalitäten,

wie beispielsweise CT- oder PET-Scans. Insbesondere die Reduzierung der Trainingszeiten und die Möglichkeit, diversifizierte und strukturierte Datensätze zu erstellen, unterstreichen das Potenzial des vorgeschlagenen Frameworks. Dieses könnte erhebliche Auswirkungen auf praktische Anwendungen im Gesundheitswesen haben, einschließlich der Entwicklung diagnostischer Werkzeuge, der Erstellung von Trainingsdatensätzen sowie der datenschutzkonformen synthetischen Datenaugmentation.

CONTENTS

I Thesis

1 Introduction 2
- 1.1 Motivation . . . 2
- 1.2 Aim of This Thesis . . . 2
- 1.3 Structure . . . 4

2 Theoretical Background 5
- 2.1 Magnetic resonance imaging Background . . . 5
- 2.2 Brain Anatomy Background . . . 6
- 2.3 Generative models . . . 8
  - 2.3.1 Generative Adversarial Networks (Generative Adversarial Network (GANs)) . . . 10
  - 2.3.2 Variational Autoencoders . . . 12
  - 2.3.3 Diffusion Models . . . 14
  - 2.3.4 Flow based Generative models . . . 16
  - 2.3.5 Flow Matching for Generative Modeling . . . 21

3 Framework Selection for Synthetic Medical Image Generation 24
- 3.1 The Challenges of Synthetic Medical Imaging . . . 24
- 3.2 Comparison of Selected Generative Models . . . 25
  - 3.2.1 GANs . . . 25
  - 3.2.2 Variational Autoencoder (VAEs) . . . 25
  - 3.2.3 Denoising Diffusion models . . . 26
  - 3.2.4 Continuous Normalizing Flows . . . 27
  - 3.2.5 Flow Matching . . . 27
- 3.3 Choosing the Desired Model . . . 28
- 3.4 Vector Quantized Generative Adversarial Network (VQ-GAN) 29
  - 3.4.1 Structure of the VQ-GAN . . . 31
- 3.5 Flow Matching Model . . . 34
  - 3.5.1 Flow Dynamics and Loss Optimization . . . 34

4 Data and Developed framework 37
- 4.1 Model Architecture . . . 37
- 4.2 Implementation and Training . . . 39
  - 4.2.1 Data Preparation . . . 39
  - 4.2.2 VQ-GAN Implementation . . . 41
  - 4.2.3 Flow Matching Model Implementation . . . 42

5 Experiment and Result 46
- 5.1 Experiment Setup . . . 46
- 5.2 Testing Process . . . 47
- 5.3 First Experiment . . . 47
  - 5.3.1 Flow Matching Model . . . 48
  - 5.3.2 Conclusion of First Experiment . . . 50
- 5.4 Second Experiment . . . 52

5.5 Evaluation 54
5.5.1 Quantitative Results 54
5.5.2 Qualitative results 56
6 Discussion 59
6.1 Analysis of Quantitative and Qualitative Results 59
6.2 Impact of Dataset Characteristics 61
6.3 Significance of Preprocessing 62
6.4 Possible Areas for Improvement 62
7 Conclusions 65

Bibliography 66

LIST OF FIGURES

Figure 2.1 Visualization of the three Magnetic Resonance Imaging (MRI) imaging planes. The axial plane (top-left) divides the brain horizontally, capturing cross-sectional images. The coronal plane (top-center) divides the brain vertically into anterior (front) and posterior (back) sections. The sagittal plane (top-right) divides the brain into left and right sections. Below each plane illustration, corresponding MRI images demonstrate the detailed structures observed in these views [49]. . . 6

Figure 2.2 Some of the brain anatomy visible in this sagittal MRI image, including the brainstem, cerebrum, and other surrounding structures 8

Figure 2.3 Taxonomy of deep generative models based on their approach to modeling probability densities, divided into explicit density models (tractable or approximate) and implicit density models (direct or Markov chain-based) [53]. 9

Figure 2.4 The generative model aims to approximate the unknown data distribution P_{data} by learning from samples. The objective is to estimate P_{model} such that it closely matches P_{data}, enabling the generation of synthetic samples that resemble real data [12]. 10

Figure 2.5 GAN Architecture for Synthetic Image Generation: A random noise vector $z \sim \mathcal{N}(0, I)$ is fed into the generator G, which produces a synthetic image $G(z)$. The discriminator D evaluates both real training images x and generated images $G(z)$, predicting the probability that each is real or fake. Based on the discriminators feedback, a loss is computed to update both G and D, improving the generators ability to create realistic images [12]. 12

Figure 2.6 VAEs Architecture: The encoder $E(x)$ maps the input x to a latent distribution characterized by μ and σ^2. Using the reparameterization trick, latent variables are sampled as $z = \mu + \sigma \odot \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$. The decoder $D(z)$ reconstructs the input as x' [12]. 13

Figure 2.7 Diffusion model process, consisting of a fixed forward process and a learned reverse process. In the forward process (top), the original sample x_0 is progressively corrupted with Gaussian noise, resulting in a noise dominated sample x_T. In the reverse process (bottom), a neural network models each denoising step $p_\theta(x_{t-1} \mid x_t)$ to reconstruct the original data, enabling new sample generation by reversing the diffusion [12]. 15

Figure 2.8 Illustration of Normalizing Flows: The process of transforming a complex data distribution $\rho_x(x)$ (left) into a simpler latent distribution $p_z(z)$ (right) using an invertible function f. This function f maps data x to a latent variable z and can be reversed using f^{-1}. 16

Figure 2.9 Simple example of Normalizing Flows. The figure shows the transformation process in normalizing flows. The top arrow shows the forward transformations $f = f_4 \circ f_3 \circ f_2 \circ f_1$, which map the complex data distribution $\rho_x(x)$ to a simpler latent distribution $p_z(z)$. The bottom arrow shows the reverse transformations $f^{-1} = f_1^{-1} \circ f_2^{-1} \circ f_3^{-1} \circ f_4^{-1}$, which reconstruct the data from the latent space. Each transformation step must be invertible and diiferentable to make the process possible 19

Figure 3.1 VQ-GAN architecture: The model uses an encoder E to generate latent representations, which are quantized via a codebook Z. A transformer learns dependencies between quantized codes for autoregressive image generation. The decoder G reconstructs the image, and a discriminator D enhances image quality via adversarial training. The model is optimized with a combined vector quantization, adversarial, and autoregressive loss [10]. 31

Figure 4.1 Two-Stage Model Architecture for 3D Medical Image Generation. The framework consists of two stages. In the first stage, a VQ-GAN model is trained to encode input images into a discrete latent space ($\hat{z}$) using a CNN encoder (E) and a codebook ($\mathcal{Z}$), then reconstruct the images via a CNN decoder (G), then an adversarial training procedure and a patch-based discriminator (D) applied to differentiate between real and reconstructed images. The trained VQ-GAN model is then used in the second stage. In the second stage, a Latent Flow Matching framework is applied, where the input data x is encoded with the pre-trained encoder of the first stage model to produce the latent representation z_0. The latent flow network predicts the velocity of the transformation from a standard normal distribution $p(z_1) = \mathcal{N}(0, I)$ to the target latent distribution $p(z_0)$. During sampling, random noise z_1 is drawn from $p(z_1)$, and the network predicts the velocity towards $p(z_0)$ via numerical integration. Finally, z_0 is decoded with the VQ-GAN decoder from the first stage to generate the image. 38
Figure 4.2 MRI Brain Scans Demonstrating Stages of Alzheimer's Disease Progression. (A) Normal brain structure; (B) Early-stage Alzheimer's with mild atrophy; (C) Moderate-stage Alzheimer's showing significant atrophy and enlargement of ventricles; (D) Advanced-stage Alzheimer's with pronounced brain shrinkage and severe ventricular enlargement. 40
Figure 5.1 Example output from the diffusion model. 48
Figure 5.2 Initial results from the flow matching model, showing poor performance. 49
Figure 5.3 Results from the flow matching model using L1 loss. . 49
Figure 5.4 Results from the flow matching model using a combination of L1 and L2 loss functions. 50
Figure 5.5 Loss function trends: gray represents L2 (0.035), yellow represents L1 (0.042), and green represents a combination of 0.9L1 + 0.1L2 (0.052). 50
Figure 5.6 Examples of input data after preprocessing, prior to being fed into the networks. 51
Figure 5.7 Examples of reconstructed data from the VQ-GAN output. 51
Figure 5.8 Input data after updated preprocessing and normalization. This adjustment allowed the VQ-GAN to process the data more effectively. 52

Figure 5.9 Reconstructed output of the VQ-GAN model after improved preprocessing. The reconstruction quality is significantly enhanced compared to Experiment 1. . . . 53
Figure 5.10 Results of the flow matching model. Improved preprocessing and reconstruction have led to clearer brain structures and better overall image quality. 53
Figure 5.11 Comparison of generated MRI images (left) with ground truth images (right). While the overall structure is well-represented in the generated images, the fine details are not fully captured, and the generated images exhibit slight blurriness. 57
Figure 5.12 Artifacts observed in the parietal and occipital lobes of generated MRI images. These grid-like noise patterns, highlighted in red, do not correspond to anatomically accurate structures and indicate inconsistencies in the model's output 58
Figure 6.1 Examples of 3D slices generated by the flow matching model. The images illustrate variability in contrast and fine details, showcasing the model's ability to produce diverse representations while maintaining overall anatomical structure. 60

LIST OF TABLES

Table 3.1 Comparison of Selected Generative Models for Medical Imaging 36
Table 4.1 Encoder Architecture 39
Table 4.2 Decoder Architecture 39
Table 5.1 Model Parameter Breakdown 46
Table 5.2 Quantitative Evaluation Metrics 54

ABKÜRZUNGSVERZEICHNIS

MRI	Magnetic Resonance Imaging
CSF	Cerebrospinal Fluid
GANs	Generative Adversarial Network
VAEs	Variational Autoencoder
ELBO	Evidence Lower Bound
DDPMs	Denoising Diffusion Probabilistic Model
CNFs	Continuous Normalizing Flows
ODE	Ordinary Differential Equation
FM	Flow Matching
SD3	Stable Diffusion 3
WGAN	Wasserstein Generative Adversarial Network
WGAN-GP	WGAN with Gradient Penalty
LSGAN	Least Squares GAN
MSE	Mean Squared Error
NMSE	Normalized Mean Squared Error
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Index Measure
MS-SSIM	Multi-Scale Structural Similarity Index Measure
DIT	Diffusion Transformer
EMA	Exponential Moving Average
ADNI	The Alzheimer's Disease Neuroimaging Initiative
VQ-GAN	Vector Quantized Generative Adversarial Network
VQ-VAE	Vector Quantized Variational Autoencoder
VP	Variance Preserving

Part I

THESIS

1 INTRODUCTION

1.1 MOTIVATION

Following the considerable success of deep learning in many applications, its potential for medical applications has also started to be investigated. However, the application of deep learning in medical contexts faces a significant challenge: the scarcity of data.

Unlike other fields where large datasets can be easily accessed, medical data collection is inherently challenging. Data is often collected intermittently across different clinical settings and typically only during necessary procedures. Consequently, there is a limited pool of data available for training deep learning models. Moreover, the inherent variability of medical pathologies necessitates large and diverse datasets for effective model training. For instance, tumors can manifest in numerous shapes, appearances, and locations, and some of them might be rarer than others.Furthermore, the protection of data privacy increases the difficulty of obtaining sufficient data for medical applications. This is particularly challenging when the data originates from different institutions. The generation of realistic artificial data can help to a certain extent to overcome these limitations.

The accurate generation of artificial data has numerous advantages. It allows the application of deep learning techniques in rare diseases for which not enough data is available. Furthermore, the generation of realistic artificial data can help to reduce the effects of data imbalance. This is particularly relevant when the data collected shows only one aspect of a certain pathology, or when considering the different acquisition protocols of different institutions. So there is always need to develop a methodology for the generation of synthetic medical data [11, 30].

The methodology will be designed to address the data scarcity problem prevalent in medical deep learning applications. It will provide a means of creating synthetic datasets that closely mimic real world medical data.

1.2 AIM OF THIS THESIS

The release of the Flow Matching (FM) framework [29] marked a major milestone in generative modeling, offering new possibilities for training models that address the limitations of diffusion and continuous normalizing flow (Continuous Normalizing Flows (CNFs)) methods. Since its introduction, the FM framework has inspired numerous adaptations and extensions over the past two years, demonstrating its potential to tackle diverse modeling challenges.

Flow matching models have achieved notable success in non-medical domains due to their ability to generate a wide range of high-quality images. Recent studies indicate that flow matching approaches outperform diffusion models in terms of faster sampling and superior generation quality [29]. Notably, the well known approach **Stable Diffusion 3** (Stable Diffusion 3 (SD3)) [9] an advanced text-to-image generation model developed by Stability AI, integrates also flow matching with a diffusion transformer design, further highlighting the framework's advancements.

Despite their significant performance improvements, flow matching models have yet to be systematically applied to three-dimensional image generation in medical contexts. This gap is particularly evident in medical fields, where generating realistic 3D medical images, such as MRI scans, is of critical importance.

To address this research gap, this thesis investigates the potential of flow matching models for the generation of 3D medical images. Specifically, it introduces a novel architecture for a flow matching model that operates in the latent space and evaluates its performance on brain MRI data sourced from publicly available datasets. This work aims to demonstrate the applicability of flow matching models to medical imaging and contribute to advancements in synthetic data generation for healthcare applications.

Research Questions

This work is guided by the following research questions:

1. How effective is the flow matching framework for generating realistic and high-quality 3D medical images, specifically brain MRI scans?
2. What are the benefits and challenges of combining VQ-GAN and flow matching models for latent-space representation and synthetic image generation?
3. Can a flow matching model operating in the latent space improve computational efficiency without compromising on generation quality?
4. What are the challenges and limitations of applying flow matching models to medical imaging, and how can they be addressed?

Objectives

To answer these research questions, the following objectives are defined:

1. To develop a novel framework by combining VQ-GAN and Flow Matching models for generating realistic and high-quality 3D medical images.
2. To evaluate the performance of the proposed model on brain MRI data in terms of quantitative and qualitative metrics.

3. To analyze the impact of preprocessing techniques, architectural integration, and computational constraints on the effectiveness of the framework.

4. To identify key limitations and propose future directions for improving the combination of VQ-GAN and Flow Matching in medical imaging applications.

1.3 STRUCTURE

This thesis is organized into following chapters, each addressing a specific aspect of the research and its outcomes:

- **Chapter 2: Theoretical Background** This chapter provides an overview of the foundational concepts necessary for understanding the research. It introduces the principles of MRI imaging, brain anatomy, and generative modeling. Specific attention is given to Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), Diffusion Models, and Flow Matching frameworks.

- **Chapter 3: Framework Selection for Synthetic Medical Image Generation** This chapter outlines the process of selecting suitable generative models for this research. It includes a comparison of various approaches, such as GANs, VAEs, diffusion models, continuous normalizing flows, and flow matching. Based on this comparison, the chapter justifies the choice of the Flow Matching model in combination with the VQ-GAN.

- **Chapter 4: Data and Developed framework** This chapter details the architecture and implementation of the proposed framework. It discusses the preprocessing of the dataset, the structure of the VQ-GAN, and the design of the flow matching model. The implementation and training strategies are also presented, highlighting the steps involved in integrating these components.

- **Chapter 5: Experiment and Results** This chapter presents the experiments conducted to evaluate the proposed model. It begins with an explanation of the experiment setup, followed by the results of the first and second experiments. Both quantitative and qualitative evaluations are included to assess the model's performance.

- **Chapter 6: Discussion** This chapter provides a critical analysis of the results and their implications. It examines the strengths and limitations of the proposed approach and discusses the impact of dataset characteristics and preprocessing strategies. Suggestions for improving the framework are also presented.

- **Chapter 7: Conclusion** The thesis concludes with a summary of the findings and their contributions to the field of generative modeling for 3D medical imaging.

2 THEORETICAL BACKGROUND

This chapter covers the fundamental theoretical background of generative models, with a focus on their application in generating synthetic images. It begins with an introduction to MRI technology, including a basic overview of brain anatomy, followed by a detailed explanation of various generative models, such as GANs, VAEs, Diffusion Models, and Flow-based Generative Models.

2.1 MAGNETIC RESONANCE IMAGING BACKGROUND

Magnetic resonance imaging (MRI) is a medical imaging technique that takes advantage of the use of magnetic fields and radio waves to generate highly detailed images of the body's internal organs.

MRI has been shown to be an effective method for examining the brain and detecting injuries, as it is capable of visualizing soft tissues such as brain structures with high resolution.

Each MRI image is displayed in three distinct planes, namely the sagittal, axial, and coronal planes, providing a three-dimensional understanding of the structures being examined.

The sagittal plane is a section that divides the body into two parts, the left and right. The image is presented as if the brain were being observed from a lateral perspective.

The axial plane represents a horizontal slice through the body, with the upper and lower parts divided by a plane of symmetry. The image is presented as if observed from an axial angle, either from above or below.

The coronal plane represents a vertical section that divides the body into front and back parts, presenting the image as if viewed from the front. The combination of these three planes provides a three-dimensional representation of the brain, which enables a radiologist to interpret the image and identify diseases, injuries, or changes in the brain [4], see Figure 2.1.

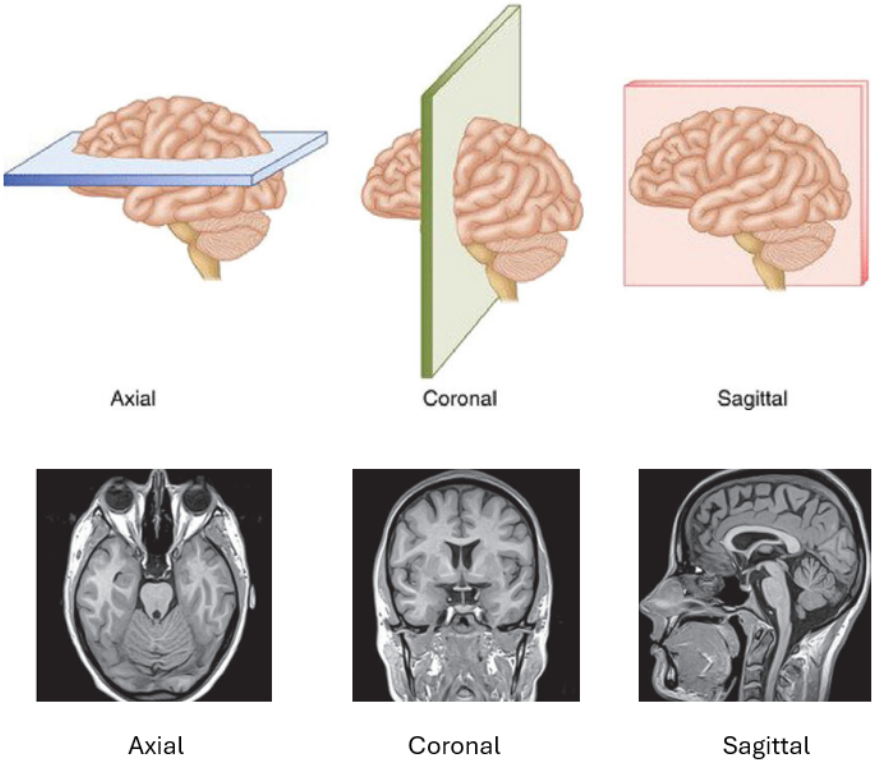

Figure 2.1: Visualization of the three MRI imaging planes. The axial plane (top-left) divides the brain horizontally, capturing cross-sectional images. The coronal plane (top-center) divides the brain vertically into anterior (front) and posterior (back) sections. The sagittal plane (top-right) divides the brain into left and right sections. Below each plane illustration, corresponding MRI images demonstrate the detailed structures observed in these views [49].

2.2 BRAIN ANATOMY BACKGROUND

The human nervous system is divided into two major components: (i) the brain and (ii) the spinal cord. The brain serves as the control center for the nervous system and is subdivided into several regions: the cerebrum, cerebellum, and brain stem, each responsible for different functions.

THE CEREBRUM

The cerebrum is the largest part of the brain and is crucial for functions such as thought processing, language, memory, and voluntary movement. The cerebrum is split into two hemispheres, each controlling functions on the opposite side of the body. It is further divided into distinct lobes, each responsible for specific tasks:

- **Frontal Lobe**: Located at the front of the cerebrum, it is involved in personality, decision-making, and voluntary motor control.

- **Temporal Lobe**: Found on the sides of the cerebrum, it handles functions like hearing and language comprehension.
- **Parietal Lobe**: Situated at the top of the cerebrum, it is involved in sensory processing and tasks such as mathematical abilities and problem-solving.
- **Occipital Lobe**: Located at the rear of the cerebrum, it is primarily responsible for visual processing.

The surface of the cerebrum is marked by pronounced folds, known as *cerebral convolutions*, separated by grooves filled with cerebral vessels.

THE CEREBELLUM

Positioned below the cerebrum, the cerebellum is responsible for motor coordination and maintaining balance and equilibrium.

THE BRAIN STEM

Acting as the bridge between the brain and the spinal cord, the brain stem controls vital life functions such as respiration, heart rate, and blood pressure.

PROTECTION AND SUPPORT

The brain is shielded by the **cranium**, a bony structure, and three protective layers of *meninges*. Additionally, the brain is surrounded by **cerebrospinal fluid (Cerebrospinal Fluid (CSF))**, which provides cushioning, transports nutrients, and removes waste products. CSF is produced in the brain's ventricles and circulates throughout, performing vital roles in protecting and nourishing the brain [35].

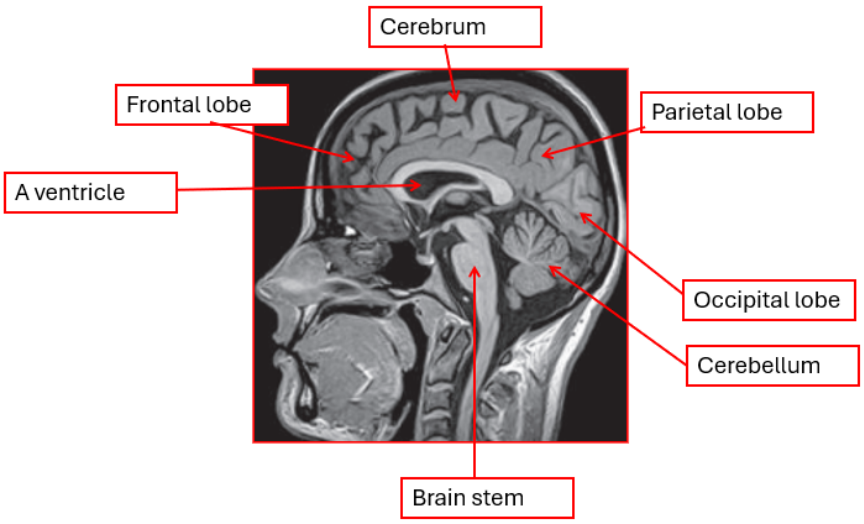

Figure 2.2: Some of the brain anatomy visible in this sagittal MRI image, including the brainstem, cerebrum, and other surrounding structures

2.3 GENERATIVE MODELS

In the field of artificial intelligence, generative models represent a collection of methods and models developed to learn the underlying structure of a dataset and to extract its key features. The main goal is to generate new instances that share similar characteristics with the original dataset.

Assuming that the training data are sampled from an unknown distribution $x \sim p_{\text{data}}(x)$. The generative model $p_\theta(x)$ which represents a parametrized family of probability distributions, is trained to estimate p_{data}, this training process involves optimizing the model's parameters θ so that $p_\theta(x)$ comes as close as possible to the true underlying distribution $p_{\text{data}}(x)$ [27]. The Estimation of the probability distribution of the training data is provided by the principle of maximum likelihood by direct and undirekt way. The reason of the estimation because in practice there is no access to the p_{data} itself only to a training set consisting of a certain number of samples from data distribution. By using them to define $\hat{p}_{data}$, an empirical distribution that places mass only on exactly those amount of the training data points and aproximating p_{data} [14]. As shown in Figure 2.4, the generative model aims to approximate the unknown data distribution. The deep generative models that work by maximizing the likelihood can be divided into two groups: (i) the group that calculates either the likelihood and its gradients (Explicit Density Models). (ii) the group that approximates these quantities (implicit density models). The construction of the taxonomy of generative models is shown in Figure 2.3. Explicit density models define a probability distribution $p_{\text{model}}(x;\theta)$, and maximizing the likelihood involves plugging the density function into the likelihood expression and computing the gradient. The main challenge is to design a model that captures the complexity of the data while remaining computationally tractable. This is addressed by two strategies: (1) constructing models with inherent tractability, such as change of

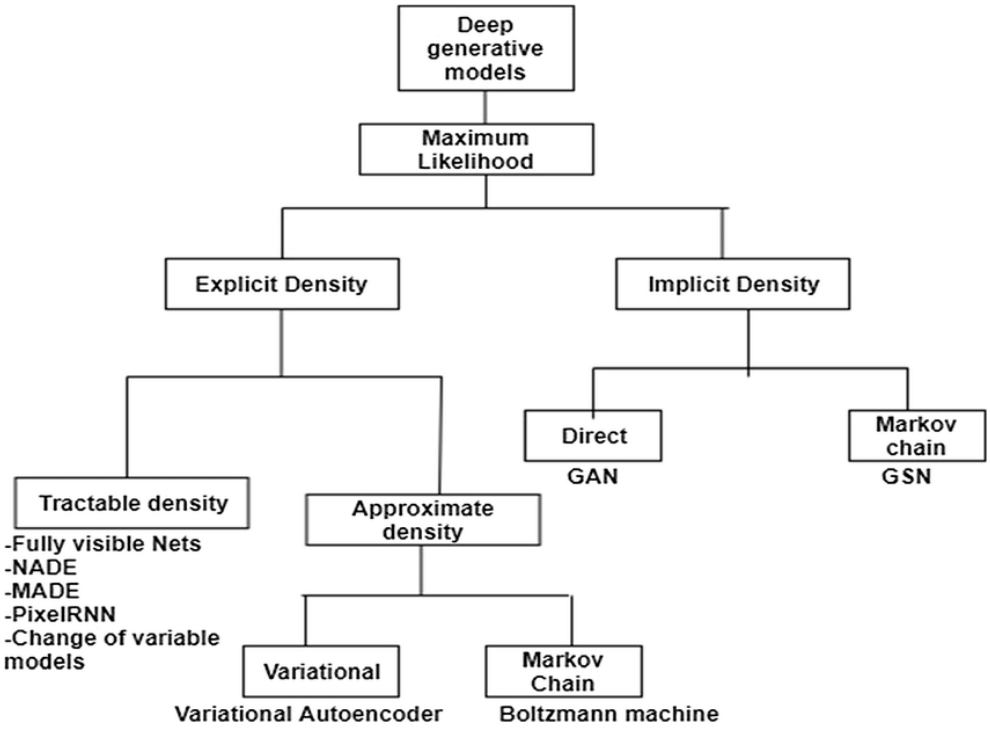

Figure 2.3: Taxonomy of deep generative models based on their approach to modeling probability densities, divided into explicit density models (tractable or approximate) and implicit density models (direct or Markov chain-based) [53].

variable models, and (2) relying on approximate variational methods, such as variational autoencoders (VAEs). In contrast, implicit density models, such as GANs, directly generate samples without explicitly modeling the density, while Markov chain-based models like GSNs rely on iterative refinement [14].

There are many benefits to trained generative models. One of the main advantages these models offer is their ability to create new data that extends beyond the original training data, incorporating additional computations and different details. This capability makes it possible to produce an unlimited number of data points simply by sampling from the model's distribution. The ability to synthesize data has become a valuable tool in medical imaging, providing a practical solution to the challenge of data sharing while maintaining patient privacy. Although medical image synthesis presents significant challenges, especially considering the complexity of medical conditions observed in three-dimensional images such as magnetic resonance imaging (MRI) and computed tomography (CT) scans, several powerful deep learning models are currently available. These models are capable of learning complex data distributions, including diffusion models, autoregressive transformers, generative adversarial networks (GANs), and variational autoencoders (VAEs)[39]. The following subsections offer a closer look at specific types of generative models, such as GANs and autoencoders. These will

cover how these models are built, their operational principles, and the potential applications they provide.

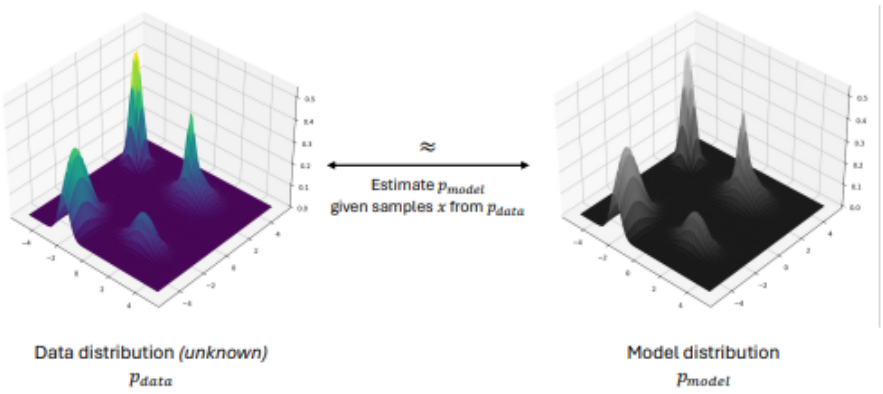

Figure 2.4: The generative model aims to approximate the unknown data distribution P_{data} by learning from samples. The objective is to estimate P_{model} such that it closely matches P_{data}, enabling the generation of synthetic samples that resemble real data [12].

2.3.1 *Generative Adversarial Networks (GANs)*

Generative adversarial Networks are a class of generative machine learning techniques. The concept of GANs was introduced at first time in 2014 by Ian Goodfellow [13]. This technology has led to major movement in artificial intelligence, particularly in fields requiring data synthesis, like data augmentation, image generation and style transfer. The framework of GANs (as shown in Figure 2.5) consists of two neural networks, a generator G and a discriminator D which have adversarial relationship. The generator as its name indicates generates new fake data which simulates input training data. The discriminator tries to distinguish between the real training data and the fake images that have been generated from the generator . In their competition, each of them have distinct and opposing goals. The GAN framework is formulated as a zero-sum minimax game, creating a dynamic where the loss function of each network is balanced by that of the other. This leads in a system in which the networks iteratively adapt to reach a Nash equilibrium where none can improve its performance without compromising the other, ideally leading to a balance where the discriminator can no longer easily distinguish between real and generated data.[40]

GANs have gained significant attention in image generation, due to their powerful ability to not only generate clear and plausible fake images that simulate the real data, but also create various details that are not present in the input data. For example, they can simulate variations in lighting, texture, or even specific visual attributes, making it easier to study different conditions or create different styles. The flexibility of GANs to manipulate and create new visual elements opens up many possibilities in research and

industry, specially in healthcare, where realistic synthetic data can accelerate model training and help in studying scenarios where real world data may be limited or difficult to obtain [52].

Despite the significant success of GANs, they have notable limitations. One major issue is mode collapse, where the fully trained model generates a limited range of synthetic data rather than a diverse array of outputs. This problem prevents GANs from capturing the full diversity of the training data.

Additionally, GANs suffer from gradient vanishing and training instability due to the adversarial nature of the training process. This instability can lead to oscillations or divergence during training, hindering model convergence. To overcome these problems, several GAN modifications have been introduced that adopt better training techniques, regularization strategies, or loss modifications. Among them are Wasserstein GAN (Wasserstein Generative Adversarial Network (WGAN)) [2], WGAN with Gradient Penalty (WGAN with Gradient Penalty (WGAN-GP)) [17], Spectral Normalization GAN (SNGAN) [36], or Least Squares GAN (Least Squares GAN (LSGAN))[34]. These adjustments have successfully reduced GAN-related problems, but do not completely eliminate them.

Another key limitation is the difficulty of evaluating the quality of generated data. Traditional metrics, such as accuracy or error rates, often fail to effectively capture the differences between original and synthesized data distributions. Furthermore, no single evaluation metric can universally apply across all GAN applications.

To address these challenges, researchers have developed specialized metrics like the Inception Score (IS), which measures image diversity and clarity, and the Fréchet Inception Distance (FID), which compares the distribution of generated and real images. New methods and techniques are also being explored to overcome these limitations, including combining GANs with other generative models to enhance performance and robustness. [20, 28]

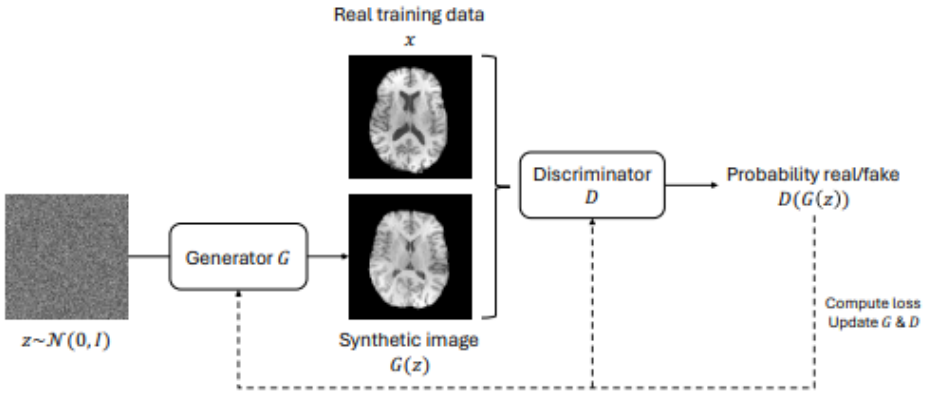

Figure 2.5: GAN Architecture for Synthetic Image Generation: A random noise vector $z \sim \mathcal{N}(0, I)$ is fed into the generator G, which produces a synthetic image $G(z)$. The discriminator D evaluates both real training images x and generated images $G(z)$, predicting the probability that each is real or fake. Based on the discriminators feedback, a loss is computed to update both G and D, improving the generators ability to create realistic images [12].

2.3.2 *Variational Autoencoders*

Variational Autoencoders (VAEs) are generative models that learn a probabilistic representation of input data in a lower-dimensional latent space, enabling both reconstruction of the input and generation of new, similar samples. A VAEs consists of two primary components:

- **Encoder**: Maps the input data to a latent space by generating a distribution parameterized by a mean μ and variance σ^2.
- **Decoder**: Reconstructs the data from the latent representation back into the output space.

The general architecture of a VAEs framework is shown in Figure 2.6.

The term "latent space" refers to a simplified representation of complex data (e.g., high-resolution images, audio, or text). By compressing the data into this latent space, VAEs make it easier to manipulate and analyze the data. This representation is particularly useful for applications requiring interpolation, sampling, or controlled data generation.

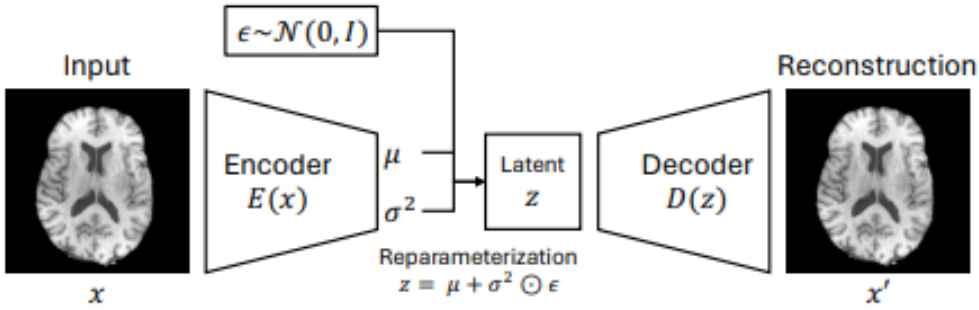

Figure 2.6: VAEs Architecture: The encoder $E(x)$ maps the input x to a latent distribution characterized by μ and σ^2. Using the reparameterization trick, latent variables are sampled as $z = \mu + \sigma \odot \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$. The decoder $D(z)$ reconstructs the input as x' [12].

To learn this latent representation, VAEs optimize the Evidence Lower Bound (Evidence Lower Bound (ELBO)), which maximizes the log-likelihood of the observed data while ensuring the latent space approximates a standard Gaussian distribution. The ELBO objective function is expressed as:

$$\mathcal{L}_{\text{ELBO}} = -D_{KL}(q_\phi(z|x) \parallel p(z)) + \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x|z)]$$

Here:

- $q_\phi(z|x)$: Approximate posterior distribution generated by the encoder.
- $p(z)$: Prior distribution (typically Gaussian $\mathcal{N}(0, I)$).
- $p_\theta(x|z)$: Likelihood function parameterized by the decoder.

The ELBO comprises two components:

1. **KL Divergence**: A regularization term that encourages the approximate posterior distribution $q_\phi(z|x)$ to align with the prior $p(z)$, ensuring a smooth and structured latent space.
2. **Reconstruction Loss**: Measures the accuracy of the decoder in reconstructing the input data x from the latent variable z, encouraging faithful reconstructions [24, 28].

By maximizing the ELBO, VAEs achieve a balance between learning a compact latent representation and generating new data samples. This dual capability makes VAEs valuable for tasks requiring controlled variability in data generation, such as anomaly detection and semi-supervised learning.

However, VAEs face notable challenges:

- **Blurry Outputs**: The use of an L2-norm-based reconstruction loss tends to average out fine details, leading to blurry images in the generated output [24].
- **Posterior Collapse**: Over-regularization by the KL term can cause the latent space to carry minimal information about the input, a phenomenon known as posterior collapse.

To mitigate these issues, researchers have explored strategies such as:

- Dynamically adjusting the weight of the KL divergence term during training.
- Introducing advanced loss functions that balance reconstruction quality and latent space regularization.
- Employing hierarchical VAEs or incorporating skip connections in the decoder to improve detail preservation.

These enhancements aim to combine the advantages of VAEs (variability in generated samples) with improved detail and image quality, making VAEs more suitable for high-fidelity generative tasks [1].

2.3.3 *Diffusion Models*

Diffusion models are a powerful class of generative models that have gained significant attention in recent years, particularly for generating high-quality images and other complex data distributions. The main idea behind these models is presented in [45], where the authors describe it as follows:

> "The essential idea, inspired by non-equilibrium statistical physics, is to systematically and slowly destroy structure in a data distribution through an iterative forward diffusion process. We then learn a reverse diffusion process that restores structure in data, yielding a highly flexible and tractable generative model of the data."

It can be understood that diffusion models are based on the concept of two-step processes (as shown in Figure 2.7), a forward diffusion process and a reverse diffusion process. In the forward process the input data distribution, which is complex and unknown is gradually converted using Markov chain to a known simple distribution usually Gaussian. The reverse diffusion process then applies the reverse conversion from the Gaussian distribution to a an approximate distribution of the generated data, with the aim of estimating the true underlying distribution of the input data [45].

Mathematical Explanation of Diffusion Processes

The forward diffusion process begins with data drawn from the initial distribution $q(x_0)$ and progressively adds noise at each step until the data approximates a standard Gaussian distribution. Each state at time t can be represented as:

$$q(x_{1:T}|x_0) := \prod_{t=1}^{T} q(x_t|x_{t-1}) \tag{2.1}$$

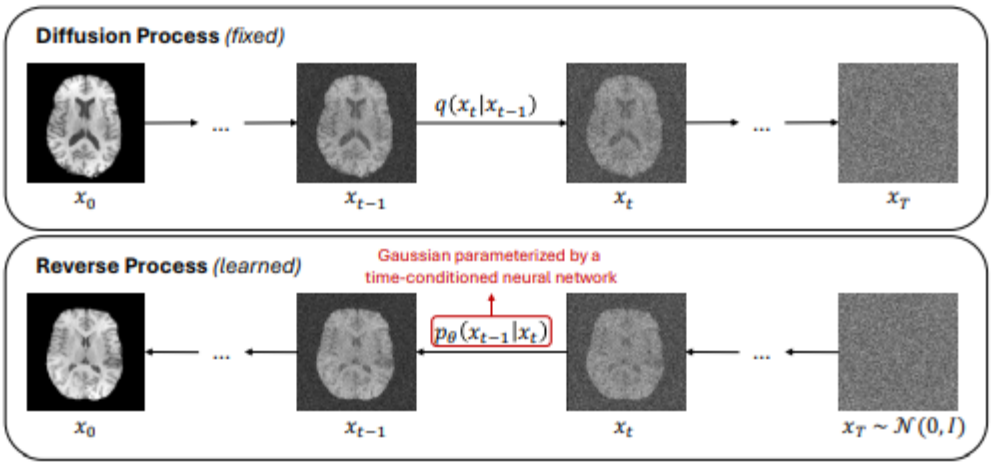

Figure 2.7: Diffusion model process, consisting of a fixed forward process and a learned reverse process. In the forward process (top), the original sample x_0 is progressively corrupted with Gaussian noise, resulting in a noise dominated sample x_T. In the reverse process (bottom), a neural network models each denoising step $p_\theta(x_{t-1} \mid x_t)$ to reconstruct the original data, enabling new sample generation by reversing the diffusion [12].

where each transition in the forward diffusion process is defined by:

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}\, x_{t-1}, \beta_t I), \tag{2.2}$$

where β_t represents the variance introduced at each step, and I is the identity matrix.

The distribution of x_t conditioned on the original data x_0 can further be expressed as:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}\, x_0, (1-\bar{\alpha}_t) I), \tag{2.3}$$

where $\bar{\alpha}_t$ is the cumulative product of all previous α values, regulating the level of noise across multiple time steps.

The reverse diffusion process then reconstructs the data by transforming this Gaussian distribution back toward the original data distribution. This can be expressed as:

$$p_\theta(x_{0:T}) := p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t), \tag{2.4}$$

where each reverse step is defined by:

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \tag{2.5}$$

where $\mu_\theta(x_t, t)$ and $\Sigma_\theta(x_t, t)$ are learned parameters that guide the denoising process. By iteratively applying this reverse process, starting from x_T

and working back to x_0, the model generates samples that approximate the original data distribution.[19]

The diffusion model has been shown to be a highly effective tool in the field of generative modeling. Its popularity can be attested by the numerous variations and diverse structures of models that have emerged in this area. The famous diffusion framework is the Denoising Diffusion Probabilistic Models (Denoising Diffusion Probabilistic Model (DDPMs)) which have got attention specially after the openAI group adopted its framework to their application and became one of the most well-known diffusion models. This model has gained popularity for its impressive ability to create high-quality and diverse images based on the prompts it receives [51]

2.3.4 *Flow based Generative models*

Suppose $x = \{x_1, x_2, \ldots, x_n\}$ is an image, which consists of pixels that follow a distribution $p(x)$. This distribution is typically complex and unknown. The idea behind flow-based models is to apply a sequence of mathematical operations (transformations) $z = f(x)$, which maps the data from its complex distribution $p(x)$ to a simple latent distribution $p_z(z)$ (often a Gaussian). Sampling from this simple latent distribution can then be reversed using f^{-1} to approximate the original data distribution (probability estimating). This reversal allows for the reconstruction of the noise, yielding generated data that resembles the underlying real data (see Figure 2.8).

For accurate probability estimate and density calculation at any point, each transformation needs to be both differentiable and invertible. This property is important in applications where precise probability estimates are important. Because of this ability to estimate densities exactly, flow-based models are sometimes referred to as "normalizing flows" [41].

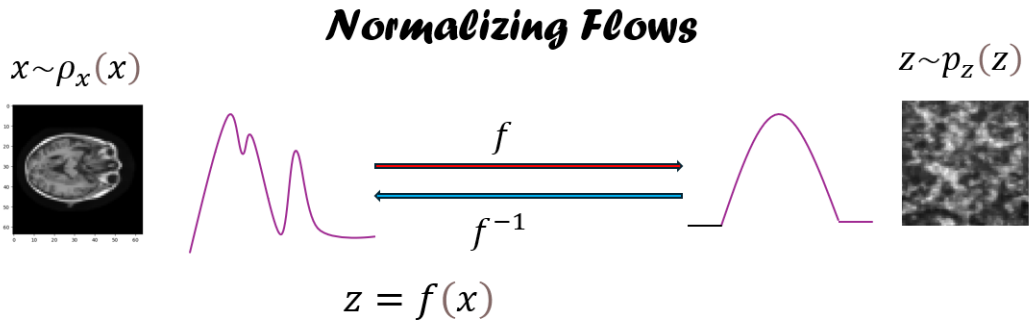

Figure 2.8: Illustration of Normalizing Flows: The process of transforming a complex data distribution $p_x(x)$ (left) into a simpler latent distribution $p_z(z)$ (right) using an invertible function f. This function f maps data x to a latent variable z and can be reversed using f^{-1}.

Mathematical Explanation of Flow-Based Generative Models

As explained above the normalizing flow use a series of invertible transformations to map data between a simple and complex distribution. These models allow for exact density estimation and likelihood computation, a property that makes them useful for many applications in generative modeling.

Suppose $Z \in \mathbb{R}^D$ is a random variable sampled from a simple, known probability density function $p_Z : \mathbb{R}^D \to \mathbb{R}$, and $X = f(Z)$ is the transformed variable obtained through an invertible function f. The transformation $X = f(Z)$ maps the simple distribution p_Z to the complex distribution p_X. The probability density $p_X(x)$ of X can be computed using the change of variables formula:

$$p_X(x) = p_Z(f^{-1}(x)) \left| \det \frac{\partial f^{-1}(x)}{\partial x} \right| \tag{2.6}$$

f^{-1} is the inverse of the transformation f. And $\frac{\partial f(x)}{\partial x}$ is the Jacobian matrix of f^{-1} at x, which describes the partial derivatives of each component of f^{-1} with respect to each component of x. The new density function $p_X(x)$ is called *pushforward*.

The concept of a pushforward density is central to flow-based generative models. The function f^{-1} "pushes forward" the simple base density p_Z to a more complex target density p_X. This transition is called "the generative direction". The inverse direction which uses the inverse f^{-1} to map x back to z, represents the " normalizing direction". This reverse direction "normalizes" complex data back to the simple base distribution, enabling exact likelihood estimation. This bidirectional capability is what gives normalizing flows their name, as they allow data to "flow" back and forth between distributions.

Since flow-based models require exact density estimation, there are three main condition, need to be filled. The flow functions must be (i) invertible (ii) differentiable and (iii)The Jacobian determinant must be well-defined, non-zero, and computationally efficient to evaluate (This will be optimized during training and if the calculation is slow, then the training will be slow too). These properties allow for exact likelihood evaluation during model training.

In practice, by using a single transformation f is often difficult to construct complex mappings capable of representing high-dimensional data distributions, and still hold this conditions. Therefore, flow-based models obtain this by composing a series of simpler invertible (bijective) functions, such as:

$$f = f_1 \circ f_2 \circ \cdots \circ f_N$$

This composition is also invertible, with the inverse given by:

$$f^{-1} = f_N^{-1} \circ f_{N-1}^{-1} \circ \cdots \circ f_1^{-1}$$

A simple example to make it clear is illustrate in Figure 2.9. This property allows the construction of complex transformations over time, while ensuring the entire model remains tractable and invertible [26]. The training process involves minimizing the negative log-likelihood loss, as following:

$$\mathcal{L}(x) = -\log p_X(x) = -\log p_Z(f^{-1}(x)) - \log \left| \det \frac{\partial f^{-1}(x)}{\partial x} \right|. \tag{2.7}$$

This loss function has two components. The first term $-\log p_Z(f(x))$ represents the likelihood of the transformed data $f(x) = z$ under the simple base distribution p_Z, often assumed to be a standard Gaussian distribution. The second term $-\log \left| \det \frac{\partial f^{-1}(x)}{\partial x} \right|$ indicates to the transformation of space volume resulting from f, which ensuring that the model adapts for any scaling or stretching effects of f.

By minimizing this loss function, the model learns the parameters of f that best map the complex data distribution to the simple base distribution and vise verse [41].

To meet the conditions of the normalizing flow, many normalizing flow model are developed, each suited for specific data structures or designed to balance computational efficiency with model flexibility. The famous models are NICE, Real NVP, Glow, and Flow++ are examples of traditional normalizing flow models that work well with discrete data. However, they have certain architectural limitations, like using rank-one weight matrices or partitioning dimensions, to get around the computational difficulties involved in calculating large determinant costs. Another model which is called Continuous Normalizing Flows, have introduced a continuous time framework for normalizing flows. It used Ordinary differential equations to define the mapping from latent variables to data, providing more expressiveness and flexibility without the same degree of architectural restrictions [15].

2.3.4.1 *Continuous Normalizing Flow*

As explained in the last section, normalizing flows traditionally use a discrete sequence of invertible transformations to map complex data distributions to simpler latent distributions (e.g., a Gaussian). However, Continuous Normalizing Flows (CNFs) take this concept further by replacing these discrete transformations with a continuous transformation, modeled as the solution to an Ordinary Differential Equation (ODE). Instead of "jumping" between fixed transformation layers, CNFs allow data to "flow" smoothly and continuously over time.

There are several advantages to use Ordinary Differential Equation (ODE)solvers, namely [5]:

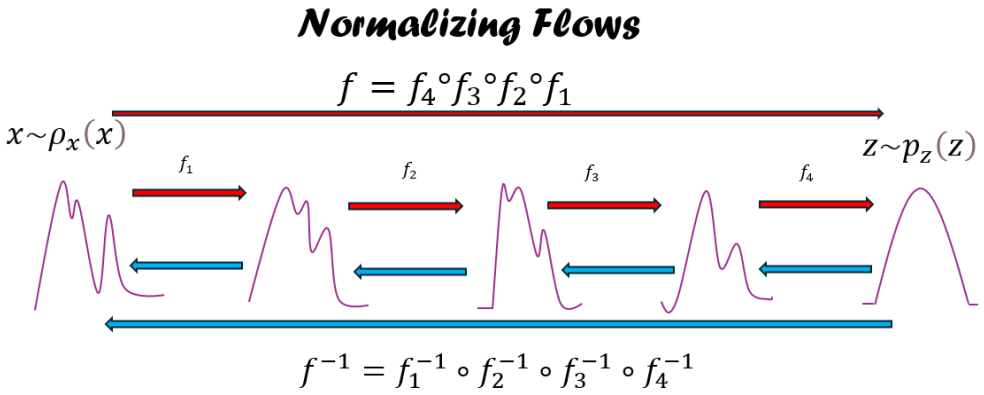

Figure 2.9: Simple example of Normalizing Flows. The figure shows the transformation process in normalizing flows. The top arrow shows the forward transformations $f = f_4 \circ f_3 \circ f_2 \circ f_1$, which map the complex data distribution $\rho_x(x)$ to a simpler latent distribution $p_z(z)$. The bottom arrow shows the reverse transformations $f^{-1} = f_1^{-1} \circ f_2^{-1} \circ f_3^{-1} \circ f_4^{-1}$, which reconstruct the data from the latent space. Each transformation step must be invertible and diiferentable to make the process possible

- ODE solvers use memory more efficiently than discrete normalizing flows and do not require the storage of intermediate states (e.g. outputs) of each layer f_i in a stack.
- Another benefit of ODE solvers is its adaptive computation. ODE solvers modify the number of steps they take dependent on the complexity of the transformation. For simpler transformations (when the data distribution is already close to the target) fewer steps are needed, saving computational resources. For complex transformations (when the data is far from the target distribution) the solver takes more steps to achieve the desired accuracy. This ensures that computation is used effectively where needed.
- CNFs handle irregular time-series data naturally by defining transformations continuously with ODEs, unlike RNNs that require discretized intervals. This flexibility makes CNFs suitable for modeling data with arbitrary sampling times
- Finally,in traditional normalizing flows, the change of variables formula for transforming probability densities involves computing the determinant of the Jacobian matrix (equation 2.7). For high-dimensional data, this can be computationally expensive and challenging. Instead CNFs use the instantaneous change of variables formula, which simplify this process by replacing the determinant with a continuous integration over time (equation 2.10). This makes the computation of density transformations more scalable and efficient.

Mathematical Explanation of CNF! (CNF!) Processes

The mathematical basis of continuous normalizing flows is based on modeling the transformations as continuous processes through ordinary differential equations (ODEs). The transformation of data points is given by the following equation:

$$f(\mathbf{z}(t), t; \theta) = \frac{d\mathbf{z}(t)}{dt}, \tag{2.8}$$

with initial condition $\mathbf{z}(0) \sim p(\mathbf{z}(0))$, a sample from a known simple distribution (Gaussian). $f(\mathbf{z}(t), t; \theta)$ is a vector field parameterized by θ that regulates the transformation. This function is often implemented as a neural network, which enables the model to learn complex mappings from a simple distribution to a complex one. The ODE solver computes the solution for $\mathbf{z}(T)$, yielding the transformed data at time T:

$$\mathbf{z}(T) = \text{ODE\_solver}\left(\frac{d\mathbf{z}(t)}{dt}, \mathbf{z}(0), t = 0, T\right). \tag{2.9}$$

To train the model, the log-likelihood of the transformed data $\mathbf{z}(T)$ need to computed. The log-density evolution along the transformation is given by the instantaneous change of variables formula [5]:

$$\frac{d \log p(\mathbf{z}(t))}{dt} = -\text{Tr}\left(\frac{\partial f(\mathbf{z}(t), t; \theta)}{\partial \mathbf{z}(t)}\right), \tag{2.10}$$

This term $\text{Tr}\left(\frac{\partial f}{\partial \mathbf{z}(t)}\right)$ represents the trace of the Jacobian matrix of f with respect to $\mathbf{z}(t)$. This trace term calculates the divergence of the transformation function f, which quantifies how the probability density changes over time.

To compute the log-density at the end time T, this expression is integrated from $t = 0$ to $t = T$:

$$\log p(\mathbf{z}(T)) = \log p(\mathbf{z}(0)) - \int_0^T \text{Tr}\left(\frac{\partial f(\mathbf{z}(t), t; \theta)}{\partial \mathbf{z}(t)}\right) dt. \tag{2.11}$$

In this equation, $\log p(\mathbf{z}(0))$ is the log density of the initial data sample.

To efficiently compute gradients in the backward Pass , CNFs use the adjoint sensitivity method. By solving an additional ODE backward in time, the gradients of the loss function are calculated with respect to parameters θ.

The adjoint state $\mathbf{a}(t)$ is defined, which captures the gradient of the loss $\mathcal{L}$ with respect to $\mathbf{z}(t)$.
The backward ODE for $\mathbf{a}(T) = \frac{\partial \mathcal{L}}{\partial \mathbf{z}(T)}$ is:

$$\frac{d\mathbf{a}(t)}{dt} = -\mathbf{a}(t)^T \nabla_{\mathbf{z}(t)} f(\mathbf{z}(t), t; \theta), \tag{2.12}$$

and the loss function is calculated by maximizing the log-likelihood of the data, minimize $\mathcal{L} = -\log p(\mathbf{z}(T))$. Which is the negative of the log-density computed in the forward pass [15].

2.3.5 *Flow Matching for Generative Modeling*

Recently, a novel generative modeling approach called Flow Matching (FM) was introduced [29]. As a specialized extension of Continuous Normalizing Flows (CNFs), FM addresses the limitations of both traditional CNFs and Diffusion Models (discussed in Sections 3.2.4 and 3.2.3). The core idea of Flow Matching is to train CNFs by leveraging probability paths—smooth, parameterized transitions between distributions over time—to guide the transformation from a base distribution to a target distribution. These paths, which can be designed or selected based on computational efficiency or model requirements, serve as supervision for defining a clear trajectory that aligns the two distributions.

Flow Matching innovates on traditional CNFs by introducing probability paths as guidance, allowing transformations to focus on direct probability alignment without the iterative and noisy steps inherent in diffusion-based processes. This refinement retains the strengths of CNFs, such as invertibility and smooth transformations, while improving their scalability and computational efficiency.

As an advanced framework within the family of CNFs, Flow Matching significantly extends their applicability. It offers new opportunities for generative modeling by enabling the training of CNFs on larger scales with improved flexibility and performance. By combining the foundational principles of CNFs with these enhancements, FM represents a meaningful advancement in the field of generative modeling.

Mathematical Explanation of Flow Matching

Let x_0 be drawn from a simple distribution $p_0(x)$, such as the standard normal distribution $\mathcal{N}(0, I)$. The goal is to approximate a target distribution $q(x_1)$ by constructing a probability path $p_t(x)$ from $p_0(x)$ to $q(x_1)$. Define the path $p_t(x)$ such that:

$$p_t(x) \to q(x_1) \quad \text{as } t \to 1.$$

This path is guided by a vector field $u_t(x)$that directs the flow from $p_0(x)$ to $q(x_1)$. At time $t \in [0, 1]$, a velocity function $v : \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$ is set to drive the flow from p_0 to p_1,to control the rate of change of the distribution over time.

The Flow Matching objective minimizes the difference between the learned velocity field $v_t(x; \theta)$, parameterized by a neural network with parameters θ, and the true vector field $u_t(x)$ is approximated by $v_t(x; \theta)$. This objective is defined as:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t,p_t(x)}\left[\|v_t(x;\theta) - u_t(x)\|^2\right], \tag{2.13}$$

where the loss is minimized by sampling t from $U[0,1]$ and x from $p_t(x)$. When zero loss is reached, the learned **CNF!** model will generate samples matching the target distribution $q(x)$ through this transformation.

The evolution of the probability density is given by the continuity equation:

$$\frac{\partial p_t(x)}{\partial t} = -div \cdot (v(x,t)p_t(x)), \tag{2.14}$$

where div is the divergence operator. This relationship describes how the probability mass flows as it transforms over time.

Since there are no prior knowledges of p_t and u_t, direct computation of the marginal paths is often impossible. Conditional paths $p_t(x|x_1)$ and vector fields $u_t(x|x_1)$ provide a practical solution. These paths simplify sampling by enabling the calculation of probabilities and vector fields per data sample $x_1 \sim q(x_1)$ rather than over the entire distribution. This approach allows the model to operate on individual data samples, making the computation more efficient.

The conditional paths are defined such that $p_0(x|x_1) = p(x)$ at $t = 0$ and $p_1(x|x_1)$ is a Gaussian distribution centered around x_1:

$$p_t(x|x_1) = \mathcal{N}(x|\mu_t(x_1), \sigma_t(x_1)^2 I), \tag{2.15}$$

where the mean $\mu_t(x_1)$ and standard deviation $\sigma_t(x_1)$ are time-dependent functions that transition from p_0 to p_1. To ensures that the flow matches the distributional dynamics necessary to approximate $q(x)$.The vector field that determines these transitions is given by:

$$u_t(x|x_1) = \frac{\sigma_t'(x_1)}{\sigma_t(x_1)}(x - \mu_t(x_1)) + \mu_t'(x_1). \tag{2.16}$$

Consequently, $u_t(x|x_1)$ generates the Gaussian path $p_t(x|x_1)$.

To approximate the full marginal path $p_t(x)$, we aggregate the conditional paths over the distribution $q(x_1)$:

$$p_t(x) = \int p_t(x|x_1)q(x_1)\,dx_1. \tag{2.17}$$

This integration shows that marginal paths can be derived from conditional paths, highlighting why conditional paths are both practical and sufficient for generating the marginal distribution.

To simplify the training process, a Conditional Flow Matching (CFM) objective is defined that utilizes conditional paths:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,x_1\sim q(x_1),x\sim p_t(x|x_1)}\left[\|v_t(x;\theta) - u_t(x|x_1)\|^2\right] \tag{2.18}$$

This objective allows the generation of unbiased gradient estimates and the efficient training of the model, without the need for the full marginal paths or vector fields [29].

3

FRAMEWORK SELECTION FOR SYNTHETIC MEDICAL IMAGE GENERATION

The main objective of this work is to develop a machine learning model that can generate synthetic $3D$ medical images, in particular magnetic resonance imaging (MRI). The synthetic images generated by this model are meant to be used in other machine learning tasks like classification and segmentation, which require high-quality, anatomically accurate data. Generating a rich dataset of synthetic images can address the constraints of limited real-world medical data, allowing for more research while protecting patient privacy. To be sure that the generated images meet the necessary requirements, the model must provide high quality and diverse images that are consistent across $3D$ slices while keeping structural continuity. To identify the optimal generative model for this purpose, this thesis considers several well-known architectures, including Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), denoising diffusion models, normalizing flows (NF), and the recently proposed Flow Matching (FM) model. Each of these models has many strengths and limitations for generating realistic and diverse images, and their comparative advantages will be examined in the context of synthetic image generation spacially for medical images. Therefore, a detailed comparison of these methods will form the basis for selecting the most suitable model to achieve reliable and high quality synthetic medical imaging.

3.1 THE CHALLENGES OF SYNTHETIC MEDICAL IMAGING

Synthetic medical imaging has the potential to address the issue of data scarcity in medical research and deep learning. However, it is not without its own challenges. The quality of the input data is of the highest importance; nevertheless, factors such as patient movement, organ motion and scanner limitations can result in images of poor quality, which prevent the observation of essential details. Insufficient resolution or coverage also compromises the reliability of synthetic models, due to the resulting data incompleteness. The introduction of noise and artefacts, whether due to imaging equipment or patient-specific factors, serves to further complicate matters by preventing the accurate observation of true anatomical features. Moreover, inconsistencies in resolution and contrast between imaging slices or modalities create additional challenges for accurate reconstruction.

The computational demands of high-resolution 3D generation represent a further barrier, particularly for hospitals and other medical facilities with limited resources. The introduction of errors in the process of segmentation, which is the key for the definition of anatomical structures, can result in

the propagation of uncertainties throughout the synthetic model. Software limitations, including an absence tools for managing large datasets or modern modalities, further restrict generation quality. Finally, operator expertise is critical, as inexperience can result in suboptimal parameter settings and interpretations.

Overcoming these challenges requires advancements in imaging technology, robust computational algorithms, and effective operator training. By addressing these barriers, synthetic medical imaging can fulfill its potential to support diagnostic tools, data augmentation, and clinical decision-making [18, 47, 50]

3.2 COMPARISON OF SELECTED GENERATIVE MODELS

This section reviews several well-established generative models, comparing their strengths and weaknesses in the context of image synthesis to identify the most suitable approach. Table 3.1 provides a detailed comparison of the selected models, including GANs, VAEs, diffusion models, CNFs, and flow matching

3.2.1 *GANs*

GANs are well known for their ability to generate realistic, sharp and high quality images thanks to their adversarial training between a generator and a discriminator. In medical imaging applications, GANs offer the advantage of producing images with fine details, an essential quality in tasks like MRI reconstruction [21]. However, GANs suffer from many issues such as training instability and mode collapse, where the generator may limit itself to a reduced range of data features, which effects the output diversity. These issues make GANs challenging to optimize and may cause high computational costs. In addition, GANs are sensitive to hyperparameter settings, which has a slight impact on training stability and image quality [32]. One study investigated the effect of hyperparameter sensitivity on GANs. They analysed 1,500 hyperparameter searches on three medical imaging datasets using different GAN structures to demonstrate sensitivity. The results showed that only a few models produced meaningful images and even fewer models achieved reasonable metric scores [33].

3.2.2 *VAEs*

Variational Autoencoders (VAEs) are probabilistic generative models that map input data to a lower dimensional latent space and then decode from this space to reconstruct or generate new samples. These VAEs compress and create a latent representation that is ideal for learning tasks such as medical image synthesis, anomaly detection, and disease diagnosis, where it is important to allow for some variability in the produced data. Dimensionality reduction in the latent space allows for more efficient data manipulation

and analysis, which can simplify tasks such as interpolation, sampling and augmentation.

In medical imaging, where data is often high-dimensional and complex, the lower-dimensional latent space facilitates simpler computations, enhances interpretability, and improves computational efficiency. VAEs optimize the ELBO to approximate the data's log-likelihood, ensuring the model learns the real image distribution. The ELBO is composed of a reconstruction loss, which measures how accurately the input is recreated, and a KL divergence term, which structures the latent space with a Gaussian prior. Balancing these terms allows VAEs to learn a detailed latent space and produce a variety of realistic samples.

However, VAEs face significant challenges, particularly in medical imaging. The reliance on L2 norm-based reconstruction loss can result in blurred images, as the averaging process during reconstruction often reduces the fidelity of fine details. This limitation affects applications requiring high-resolution or highly detailed images, such as diagnostic imaging. While synthetic data is generally not used directly in diagnostics due to strict accuracy and reliability requirements, it serves as a crucial auxiliary tool. Synthetic data is widely employed for tasks like data augmentation, model validation, testing edge cases, and training deep learning models, particularly in scenarios where real-world data is scarce or imbalanced. Improving the quality of synthetic images generated by VAEs can enhance these supporting roles, indirectly contributing to advancements in diagnostic tools.

Furthermore, the Gaussian assumption on the latent space might fail to sufficiently represent the complexity of real-world data, limiting their ability to effectively simulate complicated medical image structures. Balancing the reconstruction and KL divergence terms also presents practical challenges. If the KL divergence term is overly weighted, the model may experience posterior collapse, in which the latent space fails to capture meaningful information about the input, reducing the quality and specificity of generated samples[24, 28, 44].

3.2.3 *Denoising Diffusion models*

Diffusion models are among the most well-known generative models, characterized by several advantages. These include the ability to generate high-quality, realistic, and diverse images, as well as their effectiveness in handling fine details within the generated images.

These strengths arise from their iterative mechanism. During the training process, diffusion models gradually add noise to an image, transforming it into a fully noisy representation with a Gaussian-distributed density. The models then iteratively refine this noisy image during the generation process until a realistic output is produced. This iterative approach enhances the performance of diffusion models, often resulting in more accurate and stable outputs compared to GANs, while also reducing the risk of mode collapse. These capabilities make diffusion models particularly well-suited for

generating detailed images, which are crucial for accurate analysis and diagnosis in medical imaging [8].

Diffusion models have limitations such as high computational complexity and long training times due to their iterative structure. The inference process is also sometimes slower compared to GANs and VAEs for generating images because it involves many backward steps to generate an image. One of the most important things that diffusion models require to train effectively is large datasets, which can be challenging in areas such as high-resolution images with limited data. Although they tend to avoid mode collapse better than GANs, producing different outputs can still be challenging. In addition, training diffusion models can be complex and requires substantial computational power and memory, which might be challenging for some medical institutions without specialized resources[8, 19, 48, 51]

3.2.4 *Continuous Normalizing Flows*

Continuous Normalizing Flows (CNFs) focus on exact likelihood-based density estimation. They model a target distribution by continuously transforming a simple base distribution (like Gaussian) using a time-dependent differential equation. By optimizing the exact log-likelihood, CNFs aim to learn smooth, invertible mappings, leveraging the change of variables formula and differential equation solvers [5].

To minimize the negative log-likelihood, CNFs rely on computing the trace of the Jacobian, a process that is both computationally demanding and sensitive to numerical solvers. For example, maximum likelihood training in frameworks such as FFJORD requires the solution of costly numerical ODE simulations. These simulations become increasingly prohibitive for high-dimensional data, such as images, due to the computational cost and the need for numerical stability during continuous transformations [16].

Attempts to avoid these expensive computations include simulation-free approaches. However, these alternatives introduce new issues. Rozen et al. [43] identified challenges with intractable integrals in some methods, while Ben-Hamu et al.[3] demonstrated that other techniques result in biased gradient estimates.

In addition to computational intensity, CNFs are highly sensitive to the choice of differential equation solvers, further complicating training. Notably, no scalable training algorithms for CNFs are currently known, posing a significant barrier to their adoption for high-dimensional datasets [29]

3.2.5 *Flow Matching*

Flow Matching is introduced as a new method for training CNFs without depending on costly simulations typically used in traditional CNF training (such as in FFJORD or similar methods). The key idea of FM is to regress the vector fields of fixed conditional probability paths instead of solving the usual differential equations over time, leading to more efficient train-

ing processes. This simulation-free nature allows CNF training to be scaled to larger datasets and models. Furthermore, FM is compatible with a variety of Gaussian probability paths, including those used in diffusion models, but also opens the door to using alternative paths, such as Optimal Transport (OT), which leads to even faster training and improved generalization. These advances make Flow Matching particularly useful in tasks that require both high computational efficiency and high-quality generative performance, such as large-scale image generation. With FM, CNFs training is more accessible, faster, and provides higher-quality results, surpassing the performance of traditional diffusion-based models on datasets like ImageNet.

One of the main limitations of flow matching models is their relative newness in the field of generative modeling. While they have shown promise in generating 2D images, their application to higher-dimensional settings, such as 3D natural images, remains largely unexplored. Furthermore, no studies have yet applied flow matching to 2D medical images. This lack of dedicated research on medical image generation means that the potential and challenges of applying flow matching in these domains are not fully understood. This lack of specific research on medical image generation means the potential and challenges of using flow matching in these domains are not fully understood, highlighting the need for further studies to explore its suitability for this application.

3.3 CHOOSING THE DESIRED MODEL

The generation of high-quality, realistic three-dimensional medical images, such as magnetic resonance imaging (MRI), remains a challenging task due to the need for high-resolution details and slice consistency. Developing a framework that leverages the strengths of established generative models while minimizing their limitations is essential. To address this, this thesis proposes a hybrid approach that integrates VQ-GAN and flow matching in the latent space.

VQ-GAN, or Vector Quantized Generative Adversarial Network (briefly explained in section 3.4), is a generative model that combines the strengths of GANs and VAEs. It uses adversarial training to ensure high-quality and realistic image generation while leveraging a vector-quantized latent space, similar to VAEs, for efficient data representation and manipulation. This latent space facilitates structured image generation and controlled modifications, making VQ-GAN particularly suitable for tasks like generating detailed medical images.

The decision to use VQ-GAN stems from its ability to combine the benefits of adversarial training and latent space representation. The adversarial component of VQ-GAN ensures that the generator produces realistic, high-resolution images, refined by the feedback from the discriminator. Additionally, the vector-quantized latent space makes it easier to represent and manipulate data efficiently, a critical feature for generating medical image

variations with specific properties. These capabilities make VQ-GAN an ideal choice for high-quality image synthesis [10, 23].

While VQ-GAN ensures efficient latent space representation and high-quality image generation, flow matching complements it by providing a powerful method for learning complex data distributions within this latent space [23]. Flow matching builds on the principles of continuous normalizing flows (CNFs) but avoids the computational overhead associated with calculating exact likelihoods. Instead, it approximates the data distribution, bypassing the need for the exact likelihood computations required by traditional CNF models.

Recent studies have demonstrated that flow matching outperforms traditional diffusion models in generating high-quality two-dimensional images and video predictions. Its ability to model complex distributions without relying on exact likelihoods makes it a promising choice for generative tasks. By combining the strengths of VQ-GAN and flow matching, the proposed framework aims to tackle the challenges associated with generating high-quality 3D medical images, offering a novel and efficient solution to this complex problem [29].

3.4 VECTOR QUANTIZED GENERATIVE ADVERSARIAL NETWORK (VQ-GAN)

The Vector Quantized Generative Adversarial Network (VQ-GAN) proposed by Esser et al. [10] is a generative model that combines ideas from both Vector Quantized Variational Autoencoder (VQ-VAE) (Vector Quantized Variational Autoencoder) [37] and GANs (Generative Adversarial Networks). The VQ-GAN model is an extension of the VQ-VAE [37] framework, improving it with a transformer architecture and simply integrating an adversarial (GAN) loss from a discriminator to further improve the quality of the reconstructed images.

Its main goal is to produce high-quality, detailed images while reducing computational cost and improving memory efficiency by operating in a latent space, which is typically useful in scenarios requiring high-resolution images with a high degree of detail, such as medical imaging. It consists of four sub-models: an encoder, a decoder, the codebook with transformer architecture and the discriminator.

- **Encoder**: the encoder E is a convolutional network, which consists of convolutional layers that downsample the input image while capturing essential image features. It produces a feature map with lower spatial resolution, which is then mapped to the codebook during quantization.

- **Codebook**: The codebook Z is similar to a dictionary that contains a collection of learned vector representations (or "codes") that represent a unique feature or pattern learned directly from the data, acting as discrete latent variables. In simple terms, the codebook represents a compressed and quantized version of an image, where each image (or part of an image) is represented by one of these codebook entries. After

the image has been encoded into a latent space, each feature vector in the latent representation is quantized. This means that it is replaced by the nearest vector in the codebook, forming a discretized latent representation.

The VQ-GAN uses a transformer architecture to generate images based on the quantized latent codes. The core idea is to handle the quantized latent codes as a sequence to model the dependencies between different parts of the image, and to learn the distribution of these indices in the form of a sequence. The transformer then predicts the next index in the sequence based on the previous indices, learning an autoregressive model for image generation.

- **Decoder**: The decoder G is a convolutional network, using upsampling layers to reconstruct the image from the quantized feature map. The architecture is designed to match the feature scale of the encoder so that the high-level features can be effectively decoded back into a high-resolution image.

- **Discriminator**: By adding a discriminator to the architecture, VQ-GAN uses adversarial training to improve the perceptual quality of the resulting images. This is another neural network trained to discriminate between real and generated images. The discriminator operates on patches of the image rather than the whole image, using a convolutional "PatchGAN" classifier from [22] that only penalizes structures at the scale of image patches. This helps the model to focus on local image details (texture, edges, etc.), resulting in finer, more realistic image generation.

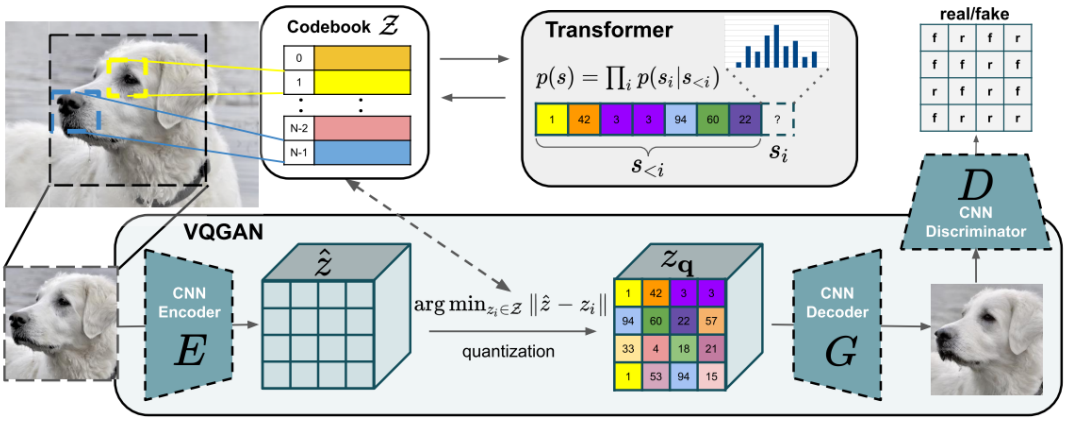

Figure 3.1: VQ-GAN architecture: The model uses an encoder E to generate latent representations, which are quantized via a codebook Z. A transformer learns dependencies between quantized codes for autoregressive image generation. The decoder G reconstructs the image, and a discriminator D enhances image quality via adversarial training. The model is optimized with a combined vector quantization, adversarial, and autoregressive loss [10].

3.4.1 *Structure of the VQ-GAN*

The VQ-GAN structure follows an encoding, quantization, and decoding pipeline with a focus on reconstructing input images x in latent space. The encoder E maps the input x into a latent representation $\hat{z} = E(x)$, which has spatial dimensions $h \times w$ and depth n_z: $\hat{z} = E(x) \in \mathbb{R}^{h \times w \times n_z}$

Quantization Process

Each spatial position $\hat{z}_{ij}$ in the encoder output $\hat{z}$ is quantized by mapping it to the nearest vector z_k in a discrete codebook Z. The quantized output z_q is obtained as follows:

$$z_q = q(\hat{z}) := \arg\min_{z_k \in Z} \|\hat{z}_{ij} - z_k\| \tag{3.1}$$

where $Z = \{z_k\}_{k=1}^{K}$ is the set of learned codebook vectors, and K denotes the number of entries in this codebook. This quantization step provides a discrete representation of the latent image structure, preserving essential features at reduced computational complexity.

Decoding and Image Reconstruction

To reconstruct the image from the quantized latent representation z_q, the decoder G is applied:

$$\hat{x} = G(z_q) = G(q(E(x))) \tag{3.2}$$

where $\hat{x}$ denotes the reconstructed image. This process approximates the original image x as closely as possible within the constraints of the learned latent space.

Backpropagation through Non-differentiable Quantization

The quantization operation $q(\cdot)$ is non-differentiable. To facilitate gradient backpropagation, a straight-through estimator is used, allowing gradients from the decoder G to flow through $q(\cdot)$ to the encoder E. This enables end-to-end training of the VQ-GAN model, including updates to the encoder, decoder, and codebook entries.

Objective Function

The training of VQ-GAN is driven by a mixed objective function that includes the following loss terms:

- **Reconstruction Loss** $L_{\mathbf{rec}}$: This measures the difference between the original image x and its reconstruction $\hat{x}$ using an L_2-norm loss:

$$L_{\text{rec}} = \|x - \hat{x}\|_2^2 \tag{3.3}$$

- **Codebook Commitment Loss**: This term encourages the encoder output $E(x)$ to stay close to the selected codebook entries z_q, ensuring stability in representation and promoting an efficient, compact encoding. It is composed of two sub-terms:
  - *Codebook Loss*: Prevents the codebook vectors from deviating too far from the latent representations, defined as:

$$\|\text{sg}[E(x)] - z_q\|_2^2 \tag{3.4}$$

  - *Commitment Loss*: Keeps the encoder outputs close to the quantized vectors, defined as:

$$\|\text{sg}[z_q] - E(x)\|_2^2 \tag{3.5}$$

Here, $\text{sg}[\cdot]$ represents the stop-gradient operation, which prevents gradients from updating the variables inside it. This separation ensures both the encoder and codebook are optimized effectively.

The Reconstruction Loss L_{rec} with Codebook Commitment Loss together give the Vector Quantization loss $L_{\text{VQ}}(E, G, Z)$ as follows:

$$L_{\text{VQ}}(E, G, Z) = L_{\text{rec}} + \|\text{sg}[E(x)] - z_q\|_2^2 + \|\text{sg}[z_q] - E(x)\|_2^2 \quad (3.6)$$

- **Adversarial Loss L_{GAN}**: Improves the perceptual quality of the images by using a discriminator D:

$$L_{\text{GAN}} = \log D(x) + \log(1 - D(\hat{x})) \quad (3.7)$$

- **Total Objective Function**: Combining the reconstruction, commitment, and adversarial terms, the full objective function for VQ-GAN can be expressed as:

$$Q^* = \arg \min_{E,G,Z} \max_D \mathbb{E}_{x \sim p(x)} \left[L_{\text{VQ}}(E, G, Z) + \lambda L_{\text{GAN}}(\{E, G, Z\}, D) \right] \quad (3.8)$$

where λ is an adaptive weight controlling the influence of the adversarial component on the model's learning process. This weight is computed to balance the gradients between the reconstruction and GAN terms:

$$\lambda = \frac{\|\nabla_G[L_{\text{rec}}]\|}{\|\nabla_G[L_{\text{GAN}}]\| + \delta} \quad (3.9)$$

where $\delta = 10^{-6}$ is added for numerical stability.

Transformer-based Sequence Modeling

With E and G in place, images can be represented as sequences of codebook indices $s \in \{0, \ldots, |Z| - 1\}^{h \times w}$, where each s_{ij} specifies a codebook entry for a given spatial position (i, j) in the quantized latent space z_q. An ordering is chosen for these indices, and a transformer is trained to predict each subsequent index in the sequence, optimizing for the log-likelihood of the entire sequence:

$$L_{\text{Transformer}} = \mathbb{E}_{x \sim p(x)} \left[-\log p(s) \right] \quad (3.10)$$

This enables autoregressive generation of images, with each index prediction conditioned on prior ones, thereby modeling spatial dependencies effectively.

3.5 FLOW MATCHING MODEL

The flow matching framework aims to model the continuous transformation between an initial data distribution p_0 and a target noise distribution p_1 (often Gaussian). Given empirical observations $x_0 \sim p_0$ and $x_1 \sim p_1$, the objective is to estimate a coupling $\pi(p_0, p_1)$ that describes the evolution between these two distributions. This process is formulated as solving an ordinary differential equation (ODE):

$$\frac{dx_t}{dt} = v(x_t, t), \tag{3.11}$$

where $t \in [0,1]$ and $v : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ is a velocity function guiding the flow from p_0 to p_1.

To train the flow matching network on a latent space, this work draws inspiration from prior frameworks such as the one presented in [6]. In this framework, flow matching was trained on a pretrained latent space generated using a Variational Autoencoder (VAEs)[25] from Stable Diffusion [42] to produce $2D$ synthetic images. Furthermore, methodologies explored in [7] demonstrated the effectiveness of pairing VQ-GAN with flow matching to train latent flow models for video prediction tasks, generating temporally coherent sequences. This approach serves as a foundation for developing methods suited to $3D$ image generation in the medical domain (more details come in the next chapter).

For this specific application, MRI images are encoded into a latent space using VQ-GAN, followed by training a flow matching model to generate synthetic $3D$ images.

3.5.1 *Flow Dynamics and Loss Optimization*

The velocity is parameterized by $v_\theta(x_t, t)$, and the parameters θ are optimized through a least-squares regression problem:

$$\hat{\theta} = \arg\min_\theta \mathbb{E}_{t,x_t}\left[\|v(x_t,t) - v_\theta(x_t,t)\|_2^2\right], \tag{3.12}$$

where the expectation is taken over the empirical path. This approach enables flexible and efficient learning of the flow dynamics, and backward sampling is achieved by integrating from x_1 to x_0 using numerical integration methods.

$$x_0 = x_1 - \int_0^1 v(x_t, t)\, dt. \tag{3.13}$$

The ODE in 3.11, known as the Lagrangian flow, describes the continuous transformation of point clouds. An alternative perspective is provided by

the Eulerian form, where a continuity equation characterizes the change in distribution over time [6]:

$$\frac{\partial p_t}{\partial t} = -\nabla \cdot (v(x,t)p_t). \tag{3.14}$$

Two main variations of $v(x,t)$ are commonly employed in the flow matching framework [46] and the constant velocity ODE [31]. As it is shown in [31] the nonlinear interpolation in the Variance Preserving (Variance Preserving (VP)) path, which is common choice to define the pathx_t between x_1 and x_0 in the probability flow ODE method, may introduce unnecessary curvature in the generative trajectories, negatively impacting training efficiency. The constant velocity ODE mitigates this by using a linear interpolation path between x_1 and x_0 such that:

$$x_t = (1-t)x_0 + tx_1. \tag{3.15}$$

The velocity is then given by $v_t = x_1 - x_0$, and the flow matching loss is formulated by:

$$\hat{\theta} = \arg\min_{\theta} \mathbb{E}_{t,x_t}\left[\|x_1 - x_0 - v_\theta(x_t,t)\|_2^2\right]. \tag{3.16}$$

For this application, the constant velocity ODE is employed as it ensures a smooth linear transformation and enhances sampling efficiency, making it particularly suitable for training on $3D$ MRI. However, the choice between the probability flow ODE and constant velocity ODE is not fixed; it depends on the specific requirements of the task. If capturing more complex data structures is necessary, the probability flow ODE may still be used, provided its trade-offs are acceptable.

Table 3.1: Comparison of Selected Generative Models for Medical Imaging

Model	Strengths	Weaknesses	Suitability for Medical Imaging
GANs	- Generates sharp, high-quality images. - Well-suited for fine details (e.g., MRI reconstruction).	- Training instability and mode collapse. - Sensitive to hyperparameter tuning. - High computational costs.	Effective for high-quality tasks like MRI reconstruction but requires careful optimization.
VAEs	- Probabilistic nature enables variability in generated data. - Latent space enables efficient manipulation and interpolation.	- L2 loss can cause blurry reconstructions. - Gaussian latent assumption may not capture complex structures. - Posterior collapse risk.	Useful for tasks requiring variability but struggles with fine detail preservation in outputs.
Diffusion Models	- Generates high-quality, diverse, and detailed images. - Less prone to mode collapse compared to GANs.	- Computationally intensive and slow inference. - Requires large datasets. - Training complexity and resource-heavy.	Suitable for generating detailed, stable images but limited by computational demands and dataset size.
CNFs	- Exact likelihood estimation. - Smooth, invertible transformations.	- Computationally expensive (ODEsolvers, Jacobian calculations). - No scalable training algorithms for high dimensions.	Limited by computational cost, making them less practical for large-scale medical imaging tasks.
Flow Matching	- Simulation-free training for CNFs. - Scales efficiently to larger datasets and models. - Compatible with diffusion paths and alternatives (e.g., Optimal Transport).	- Limited exploration in 3D or medical images. - Insufficient studies on applicability to medical imaging.	Promising for computational efficiency and high-quality results but requires further research.

4

DATA AND DEVELOPED FRAMEWORK

This chapter provides an overview of the data used for training the proposed framework and outlines the structure of the model architecture, as well as the implementation and training procedures. The focus is on leveraging a two-stage model for generating 3D medical images, specifically for the generation of synthetic MRI scans from Alzheimer's patients. The first stage utilizes a Variational Autoencoder-based Generative Adversarial Network (VQ-GAN) to learn a compact latent representation of the medical images. The second stage integrates a flow matching model, which operates in the continuous latent space and generates new images by learning continuous transformations.

4.1 MODEL ARCHITECTURE

The framework consists of a two-stage model as shown in Figure 4.1:

- **First Stage:** The first stage involves training the VQ-GAN. This stage focuses on learning to encode input images into a discrete latent space using a convolutional encoder and to reconstruct images from this latent space using a convolutional decoder. The architecture of the convolutional encoder and decoder models used in the VQ-GAN experiments is detailed in Table 4.1 and 4.2 .

- **Second Stage: Latent Flow Matching Model** In the second stage, the pre-trained VQ-GAN encoder is utilized to encode input images into the latent space, and its decoder reconstructs outputs from this space. Within this latent space, a flow matching model operates to generate new data by learning continuous transformations. A 3D UNet architecture is employed for the flow matching model due to its widespread success in medical imaging tasks and diffusion models.

  To adapt to the volumetric nature of 3D medical images, the UNet architecture replaces 2D operations with 3D convolutions, ensuring volumetric consistency. The architecture includes three main stages, detailed as follows:

  1. **Downsampling:** The model begins with a series of 3D convolutional layers that reduce the spatial dimensions of the input image while extracting relevant features. Attention blocks are integrated at certain levels to help the model focus on key areas of the image. Residual connections and time embeddings are also employed to enhance feature learning and represent temporal dynamics effectively. Dropout (0.5) is applied to prevent overfitting.

2. **Bottleneck:** The bottleneck stage processes the downsampled features using a combination of residual blocks and attention layers. This stage captures complex patterns and relationships in the latent space, enabling the model to better understand the input image.
3. **Upsampling:** Transposed convolutions are used to progressively restore the resolution of the image. Skip connections are incorporated from the downsampling path to combine fine-grained details with the upscaled features, ensuring high-resolution and accurate reconstructions. The final output is produced by applying normalization and a non-linear activation function to the upscaled features.

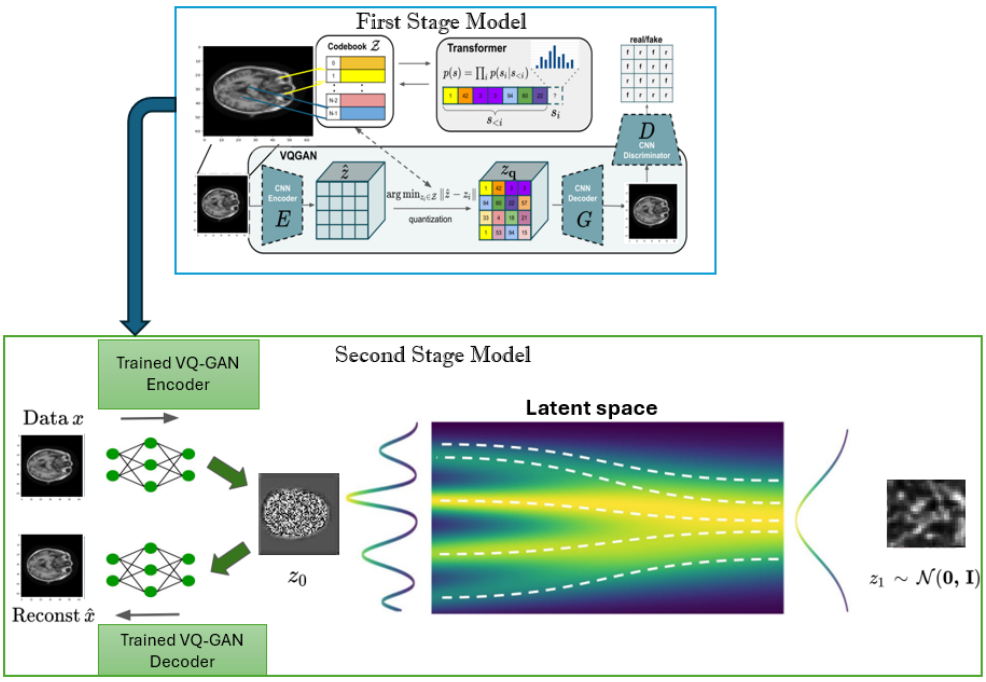

Figure 4.1: Two-Stage Model Architecture for 3D Medical Image Generation. The framework consists of two stages. In the first stage, a VQ-GAN model is trained to encode input images into a discrete latent space ($\hat{z}$) using a CNN encoder (E) and a codebook ($\mathcal{Z}$), then reconstruct the images via a CNN decoder (G), then an adversarial training procedure and a patch-based discriminator (D) applied to differentiate between real and reconstructed images. The trained VQ-GAN model is then used in the second stage. In the second stage, a Latent Flow Matching framework is applied, where the input data x is encoded with the pre-trained encoder of the first stage model to produce the latent representation z_0. The latent flow network predicts the velocity of the transformation from a standard normal distribution $p(z_1) = \mathcal{N}(0, I)$ to the target latent distribution $p(z_0)$. During sampling, random noise z_1 is drawn from $p(z_1)$, and the network predicts the velocity towards $p(z_0)$ via numerical integration. Finally, z_0 is decoded with the VQ-GAN decoder from the first stage to generate the image.

Layer	Output Shape
Input x	$\mathbb{R}^{H \times W \times C}$
Conv3D Layer (conv_first)	$\mathbb{R}^{H \times W \times n\_hiddens}$
Downsample Block + Residual Block	$\mathbb{R}^{H/s \times W/s \times 2^i \times n\_hiddens}$ (i steps, repeated based on the downsampling)
Residual Block	$\mathbb{R}^{H/s \times W/s \times 2^{i+1} \times n\_hiddens}$
Final Block (Normalize + SiLU)	$\mathbb{R}^{H/s \times W/s \times 2^{max\_ds} \times n\_hiddens}$
Output	$\mathbb{R}^{H/s \times W/s \times 2^{max\_ds} \times n\_hiddens}$

Table 4.1: Encoder Architecture

Layer	Output Shape
Input x	$\mathbb{R}^{H/s \times W/s \times 2^{max\_us} \times n\_hiddens}$
Final Block (Normalize + SiLU)	$\mathbb{R}^{H/s \times W/s \times 2^{max\_us} \times n\_hiddens}$
UpSample Block + Residual Block	$\mathbb{R}^{H/s \times W/s \times 2^{max\_us-i} \times n\_hiddens}$ (Repeated for i steps)
ConvTranspose3D (Upsampling)	$\mathbb{R}^{H/s \times W/s \times 2^{max\_us-i+1} \times n\_hiddens}$
Residual Block 1 (ResBlock)	$\mathbb{R}^{H/s \times W/s \times 2^{max\_us-i+1} \times n\_hiddens}$
Residual Block 2 (ResBlock)	$\mathbb{R}^{H/s \times W/s \times 2^{max\_us-i+1} \times n\_hiddens}$
Conv3D (conv_last)	$\mathbb{R}^{H \times W \times C}$

Table 4.2: Decoder Architecture

4.2 IMPLEMENTATION AND TRAINING

This section outlines the implementation details for adapting the pre-existing VQ-GAN framework for 3D medical images (as demonstrated in [23]) to work with the provided dataset and computational constraints, and its integration with the flow matching model (performed in this work) for generating synthetic 3D MRI images.

4.2.1 *Data Preparation*

The Alzheimer's Disease Neuroimaging Initiative (ADNI) [38] dataset contains brain MRI scans from n=2733 patients (used just 320 randomly images for training the model) . The ADNI was launched in 2003 as a publicprivate partnership, led by Principal Investigator Michael W. Weiner, MD. The primary goal of ADNI has been to test whether serial magnetic resonance imaging (MRI), positron emission tomography (PET), other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early Alzheimer's

disease (AD).

MRI images of Alzheimer's patients exhibit significant variability depending on the progression of the disease. As Alzheimer's progresses, observable changes in the brain through MRI scans include "atrophy," or the shrinkage of brain regions such as the hippocampus and cortical areas, which are critical for memory and cognitive function. In early stages, these changes may be subtle, while advanced stages show more pronounced atrophy.

The decision to use this dataset was driven by its variability, offering a distinct advantage for training models. The diversity of brain structures across different stages of Alzheimer's provides the model with a wide range of cases, enabling it to generate distinct brain images that reflect various stages of the disease. The dataset was carefully examined and labeled by a medical expert to ensure accurate categorization and to enhance understanding of brain visualization across the disease's stages (Figure 4.2).

To ensure all 3D MRI images are compatible for model training, a preprocessing pipeline is employed. Images are resampled to a target voxel spacing of $(1.0, 1.0, 1.0)$ using SimpleITK to maintain spatial consistency. Voxel intensity values are normalized to the range $[0, 1]$, and non-brain areas are cropped to focus on relevant anatomical structures, with images standardized to cubic dimensions using (TorchIO's `CropOrPad`) function. Random augmentations, including intensity scaling and flipping, are applied to improve model generalization. Finally, images are resized to $(64, 64, 64)$ to match the model input size, and the dataset is split into 80% for training and 20% for validation.

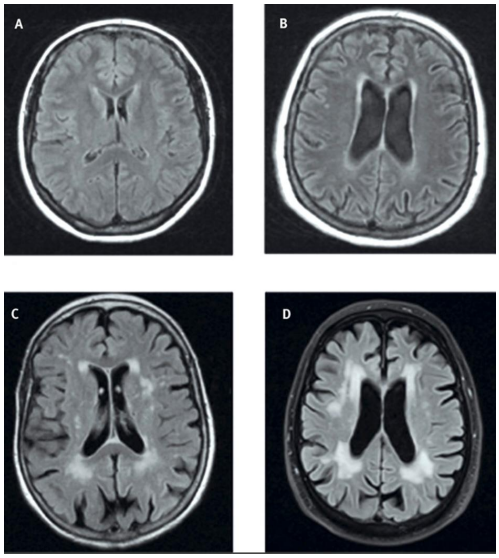

Figure 4.2: MRI Brain Scans Demonstrating Stages of Alzheimer's Disease Progression. (A) Normal brain structure; (B) Early-stage Alzheimer's with mild atrophy; (C) Moderate-stage Alzheimer's showing significant atrophy and enlargement of ventricles; (D) Advanced-stage Alzheimer's with pronounced brain shrinkage and severe ventricular enlargement.

4.2.2 *VQ-GAN Implementation*

In this work, the VQ-GAN framework proposed by [10], which was already adapted to support 3*D* MRI data by [23], was fine-tuned for the specific task of generating synthetic 3*D* MRI images. The model architecture, as modified by [23], includes significant enhancements to handle volumetric data effectively. These modifications, utilized in this work, include:

- 3*D* **Convolutions:** Replacing all 2*D* convolutions with 3*D* convolutions to enable effective processing of volumetric data.
- **Discriminators:** Incorporating a slice-wise discriminator and a 3*D* discriminator. The slice-wise discriminator evaluates individual slices of the image volume, while the 3*D* discriminator assesses the reconstructed volumes as a whole, thereby improving reconstruction quality.
- **Feature Matching Losses:** Adding feature matching losses to stabilize adversarial training by aligning the feature representations between the generator and the discriminator.

The pre-adapted VQ-GAN model from [23] was fine-tuned to ensure compatibility with the Alzheimer's dataset and address specific computational constraints, such as GPU memory limitations.

Training Setup

The training setup for the VQ-GAN included the following hyperparameters:

- **Latent Embedding Dimension:** Set to 8 to align with the requirements of the flow matching model.
- **Compression Factor:** Carefully selected to balance dimensionality reduction with reconstruction quality.
- **Losses:**
  - *Perceptual Loss:* Used as the primary reconstruction loss to enhance perceptual quality.
  - *Codebook and Commitment Losses:* Enforced quantization quality by penalizing deviations of the encoded latent vectors from the closest codebook entries.
- **Learning Rate:** Set to 3×10^{-4}, optimized based on initial experimentation.

The straight-through estimator approach was used to address the non-differentiability of the quantization step.

Fine-Tuning Configurations

The VQ-GAN model was fine-tuned using the following configurations:

- **Codebook Size:** Configured to contain 12,288 learnable vectors, providing sufficient expressivity for latent representations.
- **Number of Hidden Units:** Set to 16, balancing model capacity and computational efficiency.
- **Downsampling Rates:** Initially experimented with $[2, 2, 2]$ and $[4, 4, 4]$ to evaluate their impact on model performance. Due to GPU memory limitations, the final training was conducted using $[4, 4, 4]$.
- **Gradient Clipping:** Applied with a value of 1.0 to prevent exploding gradients.
- **GPU Utilization and Gradient Accumulation:** Training was performed on a multi-GPU setup with 4 GPUs. Given the high memory requirements of $3D$ medical images, the batch size was set to 2 per GPU. Gradient accumulation was set to 2, meaning the model updated the weights after accumulating gradients over two mini-batches. With a batch size of 2 per GPU and gradient accumulation set to 2, the effective batch size across all GPUs was 16. This setup allowed for more efficient memory usage while maintaining the same update frequency as if using a batch size of 8.

The detailed training procedure for the VQ-GAN model, including the computation of various loss terms and optimization steps, is outlined in Algorithm 1.

4.2.3 *Flow Matching Model Implementation*

Once the VQ-GAN model was trained, the pretrained VQ-GAN encoder was used to encode the MRI input images x into continuous latent vectors z_0. Since the flow matching model operates in continuous latent space, the vector quantization step was deactivated during the encoding process to ensure the latent vectors remained continuous. This adjustment enables smooth dynamics during the training of the flow matching model in latent space, ensuring compatibility between the VQ-GAN encoder and the flow matching framework.

The flow matching model was trained to predict the velocity field that controls the transformation between latent variables z_0 (produced by the VQ-GAN encoder) and z_1 (sampled as random noise).

The training setup for the flow matching included the following hyperparameters:

- **Batch Size and GPUs:** to address memory constraints during the training of the flow matching model, a batch size of 2 per GPU was used.

This configuration ensured that individual GPUs could handle the memory requirements for processing $3D$ medical image data. When multiple GPUs were available, parallelization was enabled using the PyTorch DistributedDataParallel framework. The batch size per GPU was calculated dynamically as follows:

$$\text{batch\_size\_per\_gpu} = \frac{\text{batch\_size}}{\text{total-gpu}}$$

where total-gpu determines the number of GPUs available. Each GPU was configured to handle two patches of the input data, optimizing both memory usage and computational efficiency.

- **Exponential Moving Average (Exponential Moving Average (EMA)):** An EMA, as used in [6] was employed to stabilize training by averaging the parameters of the flow matching model over time. This approach reduces noise from parameter updates and ensures smoother convergence. The EMA was implemented with a decay rate of $\alpha = 0.995$, as follows:

  $$\theta_{\text{EMA}} \leftarrow \alpha \cdot \theta_{\text{EMA}} + (1 - \alpha) \cdot \theta_{\text{current}},$$

  where θ_{EMA} are the EMA parameters, and θ_{current} are the current model parameters. The EMA parameters were periodically used for evaluation and saved as part of the checkpointing process. This ensured that the models performance was evaluated using stabilized parameters, minimizing fluctuations due to rapid updates.

- **Learning Rate:** Set to 2×10^{-5}.

- **ODE Solver:** An ordinary differential equation (ODE) solver was used to integrate the predicted velocities over time, evolving latent variables z_t from z_0 to z_1.

- **Gradient Clipping:** Applied with a threshold of 1.0 to stabilize training.

- **Velocity Prediction Loss:** The loss penalizes deviations between the predicted velocity $v(t, z_t)$ and the true velocity $u = z_1 - z_0$

  $$\mathcal{L}_v = \|v(t, z_t) - u\|^2$$

The step-by-step procedure for training the flow matching model in the latent space is detailed in Algorithm 2.

Algorithmus 1 : Training the VQ-GAN

Input : Dataset $\mathcal{D}$, encoder E_θ, decoder D_ϕ, codebook $\mathcal{C}$, 2D discriminator D_{2D}, 3D discriminator D_{3D}, learning rate η_{VQ}, hyperparameters $\lambda_{\text{recon}}, \lambda_{\text{commit}}, \lambda_{\text{perc}}, \lambda_{\text{gan}}$.

repeat

Sample mini-batch $\{x_i\}_{i=1}^{B} \sim \mathcal{D}$;

Encode latent features: $z_0 \leftarrow E_\theta(x)$;

Quantize latent features: $z_{\text{quant}} \leftarrow \text{Quantize}(z_0, \mathcal{C})$;

Reconstruct images: $x_{\text{recon}} \leftarrow D_\phi(z_{\text{quant}})$;

Compute losses:

$$L_{\text{recon}} = \lambda_{\text{recon}} \|x - x_{\text{recon}}\|_1$$

$$L_{\text{commit}} = \lambda_{\text{commit}} \cdot \text{CommitLoss}(z_0, z_{\text{quant}})$$

Select random slice x_{slice} and $\hat{x}_{\text{slice}}$ for perceptual and adversarial loss:

$$L_{\text{perc}} = \lambda_{\text{perc}} \cdot \text{LPIPS}(x_{\text{slice}}, \hat{x}_{\text{slice}})$$

$$L_{\text{gan}} = \lambda_{\text{gan}} \big(L_{\text{adv-2D}}(x_{\text{slice}}, \hat{x}_{\text{slice}}) + L_{\text{adv-3D}}(x, \hat{x}) \big)$$

Combine total loss:

$$L = L_{\text{recon}} + L_{\text{commit}} + L_{\text{perc}} + L_{\text{gan}}$$

Update encoder E_θ, decoder D_ϕ, codebook $\mathcal{C}$, and discriminators D_{2D}, D_{3D} using optimizer with learning rate η_{VQ};

until *convergence*;

Algorithmus 2 : Training Flow Matching in Latent Space

Input : Normalized latent data $Z_{\text{normalized}}$, trained VQ-GAN Encoder E, velocity estimator v_θ, learning rate η_{FM}

repeat

Sample an MRI image x from the dataset D

Encode it with pre-trained VQ-GAN encoder E to obtain latent representation: $z_0 = \text{Encoder}(x_0)$;

Sample noise $z_1 \sim \mathcal{N}(0, I)$

Sample time $t \sim \text{Uniform}(0, 1)$

Interpolate $z_t \leftarrow (1-t)z_0 + tz_1$

Compute target velocity $u(z_t) \leftarrow z_1 - z_0$

Compute loss:

$$\ell \leftarrow \|v_\theta(z_t, t) - u(z_t)\|^2$$

Update $\theta \leftarrow \theta - \eta \nabla_\theta \ell$

until *convergence*;

Algorithmus 3 : Sampling with Flow Matching

Input : Trained velocity estimator v_θ, pre-trained VQ-GAN decoder D, number of time steps N, initial noise $z_1 \sim \mathcal{N}(0, I)$

for $n = N - 1$ ***to*** 0 **do**

Compute $t_n \leftarrow \frac{n}{N}$;

Compute $t_{n+1} \leftarrow \frac{n+1}{N}$;

Update $z_{t_n} \leftarrow z_{t_{n+1}} + (t_n - t_{n+1}) \cdot v_\theta(z_{t_{n+1}}, t_{n+1})$;

Reconstruct with pre-trained VQ-GAN decoder D ($x_0 \leftarrow D(z_0)$);

return *Generated MRI sample* x_0

5 EXPERIMENT AND RESULT

To the best of our knowledge, no prior research has systematically explored the effectiveness of flow matching for generating 3D medical images. Furthermore, there is limited guidance on how to adapt, evaluate, and optimize model architectures for this purpose.

This chapter presents a series of experiments designed to investigate various configurations of the proposed model, assess their performance, and analyze the results. The primary objective is to provide insights into the practical application of flow matching for 3D medical image generation.

5.1 EXPERIMENT SETUP

All models were trained and tested on an NVIDIA RTX 2080 Ti with 11GB GPU RAM. Additional system specifications included. The VQ-GAN model parameter breakdown is presented in Table 5.1, with a total estimated model size of 119.686 MB. The flow matching model consists of 151.604 MB of parameters, representing a more streamlined architecture tailored for latent space operations.

Table 5.1: Model Parameter Breakdown

Name	**Type**	**Parameters**
Encoder	Encoder	441K
Decoder	Decoder	948K
Pre-VQ Conv	SamePadConv3D	520
Post-VQ Conv	SamePadConv3D	576
Codebook	Codebook	0
2D Image Discriminator	NLayerDiscriminator	2.8M
MRI Discriminator	NLayerDiscriminator3D	11.0M
Perceptual Model	LPIPS	14.7M
Total Trainable Params		**15.2M**
Total Non-Trainable Params		**14.7M**
Total Params		**29.9M**
Estimated Model Size		**119.686 MB**

5.2 TESTING PROCESS

The testing process involves generating new synthetic 3D MRI images by transforming a random latent vector z_1 (sampled from a standard Gaussian distribution $\mathcal{N}(0, I)$) to the target latent representation z_0 using the trained flow matching model and reconstructing the final output image using the pre-trained VQ-GAN decoder. The process is mathematically governed by the learned velocity field $v_\theta(t, z_t)$ and utilizes an Ordinary Differential Equation (ODE) solver for efficient transformation.

Steps for Testing

1. **Latent Sampling:** A latent vector $z_1 \sim \mathcal{N}(0, I)$ is sampled from a Gaussian distribution, serving as the starting point for the generative process.
2. **ODE Solver Integration:** The transformation from z_1 to z_0 is achieved by solving the following ODE backward in time:

   $$\frac{dz_t}{dt} = v_\theta(t, z_t),$$

   where z_t is the latent representation at time t. The final latent representation z_0 is computed as:

   $$z_0 = z_1 + \int_1^0 v_\theta(t, z_t)\, dt$$

   The ODE solver numerically integrates this equation over the time interval $[1, 0]$, guided by the trained velocity field $v_\theta(t, z_t)$.
3. **Image Reconstruction:** Once the latent vector z_0 is obtained, it is passed through the pre-trained VQ-GAN decoder to reconstruct the corresponding 3D MRI image:

   $$x_{\text{recon}} = D(z_0),$$

   where D is the VQ-GAN decoder.

The sampling process for generating synthetic 3D MRI images using the flow matching model is outlined in Algorithm 3.

5.3 FIRST EXPERIMENT

The first experiment involved training the VQ-GAN model to learn lower-dimensional representations of medical images, followed by training a denoising diffusion model. The objectives of this experiment were:

1. To evaluate the performance and representational ability of the VQ-GAN model.

2. To assess the generative capabilities of the diffusion model and compare its results with those of the flow matching model.

The primary purpose of training the diffusion model was to evaluate the VQ-GAN's ability to encode the input data into a latent space and reconstruct meaningful images from this representation. The diffusion model served as a test mechanism to verify whether the VQ-GAN's latent space encoding retained sufficient structural and anatomical information to produce plausible outputs. An example of the output generated by the diffusion model is shown in Figure 5.1.

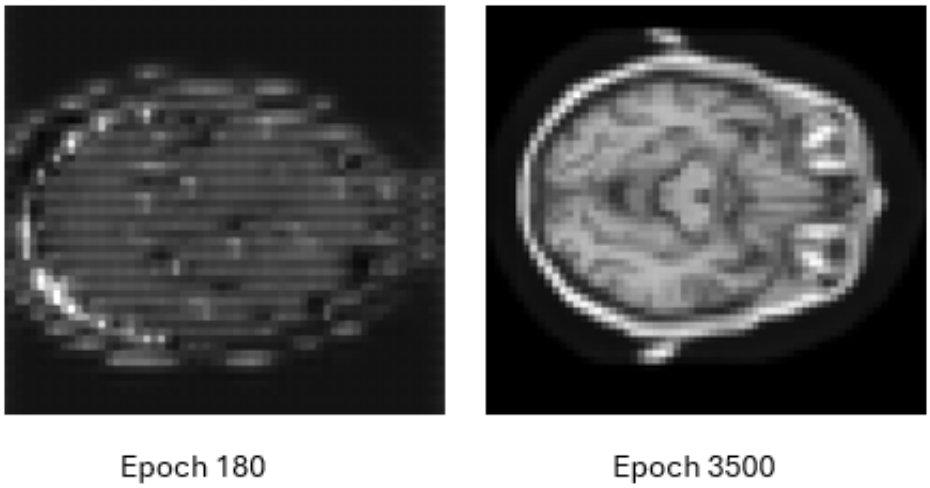

Figure 5.1: Example output from the diffusion model.

As shown in Figure 5.1, the diffusion model produced results that qualitatively resembled real medical images. Based on these outcomes, the focus shifted to implementing the flow matching model within the latent space of the VQ-GAN to evaluate its generative potential.

5.3.1 *Flow Matching Model*

The flow matching model was initially trained using a standard configuration. However, the results were unsatisfactory, as illustrated in Figure 5.2.

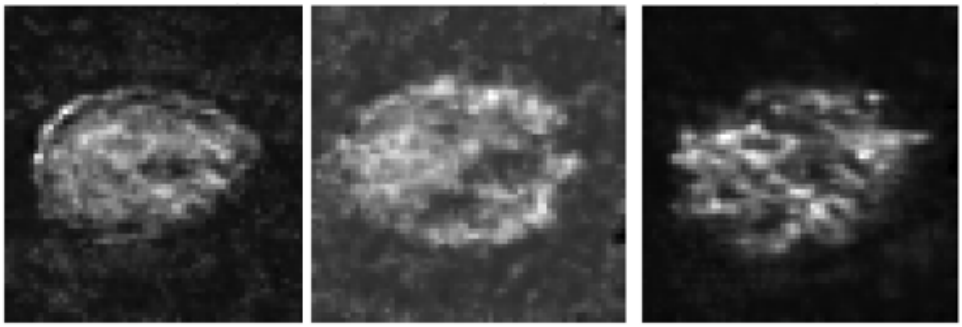

Figure 5.2: Initial results from the flow matching model, showing poor performance.

To address the issues observed, multiple attempts were made to optimize hyperparameters and explore alternative model architectures, but these attempts were unsuccessful. Eventually, an alternative loss function (L1 loss) was introduced to replace the standard L2 loss. The L1 loss, defined as the mean absolute error between the predicted and target values, penalizes large deviations less harshly than L2 loss, which calculates the mean squared error. This adjustment resulted in significantly improved performance, as shown in Figure 5.3.

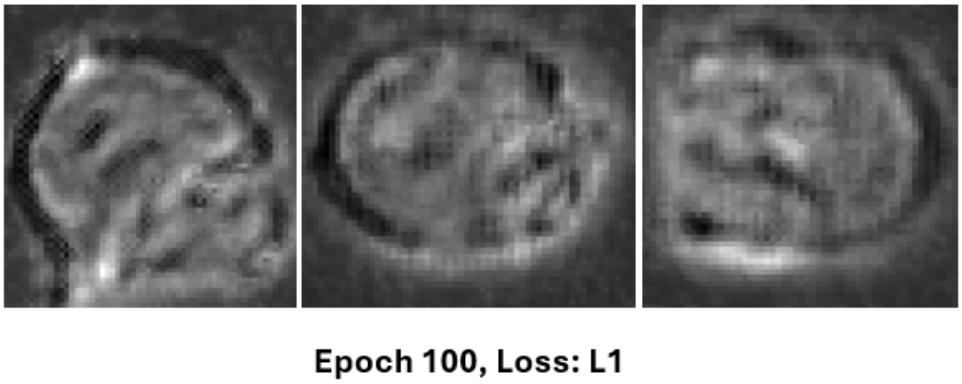

Figure 5.3: Results from the flow matching model using L1 loss.

To further enhance the results, a combination of L1 and L2 loss functions was employed with weighted contributions, balancing their strengths. This approach improved the model's ability to preserve structural consistency while maintaining robustness to outliers. One of the results from this configuration is presented in Figure 5.4, and a comparison of loss trends for different configurations is shown in Figure 5.5.

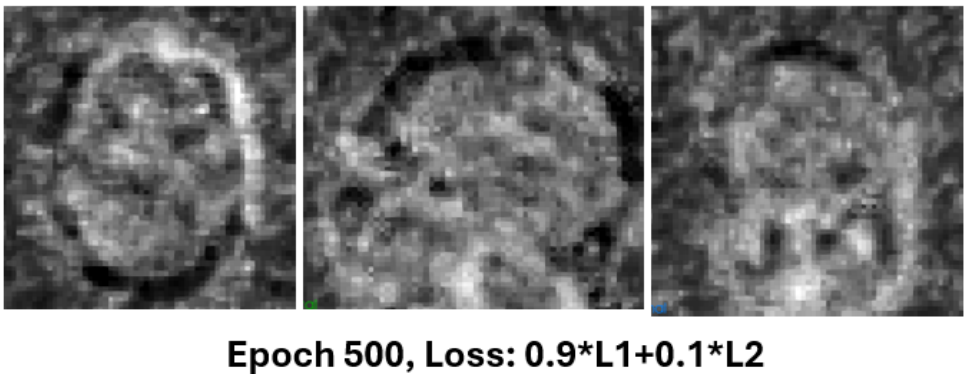

Figure 5.4: Results from the flow matching model using a combination of L1 and L2 loss functions.

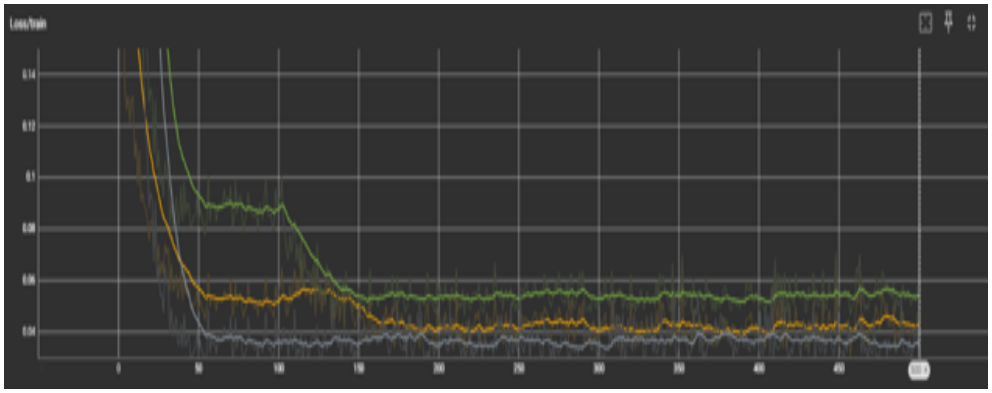

Figure 5.5: Loss function trends: gray represents L2 (0.035), yellow represents L1 (0.042), and green represents a combination of 0.9L1 + 0.1L2 (0.052).

5.3.2 *Conclusion of First Experiment*

The first experiment demonstrated that the VQ-GAN and diffusion models were sufficient for generating moderate-quality 3D medical images. However, the flow matching model initially produced unsatisfactory results, and its performance significantly improved only after adjustments to the loss function, including the introduction of L1 loss and a weighted combination of L1 and L2 losses. These findings informed subsequent experiments and highlighted the importance of loss function selection for optimizing model performance.

Before concluding that the flow matching model lacked the capability to generate meaningful outputs, the decision was made to revisit data preprocessing. Initially, preprocessing followed the procedures described in [8]. To evaluate how the VQ-GAN interpreted the input data, an additional step was introduced during training: reconstructing and visualizing multiple 2D slices extracted from the 3D medical images. This step allowed for a direct assessment of the quality of the preprocessed input data and the reconstructed outputs. Examples of the input images as seen by the VQ-GAN (prior to be-

ing fed into the networks) are shown in Figure 5.6, and the corresponding reconstruction results are presented in Figure 5.7.

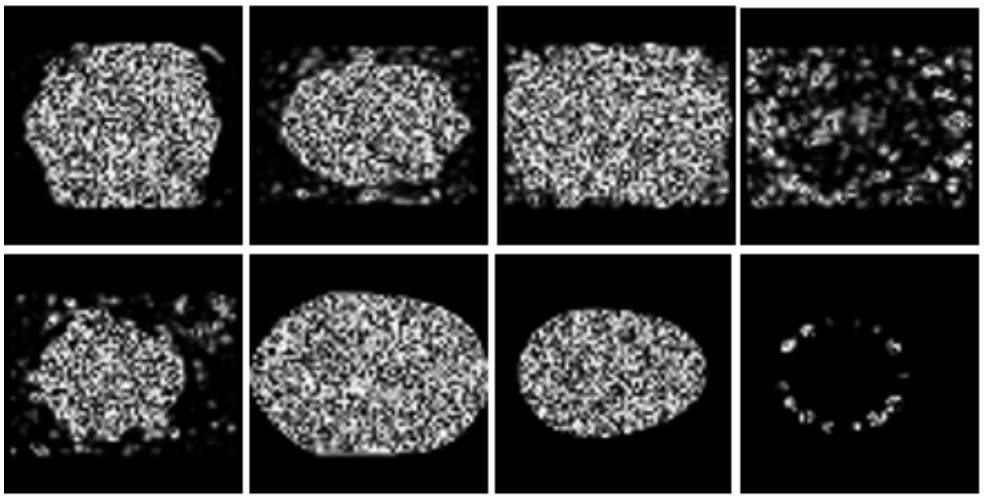

Figure 5.6: Examples of input data after preprocessing, prior to being fed into the networks.

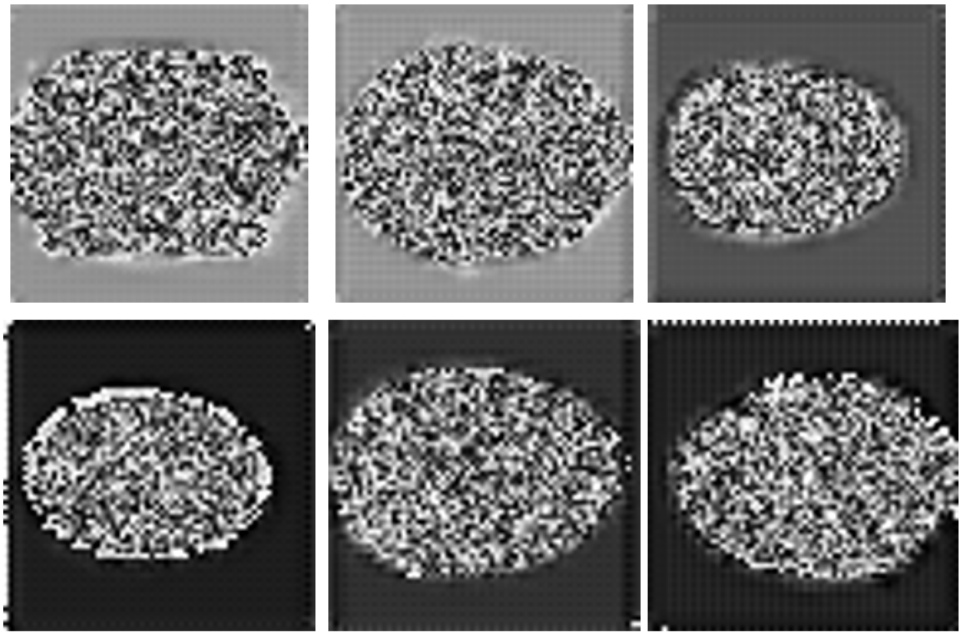

Figure 5.7: Examples of reconstructed data from the VQ-GAN output.

This additional analysis highlighted shortcomings in the initial preprocessing pipeline, which influenced the model's ability to generate high-quality outputs. These findings emphasized the importance of preprocessing in ensuring that the VQ-GAN effectively captures critical structural information from the input data. This insight became a a focus for further refinement in subsequent experiments.

5.4 SECOND EXPERIMENT

The first experiment highlighted a critical issue: the preprocessing pipeline for the VQ-GAN input data needed improvement. In response, the input data was normalized between 0 and 1 to handle extreme values effectively. Due to a CUDA out-of-memory issue, the models were trained with downsampling factors of [4, 4, 4].

Figure 5.8 demonstrates how the revised preprocessing allowed the VQ-GAN to interpret the input data more effectively, as opposed to Experiment 1, where the input data was primarily noise. This adjustment resulted in significantly better reconstruction performance, as seen in Figure 5.9.

The improved preprocessing also had a positive impact on the flow matching model. The results, shown in Figure 5.10, exhibit a significant improvement over Experiment 1. The flow matching model now generates images with well-defined brain structures and noticeably better overall quality.

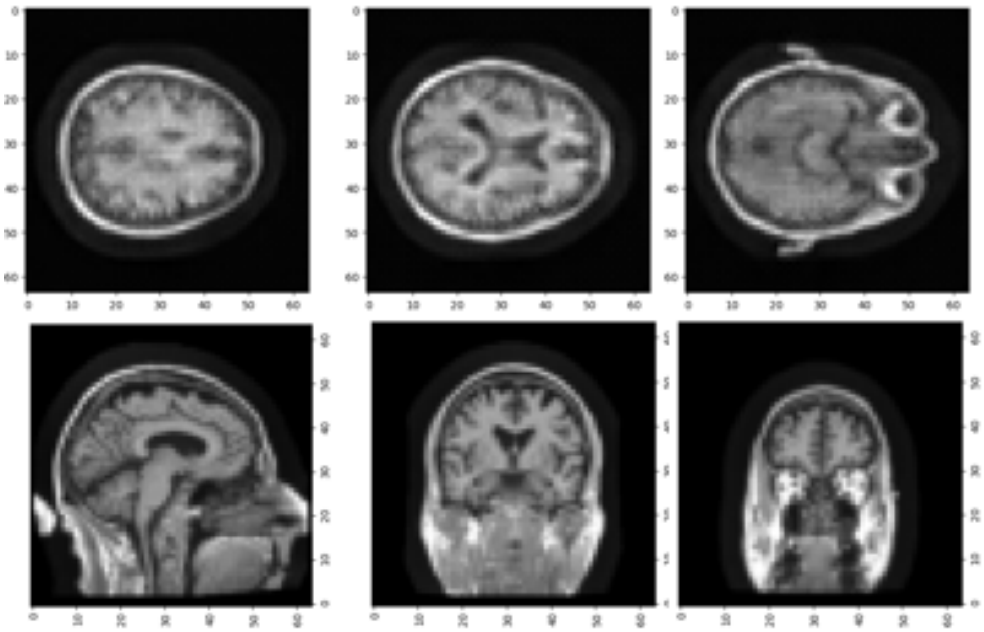

Figure 5.8: Input data after updated preprocessing and normalization. This adjustment allowed the VQ-GAN to process the data more effectively.

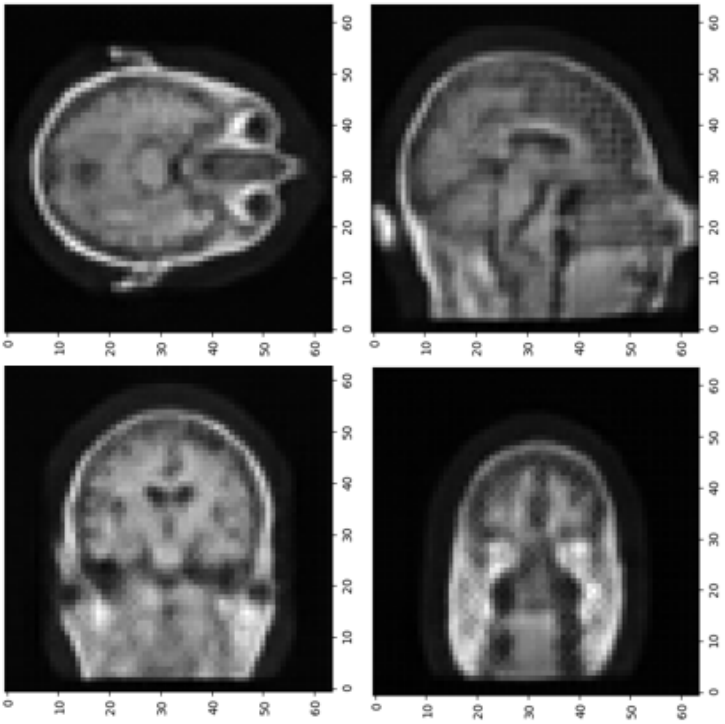

Figure 5.9: Reconstructed output of the VQ-GAN model after improved preprocessing. The reconstruction quality is significantly enhanced compared to Experiment 1.

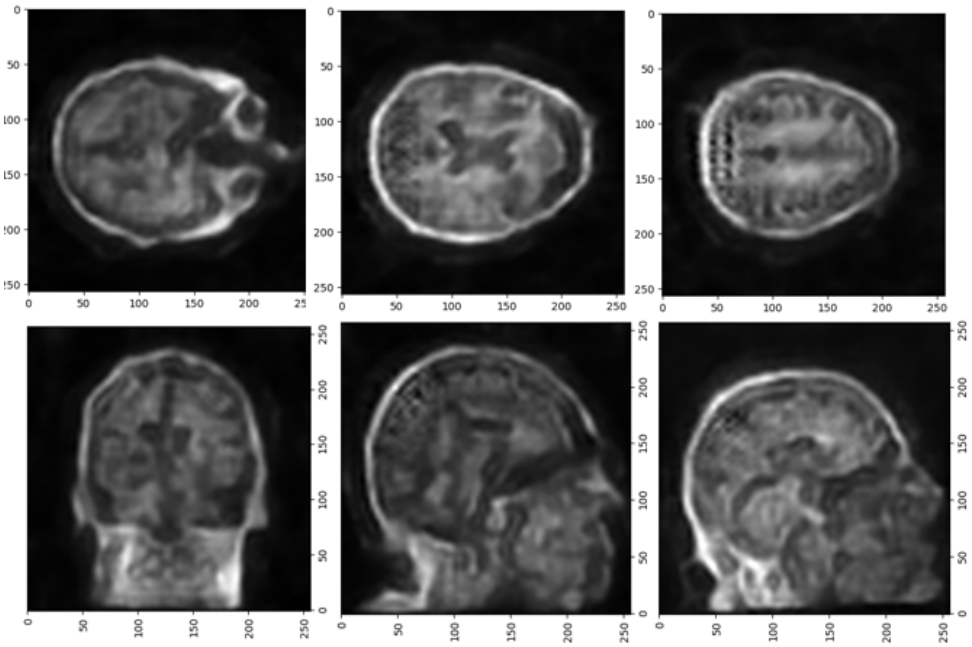

Figure 5.10: Results of the flow matching model. Improved preprocessing and reconstruction have led to clearer brain structures and better overall image quality.

5.5 EVALUATION

In order to provide a comprehensive and detailed assessment of the proposed model's performance, both quantitative and qualitative evaluations were implemented. The quantitative metrics offer objective, measurable insights into the model's ability to reproduce high-quality MRI images at both pixel and structural levels. In contrast, the qualitative assessment provides a deeper understanding of the model's strengths and limitations from a perceptual perspective.

However, due to constrained resources (time and GPU availability), the evaluation was limited to a single test example, which restricts the scope and generalizability of the findings.

5.5.1 *Quantitative Results*

The VQ-GAN and diffusion models, used for comparison with the flow matching model, required approximately 10 days of training. In contrast, the flow matching model training completed in approximately 13 hours, reflecting differences in architectural complexity and training processes.

The performance of the proposed model was evaluated using several quantitative metrics, including the Mean Squared Error (Mean Squared Error (MSE)), Normalized Mean Squared Error (Normalized Mean Squared Error (NMSE)), Peak Signal-to-Noise Ratio (Peak Signal-to-Noise Ratio (PSNR)), Structural Similarity Index Measure (Structural Similarity Index Measure (SSIM)), and Multi-Scale Structural Similarity Index Measure (Multi-Scale Structural Similarity Index Measure (MS-SSIM)). These metrics were selected to assess both pixel-level accuracy and perceptual quality. The results are summarized in Table 5.2.

Table 5.2: Quantitative Evaluation Metrics

Metric	**Value**
MSE	3.1761×10^{-2}
NMSE	1.8250
PSNR (dB)	21.00
SSIM	0.7364
MS-SSIM	0.6505

Metric Descriptions and Interpretation

- **Mean Squared Error (MSE):** MSE measures the average squared difference between corresponding pixel values in the generated and ground truth images:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2,$$

where x_i is the pixel value in the ground truth image, $\hat{x}_i$ is the corresponding pixel in the generated image, and N is the total number of pixels. A lower MSE value indicates a higher degree of pixel-wise similarity. However, MSE is sensitive to large errors and does not capture structural or perceptual differences, which can limit its usefulness in evaluating fine structural details.

- **Normalized Mean Squared Error (NMSE):** NMSE normalizes the MSE by dividing it by the norm of the ground truth image, making it more robust across datasets with varying intensity distributions:

$$\text{NMSE} = \frac{\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}{\sum_{i=1}^{N} x_i^2}.$$

A value close to zero indicates strong similarity between the generated and ground truth images. The observed NMSE of 1.8250 reflects deviations in finer details, emphasizing the need for improved structural fidelity in the generated images.

- **Peak Signal-to-Noise Ratio (PSNR):** PSNR measures the ratio between the maximum possible pixel intensity (I_{max}) and the noise present in the image, expressed in decibels (dB):

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{I_{\text{max}}^2}{\text{MSE}}\right).$$

Higher PSNR values indicate better image fidelity. A PSNR of 21.00 dB suggests moderate quality, with visible noise and artifacts impacting the clarity and sharpness of the images.

- **Structural Similarity Index Measure (SSIM):** SSIM assesses the perceived similarity of structural patterns between the generated and ground truth images by considering luminance (l), contrast (c), and structure (s):

$$\text{SSIM}(x, \hat{x}) = \frac{(2\mu_x\mu_{\hat{x}} + C_1)(2\sigma_{x\hat{x}} + C_2)}{(\mu_x^2 + \mu_{\hat{x}}^2 + C_1)(\sigma_x^2 + \sigma_{\hat{x}}^2 + C_2)},$$

where μ_x and $\mu_{\hat{x}}$ are the mean intensities, σ_x^2 and $\sigma_{\hat{x}}^2$ are variances, $\sigma_{x\hat{x}}$ is the covariance, and C_1 and C_2 are small constants to stabilize the

division. The SSIM score of 0.7364 indicates moderate preservation of structural features but suggests room for improvement in fine structural fidelity.

- **Multi-Scale Structural Similarity Index Measure (MS-SSIM):** MS-SSIM extends SSIM to multiple spatial scales, capturing structural similarity across both local and global patterns. It is computed as:

$$\text{MS-SSIM} = \prod_{j=1}^{M} \left[\text{SSIM}_j(x, \hat{x})\right]^{\alpha_j},$$

where M is the number of scales, and α_j are weighting factors for each scale. The MS-SSIM value of 0.6505 highlights challenges in reproducing fine-grained textures and subtle anatomical features, which are critical for medical imaging tasks.

The combination of these metrics provides clear evidence of the flow matching model's capacity to generate images that are, to a reasonable extent, an accurate representation of the target data. Nevertheless, further enhancements are required, particularly with regard to the enhancement of finer details and the reduction of artifacts [12].

5.5.2 *Qualitative results*

The qualitative evaluation of the generated images offers valuable insights into the performance and limitations of the proposed model. Three main aspects are examined: the quality and similarity of the generated images to the ground truth, and the challenges in achieving precise anatomical accuracy.

Quality and Similarity of Generated Images

A qualitative assessment, as demonstrated in Figure 5.11, involves comparing the ground truth images (right) and the generated ones (left). This comparison can be conducted even by individuals lacking expertise in MRI or radiology This comparison highlights both the strengths and limitations of the generative model.

Overall, the generated images effectively capture the macro-level anatomical structures, such as the general shape and layout of brain regions. However, the models are unable to reproduce the fine structural details evident in the ground truth images. This limitation is particularly noticeable in areas requiring intricate textures and sharp boundaries, which are fundamental for anatomical accuracy.

Additionally, the generated images appear slightly blurry, further reducing their quality in comparison to the ground truth. This blurriness suggests that the model struggles with high-frequency details, potentially due to limitations in resolution or training methodology. Despite these challenges, the

generated images exhibit a reasonable degree of structural similarity, indicating the model's potential for improvement with further refinements

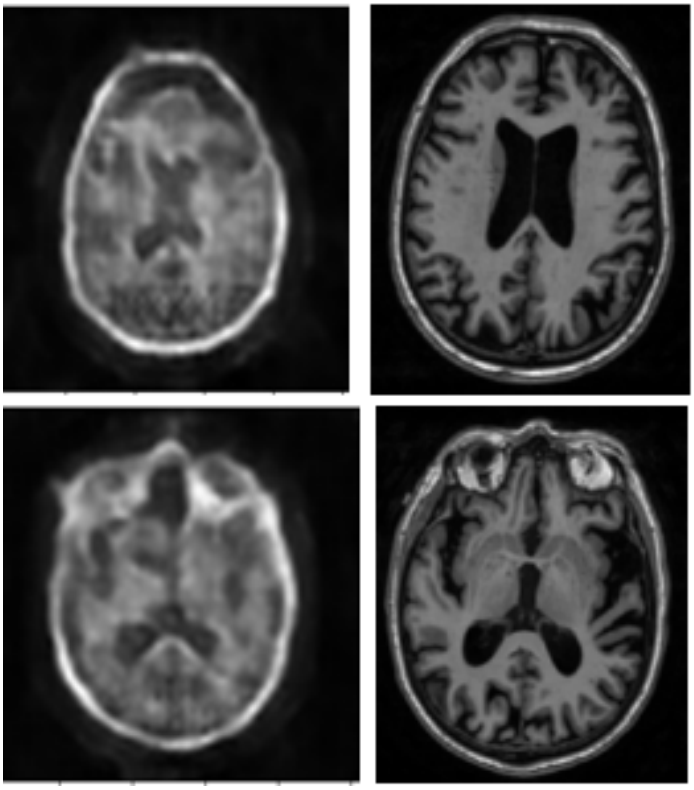

Figure 5.11: Comparison of generated MRI images (left) with ground truth images (right). While the overall structure is well-represented in the generated images, the fine details are not fully captured, and the generated images exhibit slight blurriness.

Challenges in Anatomical Accuracy

While examining the generated MRI images, artifacts were observed particularly in the parietal and occipital lobes of the brain as shown in Figure 5.12. These artifacts manifest as grid-like or structured noise patterns, which do not correspond to anatomically accurate structures. Such inconsistencies have a significant impact on the quality and usability of the generated images for medical applications, where precise anatomical accuracy is important for tasks such as diagnosis or segmentation.

The presence of these artifacts reduces the reliability of the model for real-world medical applications. Although the model generates well-structured and realistic images in other regions, it appears to struggle with capturing the finer details and complex spatial patterns in the parietal and occipital regions. This failure likely leads to inaccurate representations in the output, reducing the overall usefulness of the images for tasks that demand high anatomical fidelity.

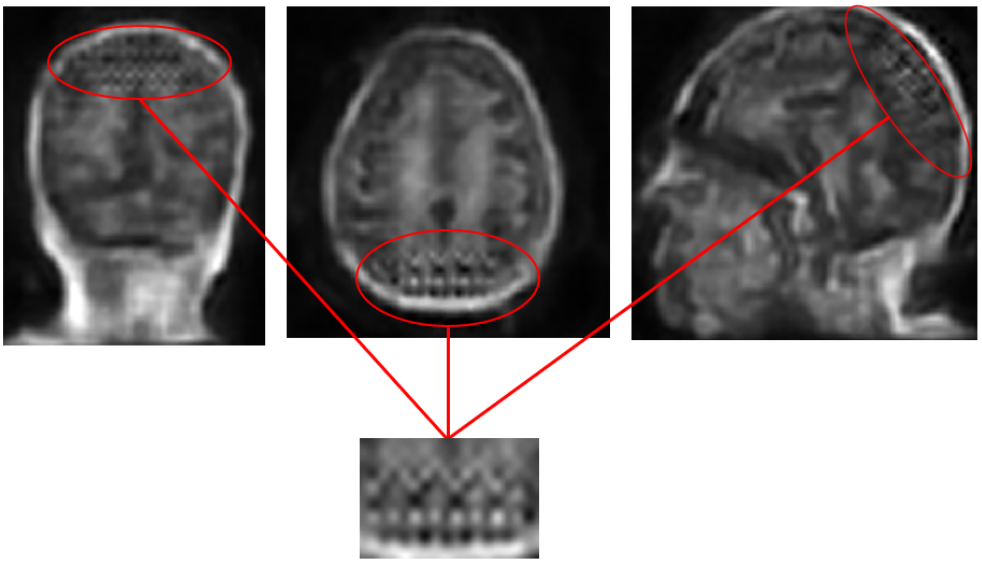

Figure 5.12: Artifacts observed in the parietal and occipital lobes of generated MRI images. These grid-like noise patterns, highlighted in red, do not correspond to anatomically accurate structures and indicate inconsistencies in the model's output

6 DISCUSSION

This chapter analyzes the experimental results within the framework proposed for generating three-dimensional medical images. The discussion highlights the effectiveness of the flow matching model, exploring its key strengths, limitations, and implications. While the analysis is based on the evaluations presented in Chapter **??**, it must be noted that the evaluation is based on a single test example, which limits the scope of the findings. This limitation underscores the need for a more thorough evaluation to validate the initial promising results and confirm the model's potential.

6.1 ANALYSIS OF QUANTITATIVE AND QUALITATIVE RESULTS

Quantitative Performance

The quantitative metrics (MSE, NMSE, PSNR, SSIM, and MS-SSIM) provide objective insights into the performance of the flow matching model, which presented in table 5.2. Although the model shows promising results in generating medical images with structural consistency and visual plausibility, its performance metrics indicate areas for improvement:

1. **Pixel-level Accuracy:** The MSE and NMSE values indicate that the generated images present moderate pixel-wise similarity to the ground truth. However, the NMSE indicates that deviations are more pronounced for finer structural details. The relatively low PSNR value (21.00 dB) reflects the presence of residual noise and the lack of high signal fidelity.

2. **Structural Consistency:** The SSIM and MS-SSIM metrics indicate that the model captures macro-level anatomical structures reasonably well. However, lower MS-SSIM values highlight challenges in preserving finer multi-scale features, such as intricate brain textures and boundaries.

While the model demonstrates an ability to reconstruct and generate images with moderate fidelity, the results suggest that the proposed method is still potentially impacted by challenges such as noisy reconstructions and loss of fine details. Further experiments could investigate the impact of further pre-processing of the input data or optimization strategies to address these issues.

Qualitative Performance

The qualitative evaluation of the generated images aligns with the quantitative results, demonstrating the model's ability to capture realistic anatomical

variability while highlighting challenges with fine detail reproduction and artifact reduction.

The variability in the outputs generated by the flow matching model, illustrated in Figure 6.1, underscores its capacity to produce a diverse range of realistic images. Such diversity is critical in medical imaging applications, where accurate representation of anatomical and pathological variations is essential.

The three examples in Figure 6.1 reveal differences in anatomical details, including tissue contrast and boundary sharpness. For instance:

- The first image exhibits sharper boundaries of the brain's outer structures, with more defined edges.
- The second image shows more uniform contrast but smoother boundaries, suggesting a different traversal pathway in latent space.
- The third image demonstrates variations in contrast and texture, reflecting subtle differences in fine-grained feature representation.

These variations highlight the model's capacity to generate a range of plausible outputs while maintaining overall anatomical fidelity. However, they also reveal challenges in achieving consistent quality across samples, potentially linked to how the flow matching process interacts with the latent space.

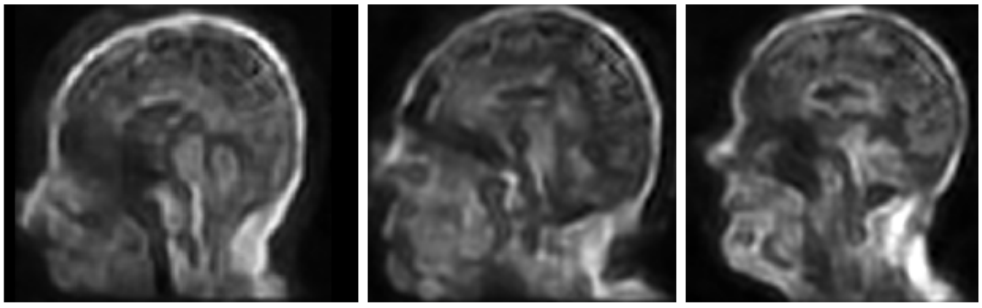

Figure 6.1: Examples of 3D slices generated by the flow matching model. The images illustrate variability in contrast and fine details, showcasing the model's ability to produce diverse representations while maintaining overall anatomical structure.

Strengths:

- The generated images successfully represent macro-level anatomical structures and maintain continuity across slices.
- The diversity in the generated images demonstrates the model's ability to explore and capture variations in anatomical details, which is vital for medical imaging applications.

Limitations:

- Fine structural details and textures, crucial for medical imaging, are often poorly represented or missing.
- The generated images appear slightly blurred, which could be attributed to limitations in preprocessing techniques or the resolution of the latent space.

6.2 IMPACT OF DATASET CHARACTERISTICS

The dataset used for training played a major role in influencing the performance of the proposed framework. While it demonstrated the potential of latent flow matching models with limited data, several inherent characteristics influenced the results:

Size and Design:

The dataset contained approximately 320 MRI images, a relatively small size for training deep generative models. This limitation was partly by design, to demonstrate the ability to generate synthetic images with limited resources, and partly due to the availability of computational resources. Despite the small size, the model successfully converged and generated realistic synthetic images without fine-tuning hyperparameters for flow matching model or encountering mode collapse during training.

Resolution Constraints:

Reduced image resolution due to preprocessing negatively impacted the quality of reconstructed images, limiting the model's ability to capture intricate anatomical details.

Data Imbalance:

An expert examination of the dataset revealed a significant imbalance, with 70% of the images representing late-stage Alzheimer's disease, 1% representing early-stage, and the remainder depicting moderate stages (see Figure 4.2 for an illustration of the differences). This imbalance introduced a bias into the generative model, leading to outputs predominantly simulating late-stage atrophy. While the model successfully captured the characteristics of late-stage Alzheimer's disease, the lack of diverse representations limited its ability to generate images depicting other stages of the disease or healthy brains.

Challenges Due to Brain Anatomy:

The complexity and fine-grained structure of the human brain present additional challenges for generative modeling. The brain's intricate textures,

subtle boundaries, and fine anatomical details, particularly in regions affected by Alzheimer's disease, require extremely high-resolution latent representations to be faithfully reproduced. In contrast, if the region of interest were an area with less intricate structures, such as the abdomen, the model might achieve better results. Organs in the abdominal region generally have smoother and less complex anatomical features, which may be easier for the model to capture and reproduce accurately, even with limited data or resolution constraints.

6.3 SIGNIFICANCE OF PREPROCESSING

The improved performance observed in the second experiment highlights the important role of preprocessing in medical image generation. Normalizing input voxel intensities to a range between 0 and 1 enhanced data consistency by ensuring uniformity across input images, allowing the model to focus on structural features rather than intensity variations. This step also contributed to more stable and efficient convergence during training by mitigating gradient-related issues. Furthermore, downsampling factors played a significant role in improving the model's ability to encode and reconstruct images effectively. However, a major factor contributing to the suboptimal qualitative and quantitative results was the choice of downsampling factor during the VQ-GAN training phase. Due to GPU memory constraints, the model was trained with a downsampling factor of [4,4,4] instead of [2,2,2]. While this decision was necessary to address computational limitations, it had a significant negative impact on the quality of the latent space representations. The higher downsampling factor reduced the overall resolution of the input data, resulting in the loss of both high-frequency anatomical details and important structural information, rather than specifically addressing noise. As a result, the latent space primarily captured macro-level features, while ignoring fine anatomical details, that are critical for high-quality image generation.

These limitations were evident in the model's failure to produce outputs with sharp boundaries and fine-grained textures, particularly in regions requiring high anatomical fidelity. These findings emphasize that preprocessing decisions, in particular the choice of downsampling factor, are not just technical adjustments but critical determinants of model performance. Future work should prioritize training with finer downsampling factors, such as [2,2,2], to better preserve the structural integrity of the latent space, and should explore advanced preprocessing techniques, such that dynamic downsampling and data augmentation, to optimize both computational efficiency and anatomical accuracy.

6.4 POSSIBLE AREAS FOR IMPROVEMENT

Several areas can be explored to enhance the performance and applicability of the proposed framework:

1. **Training with Finer Downsampling Factors:** Using a finer downsampling factor, such as [2,2,2], during the VQ-GAN training stage could help preserve more intricate anatomical details in the latent space. While this requires higher computational resources, it would significantly improve the quality and fidelity of the generated images, particularly for regions requiring diagnostic precision.

2. **Balancing and Expanding the Dataset:** Addressing dataset imbalance is crucial to improving the diversity and generalization ability of the model. Including more examples of early- and moderate-stage Alzheimer's disease, as well as healthy brains, would help the model generate outputs that better represent the full spectrum of conditions. Additionally, data augmentation techniques, such as rotation, flipping, scaling, and elastic deformations, could enhance the dataset without requiring additional MRI scans.

3. **Using Higher-Resolution Data:** Training with state-of-the-art resolution MRI datasets would allow the model to learn and generate images with greater structural fidelity. Higher-resolution input data would enhance the model's ability to capture fine-grained textures, subtle anatomical features, and sharp boundaries, particularly in critical regions affected by Alzheimer's disease.

4. **Adopting Region-Specific Models:** Regions of the body with less complex structures, such as the abdomen, could serve as a benchmark for testing and refining the framework. These areas often lack the high complexity of brain anatomy, making them more suitable for generative modeling. Success in simpler anatomical regions could inform strategies for tackling more complex areas like the brain.

5. **Using Diffusion Transformer (DIT) and Fine-Tuning:** Replacing the U-net architecture with a diffusion transformer (DIT) based architecture for the flow matching model could improve the performance of the framework. DIT offers the capability to better capture long-range dependencies and complex structural details. Fine-tuning the model after replacing the architecture would optimize its parameters for high quality image generation [6].

6. **Using a Refinement Network:** To enhance the quality and temporal consistency of the generated images, a refinement network, which is suggested in [7] could be utilized. This network would take two images as input, refining the second image based on the first. Such an approach could reduce artifacts, enhance smooth transitions across slices, and improve the overall coherence of the generated 3D medical volumes.

7. **Exploring Further Preprocessing Techniques:** Advanced preprocessing strategies could enhance the framework's ability to interpret and process medical images. For instance, dynamic normalization methods,

histogram equalization, or advanced noise reduction algorithms could improve the input data quality. Additionally, better handling of voxel intensity distributions and employing cropping techniques tailored to specific anatomical regions could ensure greater consistency and relevance of the input data.

8. **Using a More Powerful GPU:** Training and testing on better GPUs would mitigate issues such as CUDA out-of-memory errors, allowing for smoother execution of models with higher resolution data and finer downsampling factors. Access to advanced hardware would also enable the use of larger batch sizes, leading to improved model stability and convergence.

7 CONCLUSIONS

This thesis explores the integration of VQ-GAN and flow matching models for the generation of synthetic 3D medical images, focusing on MRI scans of Alzheimer's patients. The results demonstrate the potential of this novel approach in addressing challenges related to data scarcity, patient privacy, and the need for high-quality, diverse medical imaging datasets.

The proposed framework successfully captured macro-level anatomical structures and generated plausible 3D images, reducing training time dramatically from 10 days, as required by diffusion models, to just 13 hours. This indicates that the framework is a promising candidate for use in medical applications. Nevertheless, the presence of limitations, including the loss of fine details, the appearance of artefacts in specific regions, and the potential for dataset bias, indicates the necessity for further improvement. The importance of preprocessing, dataset diversity, and loss function selection was emphasized throughout the evaluation, as these factors significantly influenced model performance.

Despite these challenges, the combination of VQ-GAN's latent space encoding and flow matching's continuous transformations represent a promising direction in medical image synthesis. With further refinements, such as training on higher-resolution data, addressing dataset imbalances, and exploring advanced model architectures, the framework could be extended to generate high-quality, reliable images for medical imaging.

This research provides a foundation for future work in generative modeling for medical imaging, with potential applications in disease diagnosis, treatment planning, and the development of AI-driven healthcare tools. It also underscores the need for continued exploration of resource-efficient, high-quality generative models to advance the field of medical image synthesis.

BIBLIOGRAPHY

[1] Andrea Asperti. *Sparsity in Variational Autoencoders*. 2019. arXiv: 1812.07238 [cs.LG]. URL: https://arxiv.org/abs/1812.07238 (cit. on p. 14).

[2] 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." In: *Scientific Data* 4 (2017). URL: https://api.semanticscholar.org/CorpusID:3697707 (cit. on p. 11).

[3] Heli Ben-Hamu, Samuel Cohen, Joey Bose, Brandon Amos, Aditya Grover, Maximilian Nickel, Ricky T. Q. Chen, and Yaron Lipman. *Matching Normalizing Flows and Probability Paths on Manifolds*. 2022. arXiv: 2207.04711 [stat.ML]. URL: https://arxiv.org/abs/2207.04711 (cit. on p. 27).

[4] R. W. Brown. *Magnetic Resonance Imaging: Physical Principles and Sequence Design*. 4th. Hoboken, NJ: Wiley, 2018 (cit. on p. 5).

[5] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. *Neural Ordinary Differential Equations*. 2019. arXiv: 1806.07366 [cs.LG]. URL: https://arxiv.org/abs/1806.07366 (cit. on pp. 18, 20, 27).

[6] Quan Dao, Hao Phung, Binh Nguyen, and Anh Tran. *Flow Matching in Latent Space*. 2023. arXiv: 2307.08698 [cs.CV]. URL: https://arxiv.org/abs/2307.08698 (cit. on pp. 34, 35, 43, 63).

[7] Aram Davtyan, Sepehr Sameni, and Paolo Favaro. "Efficient Video Prediction via Sparsely Conditioned Flow Matching." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023, pp. 23263–23274 (cit. on pp. 34, 63).

[8] Prafulla Dhariwal and Alex Nichol. *Diffusion Models Beat GANs on Image Synthesis*. 2021. arXiv: 2105.05233 [cs.LG]. URL: https://arxiv.org/abs/2105.05233 (cit. on pp. 27, 50).

[9] Encord. *Stable Diffusion 3: Text-to-Image Model*. Accessed: 2023-12-04. 2023. URL: https://encord.com/blog/stable-diffusion-3-text-to-image-model/?utm_source=chatgpt.com (cit. on p. 3).

[10] Patrick Esser, Robin Rombach, and Björn Ommer. *Taming Transformers for High-Resolution Image Synthesis*. 2021. arXiv: 2012.09841 [cs.CV]. URL: https://arxiv.org/abs/2012.09841 (cit. on pp. 29, 31, 41).

[11] Maayan Frid-Adar, Idit Diamant, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. "GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification." In: *Neurocomputing* 321 (Dec. 2018), 321–331. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2018.09.013. URL: http://dx.doi.org/10.1016/j.neucom.2018.09.013 (cit. on p. 2).

[12] Paul Friedrich, Yannik Frisch, and Philippe C. Cattin. *Deep Generative Models for 3D Medical Image Synthesis*. 2024. arXiv: 2410.17664 [eess.IV]. URL: https://arxiv.org/abs/2410.17664 (cit. on pp. 10, 12, 13, 15, 56).

[13] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. *Generative Adversarial Networks*. 2014. arXiv: 1406.2661 [stat.ML]. URL: https://arxiv.org/abs/1406.2661 (cit. on p. 10).

[14] Ian Goodfellow. *NIPS 2016 Tutorial: Generative Adversarial Networks*. 2017. arXiv: 1701.00160 [cs.LG]. URL: https://arxiv.org/abs/1701.00160 (cit. on pp. 8, 9).

[15] Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. *FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models*. 2018. arXiv: 1810.01367 [cs.LG]. URL: https://arxiv.org/abs/1810.01367 (cit. on pp. 18, 21).

[16] Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. *FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models*. 2018. arXiv: 1810.01367 [cs.LG]. URL: https://arxiv.org/abs/1810.01367 (cit. on p. 27).

[17] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. *Improved Training of Wasserstein GANs*. 2017. arXiv: 1704.00028 [cs.LG]. URL: https://arxiv.org/abs/1704.00028 (cit. on p. 11).

[18] Pengfei Guo et al. *MAISI: Medical AI for Synthetic Imaging*. 2024. arXiv: 2409.11169 [eess.IV]. URL: https://arxiv.org/abs/2409.11169 (cit. on p. 25).

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. *Denoising Diffusion Probabilistic Models*. 2020. arXiv: 2006.11239 [cs.LG]. URL: https://arxiv.org/abs/2006.11239 (cit. on pp. 16, 27).

[20] Guillermo Iglesias, Edgar Talavera, and Alberto Díaz-Álvarez. "A survey on GANs for computer vision: Recent research, analysis and taxonomy." In: *Computer Science Review* 48 (2023), p. 100553. ISSN: 1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2023.100553. URL: https://www.sciencedirect.com/science/article/pii/S1574013723000205 (cit. on p. 11).

[21] Showrov Islam, Md. Tarek Aziz, Hadiur Rahman Nabil, Jamin Rahman Jim, M. F. Mridha, Md. Mohsin Kabir, Nobuyoshi Asai, and Jungpil Shin. "Generative Adversarial Networks (GANs) in Medical Imaging: Advancements, Applications, and Challenges." In: *IEEE Access* 12 (2024), pp. 35728–35753. DOI: 10.1109/ACCESS.2024.3370848 (cit. on p. 25).

[22] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. *Image-to-Image Translation with Conditional Adversarial Networks*. 2018. arXiv: 1611.07004 [cs.CV]. URL: https://arxiv.org/abs/1611.07004 (cit. on p. 30).

[23] Firas Khader et al. *Medical Diffusion: Denoising Diffusion Probabilistic Models for 3D Medical Image Generation*. 2023. arXiv: 2211.03364 [eess.IV]. URL: https://arxiv.org/abs/2211.03364 (cit. on pp. 29, 39, 41).

[24] Diederik P. Kingma and Max Welling. "An Introduction to Variational Autoencoders." In: *Foundations and Trends® in Machine Learning* 12.4 (2019), 307–392. ISSN: 1935-8245. DOI: 10.1561/2200000056. URL: http://dx.doi.org/10.1561/2200000056 (cit. on pp. 13, 26).

[25] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2022. arXiv: 1312.6114 [stat.ML]. URL: https://arxiv.org/abs/1312.6114 (cit. on p. 34).

[26] Ivan Kobyzev, Simon J.D. Prince, and Marcus A. Brubaker. "Normalizing Flows: An Introduction and Review of Current Methods." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.11 (Nov. 2021), 3964–3979. ISSN: 1939-3539. DOI: 10.1109/tpami.2020.2992934. URL: http://dx.doi.org/10.1109/TPAMI.2020.2992934 (cit. on p. 18).

[27] Alex Lamb. *A Brief Introduction to Generative Models*. 2021. arXiv: 2103.00265 [cs.LG]. URL: https://arxiv.org/abs/2103.00265 (cit. on p. 8).

[28] J. Langr and V. Bok. *GANs in Action: Deep learning with Generative Adversarial Networks*. Manning, 2019. ISBN: 9781617295560. URL: https://books.google.de/books?id=HojvugEACAAJ (cit. on pp. 11, 13, 26).

[29] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. *Flow Matching for Generative Modeling*. 2023. arXiv: 2210.02747 [cs.LG]. URL: https://arxiv.org/abs/2210.02747 (cit. on pp. 2, 3, 21, 23, 27, 29).

[30] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A.W.M. van der Laak, Bram van Ginneken, and Clara I. Sánchez. "A survey on deep learning in medical image analysis." In: *Medical Image Analysis* 42 (Dec. 2017), 60–88. ISSN: 1361-8415. DOI: 10.1016/j.media.2017.07.005. URL: http://dx.doi.org/10.1016/j.media.2017.07.005 (cit. on p. 2).

[31] Xingchao Liu, Chengyue Gong, and Qiang Liu. *Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow*. 2022. arXiv: 2209.03003 [cs.LG]. URL: https://arxiv.org/abs/2209.03003 (cit. on p. 35).

[32] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. *Are GANs Created Equal? A Large-Scale Study*. 2018. arXiv: 1711.10337 [stat.ML]. URL: https://arxiv.org/abs/1711.10337 (cit. on p. 25).

[33] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. *Are GANs Created Equal? A Large-Scale Study*. 2018. arXiv: 1711.10337 [stat.ML]. URL: https://arxiv.org/abs/1711.10337 (cit. on p. 25).

[34] Xudong Mao, Qing Li, Haoran Xie, Raymond Lau, Wang Zhen, and Stephen Smolley. "Least Squares Generative Adversarial Networks." In: Oct. 2017, pp. 2813–2821. DOI: 10.1109/ICCV.2017.304 (cit. on p. 11).

[35] Frederic H. Martini, Judi L. Nath, and Edwin F. Bartholomew. *Fundamentals of Anatomy & Physiology*. 11th. Hoboken, NJ: Pearson, 2018. ISBN: 9780134396026 (cit. on p. 7).

[36] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. *Spectral Normalization for Generative Adversarial Networks*. 2018. arXiv: 1802.05957 [cs.LG]. URL: https://arxiv.org/abs/1802.05957 (cit. on p. 11).

[37] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. *Neural Discrete Representation Learning*. 2018. arXiv: 1711.00937 [cs.LG]. URL: https://arxiv.org/abs/1711.00937 (cit. on p. 29).

[38] R. C. Petersen et al. "Alzheimer's Disease Neuroimaging Initiative (ADNI): Clinical characterization." English (US). In: *Neurology* 74.3 (Jan. 2010), pp. 201–209. ISSN: 0028-3878. DOI: 10.1212/WNL.0b013e3181cb3e25 (cit. on p. 39).

[39] Walter H. L. Pinaya et al. *Generative AI for Medical Imaging: extending the MONAI Framework*. 2023. arXiv: 2307.15208 [eess.IV]. URL: https://arxiv.org/abs/2307.15208 (cit. on p. 9).

[40] Anoushka Popuri and John Miller. "Generative Adversarial Networks in Image Generation and Recognition." In: *2023 International Conference on Computational Science and Computational Intelligence (CSCI)*. 2023, pp. 1294–1297. DOI: 10.1109/CSCI62032.2023.00212 (cit. on p. 10).

[41] Simon J.D. Prince. *Understanding Deep Learning*. The MIT Press, 2023. URL: http://udlbook.com (cit. on pp. 16, 18).

[42] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. *High-Resolution Image Synthesis with Latent Diffusion Models*. 2022. arXiv: 2112.10752 [cs.CV]. URL: https://arxiv.org/abs/2112.10752 (cit. on p. 34).

[43] Noam Rozen, Aditya Grover, Maximilian Nickel, and Yaron Lipman. *Moser Flow: Divergence-based Generative Modeling on Manifolds*. 2021. arXiv: 2108.08052 [stat.ML]. URL: https://arxiv.org/abs/2108.08052 (cit. on p. 27).

[44] Yasin Shokrollahi, Sahar Yarmohammadtoosky, Matthew M. Nikahd, Pengfei Dong, Xianqi Li, and Linxia Gu. *A Comprehensive Review of Generative AI in Healthcare*. 2023. arXiv: 2310.00795 [cs.LG]. URL: https://arxiv.org/abs/2310.00795 (cit. on p. 26).

[45] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. *Deep Unsupervised Learning using Nonequilibrium Thermodynamics*. 2015. arXiv: 1503.03585 [cs.LG]. URL: https://arxiv.org/abs/1503.03585 (cit. on p. 14).

[46] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. *Score-Based Generative Modeling through Stochastic Differential Equations*. 2021. arXiv: 2011.13456 [cs.LG]. URL: https://arxiv.org/abs/2011.13456 (cit. on p. 35).

[47] Digital Vital. *Medical Imaging 3D Reconstruction Challenges*. Accessed: 2024-12-04. 2024. URL: https://digitalvital.org/medical-imaging-3d-reconstruction-challenges/ (cit. on p. 25).

[48] Sanchayan Vivekananthan. *Comparative Analysis of Generative Models: Enhancing Image Synthesis with VAEs, GANs, and Stable Diffusion*. 2024. arXiv: 2408.08751 [cs.CV]. URL: https://arxiv.org/abs/2408.08751 (cit. on p. 27).

[49] Jason Walsh, Alice Othmani Hiring Postdocs, Mayank Jain, and Soumyabrata Dev. "Using U-Net network for efficient brain tumor segmentation in MRI images." In: *Healthcare Analytics* 2 (Aug. 2022), p. 100098. DOI: 10.1016/j.health.2022.100098 (cit. on p. 6).

[50] Yidan Xu, Jiaqing Liang, Yaoyao Zhuo, Lei Liu, Yanghua Xiao, and Lingxiao Zhou. "TDASD: Generating medically significant fine-grained lung adenocarcinoma nodule CT images based on stable diffusion models with limited sample size." In: *Computer Methods and Programs in Biomedicine* 248 (2024), p. 108103. ISSN: 0169-2607. DOI: https://doi.org/10.1016/j.cmpb.2024.108103. URL: https://www.sciencedirect.com/science/article/pii/S0169260724000993 (cit. on p. 25).

[51] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. *Diffusion Models: A Comprehensive Survey of Methods and Applications*. 2024. arXiv: 2209.00796 [cs.LG]. URL: https://arxiv.org/abs/2209.00796 (cit. on pp. 16, 27).

[52] Xin Yi, Ekta Walia, and Paul Babyn. "Generative adversarial network in medical imaging: A review." In: *Medical Image Analysis* 58 (Dec. 2019), p. 101552. ISSN: 1361-8415. DOI: 10.1016/j.media.2019.101552. URL: http://dx.doi.org/10.1016/j.media.2019.101552 (cit. on p. 11).

[53] Chika Yinka-Banjo and Ogban-Asuquo Ugot. "A review of generative adversarial networks and its application in cybersecurity." In: *Artificial Intelligence Review* 53.3 (Mar. 2020), pp. 1721–1736. ISSN: 1573-7462. DOI: 10.1007/s10462-019-09717-4. URL: https://doi.org/10.1007/s10462-019-09717-4 (cit. on p. 9).