Imperial College London
Department of Earth Science and Engineering

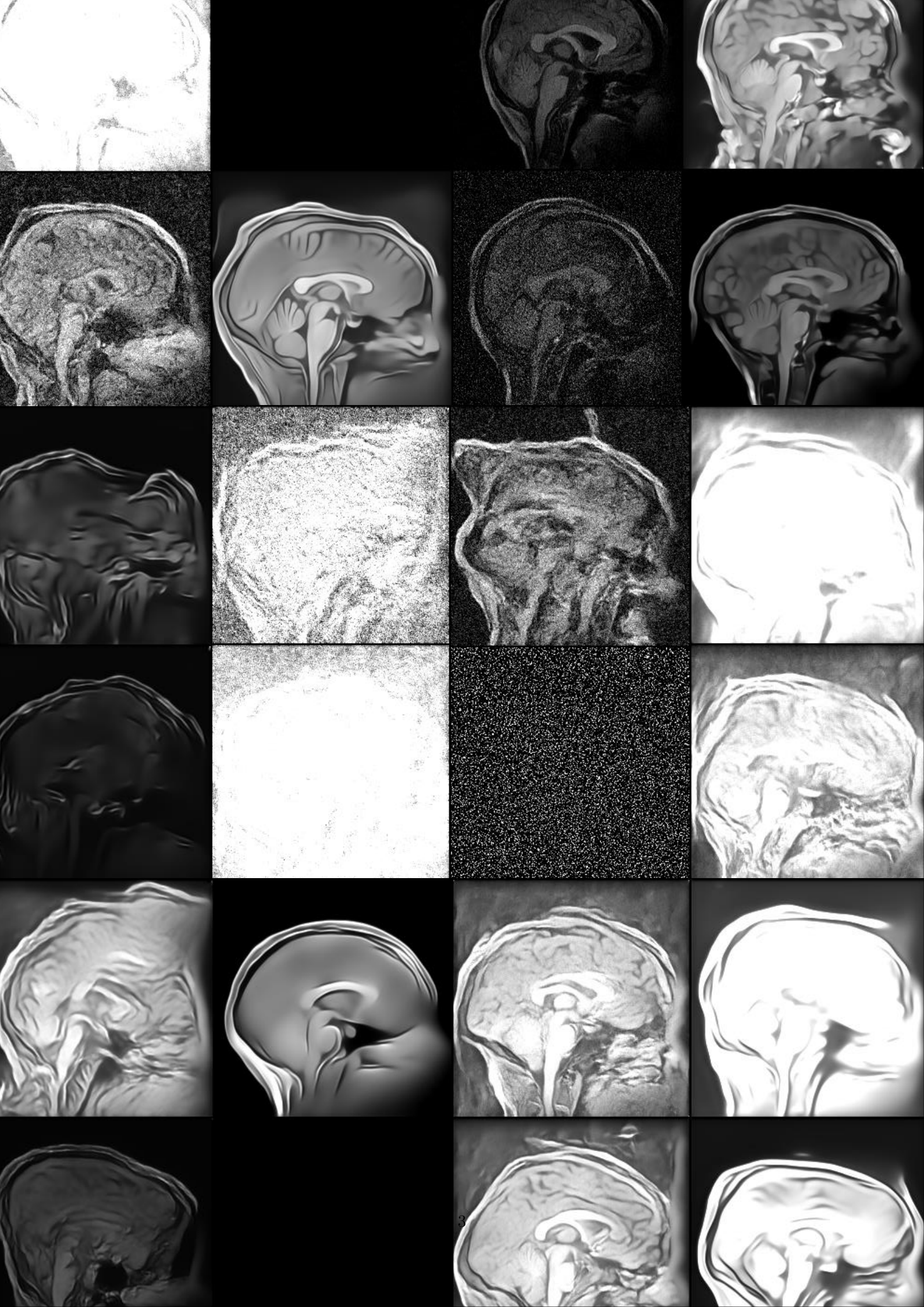
Ultrasound Brain Imaging with Latent Diffusion Models

Deborah Pelacani Cruz

December 2024

Supervised by Dr. Lluis Guasch and Prof. Mike Warner

Submitted in partial fulfilment of the requirements for the degree of
Doctor of Philosophy of Imperial College London and
the Diploma of Imperial College London



Faces of Intelligence

© *Deborah Pelacani Cruz, 2024*

Declaration of Originality

I hereby declare that this dissertation is my own work and has not been submitted in any form for another degree or diploma at any university or other institution. I certify that all material in this dissertation which is not my own work has been duly acknowledged.

Deborah Pelacani Cruz

Copyright Declaration

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC).

Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes.

Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Abstract

Recent advancements in medical imaging have demonstrated the potential of full-waveform inversion (FWI) of ultrasound data, a technique developed in geophysics, to produce high-fidelity images of the human body at remarkable resolutions. For the first time, FWI has enabled the visualisation of fine anatomical structures in the human brain using ultrasound tomography, marking an unprecedented achievement in healthcare research. However, FWI's ill-posed nature, coupled with its computational demands, renders it impractical for immediate clinical deployment. Deep neural networks, meanwhile, have driven progress across scientific domains, with diffusion models gaining recognition for their ability to model complex probability distributions and synthesise high-quality data. In this work, I propose a method for recovering high-fidelity, high-resolution images of the human brain with conditioned latent diffusion models (LDMs), as a means to bypass the limitations of FWI.

The research unfolds in three stages, each offering distinct contributions that advance ultrasound brain imaging. The first introduces a novel loss function, the Wiener Loss, which employs convolutional analysis to compare data without relying on local, element-wise relationships or costly model inferences. This enables high-fidelity analysis of complex structures, such as the human brain, making it well-suited to deep learning tasks requiring contextual and perceptual accuracy. The second stage constructs and trains a condition-free LDM on a limited dataset of acoustic neuroimages, demonstrating its ability to rapidly generate statistically conforming, diverse synthetic samples for data augmentation. The third employs conditional LDMs to learn the statistical mapping between acoustic velocity models of the brain and associated ultrasound data in a synthetic setting, enabling high-fidelity image recovery at near-real time. Although time and resource constraints limited success in the final stage, preliminary results show strong potential for this approach as a fast, efficient alternative to FWI. Its independence from physical constraints suggests broader applicability to other data inversion challenges.

Keywords: full-waveform inversion; deep learning; diffusion models; generative artificial intelligence; ultrasound tomography; brain imaging; loss functions; synthetic data augmentation.

About the Author

Deborah Pelacani Cruz is an AI researcher, software developer and science educator, passionate about advancing scientific understanding through innovative applications of artificial intelligence. Born in Salvador, Brazil, to academic parents Claudieia Regina Pelacani and Jailson Lopes Cruz, they moved to the United Kingdom at the age 17 in 2013 to pursue their academic studies. Excelling at the University of Southampton, Deborah completed an undergraduate degree in Geophysical Sciences with distinction, achieving the highest yearly academic average within the School of Ocean and Earth Sciences from 2014 to 2017. Deborah continued with their graduate studies at Imperial College London, earning an MSc in Applied Computational Science and Engineering with distinction under a prestigious scholarship from TotalEnergies SE, where they conducted impactful research in seismic imaging. In 2019, Deborah began a PhD at Imperial College London, initially funded by TotalEnergies SE, to explore novel applications of deep learning in geophysical imaging. Partway through, they were appointed as a Research Assistant, transferring their expertise to the field of medical imaging. Recently, their work has been published and presented at the 2024 Neural Information Processing Systems (NeurIPS) conference's Medical Imaging Workshop. As a Scientific Software Developer at Sonalis, Deborah further contributed to novel research in full-waveform inversion for brain ultrasound imaging. Alongside research, Deborah has been deeply involved in teaching roles. After several years as a Graduate Teaching Assistant, they received the 2021 GTA of the Year Award from Imperial's Faculty of Engineering, an honour voted on by students. Recently appointed as a Teaching Fellow at Imperial College London, Deborah now actively partakes in graduate-level courses in deep learning, numerical inversion and optimisation, within the Department of Earth Science and Engineering, supporting and guiding students across AI and computational science domains.

Acknowledgements

I would like to express my deepest gratitude to the people that have followed closely and have been instrumental in my PhD journey.

To my supervisors, Dr. Lluis Guasch and Prof. Mike Warner, and my academic tutor Dr. Adriana Paluszny, who believed in my potential and encouraged me to believe in it as much as they did. Thank you for giving me opportunities I had only dreamed of having and for seeing my limitations without letting them define me.

To my research group, for being a collaborative, passionate, and motivational team. Special thanks to Dr. George Strong, Dr. Carlos Cueto, and Dr. Oscar Bates for the infinite wisdom and guidance they provided for this work. To all the administrative staff and technicians at Imperial College London, your support behind the scenes made everything possible.

To my friends, John, Oli, Wassay, Johannes, Mina, Tatiana, Marie, Nicolle, Heloisa, Manu, Jujuba, Anna, Divya, Felix, and Evie, thank you for being there when I needed you the most. For the endless moments of laughter and joy, and for the love and support you have given me. The memories we have created together are cherished. And to my best mate John: we go to the trenches together.

To Bella, your love and partnership have sustained me through this journey. I am grateful for all the good moments we have shared. You bring me home.

To my late grandmothers, who taught me kindness above all, your lessons remain in my heart. To my parents, who have unconditionally loved me and endured the joy and the grief of seeing me so far from home. You have supported me in all aspects of my life and taught me the value of discipline, hardship, and perseverance. Your stories are a reminder of what is good in humanity. To my caring mother and to my loving and stubborn father, thank you.

To my queer community, who always resists and persists, thank you for showing me that my existence is beautiful, for giving me a voice to shout, for respecting my doubts, and for embracing all versions of me. Your joy and community means everything to me.

Sorellina, this thesis is dedicated to you. *Você que sempre me mostra carinho e respeito incondicionais, que me eleva e me transforma em uma pessoa melhor a cada dia. Obrigada por escolher caminhar ao meu lado.*

Thank you all from the bottom of my heart.

Contents

Abstract **v**

About the Author **vii**

Acknowledgements **ix**

List of Algorithms **xv**

List of Tables **xvii**

List of Terms **xix**

List of Figures **xxiii**

1 Introduction **1**

1.1 Imaging the Human Brain 1

1.2 Deep Learning: from classifiers to creators 4

1.3 Motivation and Contributions 7

1.3.1 Chapters Overview 7

1.3.2 Publications 9

1.4 Ethical Statement 10

2 Background Theory **13**

2.1 Generative Artificial Intelligence 13

2.1.1 Deep Learning Foundations 15

2.1.2 Loss Functions and Optimisation Algorithms 30

2.1.3 Variational Autoencoders 43

2.1.4 Generative Adversarial Networks 49

2.1.5 Diffusion Models 56

2.1.6 Transformers and Attention Mechanisms 80

2.1.7 Additional Generative Frameworks 85
2.1.8 Metrics and Benchmarks for Image Generation 88
2.2 Ultrasound Tomography 93
2.2.1 Full-waveform Inversion 96
2.2.2 Full-waveform Inversion with Neural Networks 112

3 Convolutional Data Comparison with Wiener Filters 119
3.1 Wiener Loss Formulation 121
3.1.1 Fourier-Domain Implementation 125
3.1.2 Statistical Interpretation 125
3.1.3 Optimisation Landscape 127
3.2 Results and Discussion 129
3.2.1 Framework and Implementation 129
3.2.2 Reconstruction of MNIST 135
3.2.3 Normalisation and Relative Pre-Whitening 141
3.2.4 Reconstruction of Natural Images 141
3.2.5 Medical Scan Imputation 147
3.3 Further Applications 151
3.3.1 Translation Invariance 151
3.3.2 Wiener Diffusion 153
3.3.3 Wiener Attention 157
3.4 Future Work 158
3.5 Summary 159

4 Generative Ultrasound Brain Imaging with Unconditional LDMs 161
4.1 Latent Diffusion Models 164
4.2 Dataset and Pre-Processing 169
4.3 Results and Discussion 173
4.3.1 Framework and Implementation 173
4.3.2 Structural Recovery with Perceptual Losses 176
4.3.3 Generative Modelling 181
4.4 Future Work 187
4.5 Summary 190

5 Towards Ultrasound Tomography with Generative Models **195**

5.1 Conditional Latent Diffusion Models 197

5.2 Ultrasound Forward Modelling 201

5.3 Encoding Ultrasound Data 203

5.3.1 Shot-Gather Stacking 203

5.3.2 Automatic Gain Control 205

5.3.3 Principal Component Analysis 205

5.3.4 Feature Learning with Variational Autoencoders 210

5.3.5 Contrastive Representation Learning 213

5.4 Results and Discussion 220

5.5 Future Work 226

5.6 Summary 231

6 Conclusions **233**

6.1 Summary of Contributions 234

6.2 Future Work 236

6.3 Impact & Final Remarks 237

Bibliography **241**

Appendices **277**

List of Algorithms

1 Training a Simple CNN Model 23
2 Training Algorithm for VAEs 48
3 Training Algorithm for GANs — extracted from Goodfellow et al. (2014) 51
4 Training Algorithm for DDPMs — adapted from Ho et al. (2020) . . 77
5 Sampling Algorithm for DDPMs — adapted from Ho et al. (2020) . 77
6 Full-Waveform Inversion (FWI) Algorithm 108
7 Training Algorithm for LDMs - second stage model 168
8 Sampling Algorithm for LDMs 168
9 Training Algorithm for Conditional LDMs 200
10 Sampling Algorithm for Conditional LDMs 200
11 Automatic Gain Control with Least-Mean-Square (LMS) Adjustment 291

List of Tables

3.1 Performance metrics for the reconstruction problem of CelebA (Liu et al., 2015) with a Variational Autoencoder (VAE) using different loss functions (rows). Metrics include Mean Squared Error (MSE), Wiener Loss, Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018a), Multi-Scale Structural Similarity (MS-SSIM), Peak Signal-to-Noise Ratio (PSNR), and Fréchet Inception Distance (FID) (Heusel et al., 2017). The LPIPS metric employed a VGG-16 (Simonyan and Zisserman, 2014) network pre-trained on ImageNet (Deng et al., 2009), whereas the FID metric utilised features from an InceptionV3 (Szegedy et al., 2016) network, also pre-trained on ImageNet (Deng et al., 2009), following standard deep-learning practices. These metrics were generated using the 'torchmetrics' (Detlefsen et al., 2022) package. The Wiener Loss demonstrates superior fidelity, as indicated by the lowest MSE values, the highest MS-SSIM and PSNR, and competitive FID and LPIPS scores, outperforming other losses in terms of overall image quality and structural preservation. 147

3.2 Comparative metrics for the imputation problem of heavily-corrupted MRI scans using a variational autoencoder (VAE) with different loss functions (rows). The LPIPS (Zhang et al., 2018a) metric utilised a VGG-16 (Simonyan and Zisserman, 2014) network pre-trained on ImageNet (Deng et al., 2009), while the FID (Heusel et al., 2017) metric employed features from an InceptionV3 (Szegedy et al., 2016) network, also pre-trained on ImageNet (Deng et al., 2009). These metrics were generated using the *torchmetrics* package (Detlefsen et al., 2022). The Wiener Loss exhibits superior performance in terms of MSE and MS-SSIM, and shows competitive results in the LPIPS metric, outperforming traditional losses such as MSE, FFL and AAE, and

the bicubic interpolation. The model trained under the LPIPS metric shows better performance due to its sharper resolution. But overall, these metrics demonstrate the efficacy of the Wiener Loss in recovering high-fidelity details when imputing MRI scans. 152

4.1 Summary of hyperparameters that describe the four VAE models explored in this work for perceptual compression. 178

4.2 Summary of hyperparameters that describe the five LDM models explored in this work. 183

4.3 Fréchet Inception Distance (FID) (Heusel et al., 2017) and Kernel Inception Distance (KID) (Bińkowski et al., 2018) metrics over 128 generated samples of each experiment, compared to 1,000 samples of the original dataset. All metrics performed using features of a pre-trained InceptionV3 (Szegedy et al., 2016) network architecture, consistent with current common practices in generative deep learning. Sampling was obtained with the DDIM equations for reverse Markov chain for 25 steps, except for LDM-5-DDPM where the DDPM equations were used with 1,000 reverse steps. FID (ImageNet) and KID computed with the default parameters in `torch-fidelity` (Obukhov et al., 2020). FID (RadImageNet) computed with Monai Generative (Pinaya et al., 2023). 185

C.1 Hyperparameters used for contrastive embedding models. 297

List of Terms

AAE Adversarial Autoencoder.

AGC Automatic Gain Control.

AI Artificial Intelligence.

AWI Adaptive Waveform Inversion.

BERT Bidirectional Encoder Representations from Transformers.

BM Boltzmann Machine.

CLIP Contrastive Language-Image Pre-Training.

CLT Central Limit Theorem.

CNN Convolutional Neural Network.

CT Computed Tomography.

CVAE Conditional Variational Autoencoder.

DDIM Denoising Diffusion Implicit Models.

DDPM Denoising Diffusion Probabilistic Models.

DL Deep Learning.

DM Diffusion Model.

EBM Energy-Based Model.

EEG Electroencephalography.

ELBO Evidence Lower Bound.

FDTD Finite-Difference Time-Domain.

FFL Focal Frequency Loss.

FFT Fast Fourier Transform.

FID Fréchet Inception Distance.

FNN Feedforward Neural Network.

FWI Full-Waveform Inversion.

GAN Generative Adversarial Networks.

GMM Gaussian Mixture Models.

GPR Ground Penetrating Radar.

GPT Generative Pre-trained Transformer.

GPU Graphics Processing Unit.

HPC Human Connectome Project.

InfoNCE Information Noise Contrastive Estimation.

IS Inception Score.

JSD Jensen-Shannon Divergence.

KID Kernel Inception Distance.

KLD Kullback-Leibler Divergence.

KNN K-Nearest Neighbours.

LDM Latent Diffusion Model.

LDR Levinson-Durbin Recursion.

LPIPS Learned Perceptual Image Patch Similarity.

LS-GAN Least-Square Generative Adversarial Network.

LSM Least Mean Square.

MAE Mean Absolute Error.

MCM Markov Chain Model.

MEG Magnetoencephalography.

ML Machine Learning.

MLE Maximum Likelihood Estimation.

MMD Maximum Mean Discrepancy.

MRI Magnetic Resonance Imaging.

MS-SSIM Multi-Scale Structural Similarity Index Measure.

MSE Mean Square Error.

NLL Negative Log-Likelihood.

NLP Natural Language Processing.

ODE Ordinary Differential Equation.

PCA Principal Component Analysis.

PDE Partial Differential Equation.

PDF Probability Density Function.

PSNR Peak Signal-to-Noise Ratio.

ReLU Rectified Linear Unit.

RMSE Root Mean Square Error.

RMSProp Root Mean Square Propagation.

RNN Recurrent Neural Network.

SDE Stochastic Differential Equation.

SNR Signal-to-Noise Ratio.

SoS Speed of Sound.

SSIM Structural Similarity Index Measure.

UT Ultrasound Tomography.

VAE Variational Autoencoder.

VDM Variational Diffusion Models.

ViT Vision Transformers.

VLB Variational Lower Bound.

VQ-VAE Vector Quantized Variational Autoencoder.

List of Figures

2.1 Illustration of a simple feedforward neural network architecture with one input, one hidden and one output layer. Each node (circles) in the previous layer is connected to every node in the subsequent layer through a weighted sum, where weights (lines) are learned during training. Image obtained from Ahmadian and Khanteymoori (2015) 17

2.2 (a) Decision boundary of a simple neural network classifier with no hidden layers or activation functions. (b) Manifold representation of the data learned through hidden layers and activation functions in a neural network. The network learns to transform the data into a representation that is more linearly separable. Images obtained directly from Olah (2014) 19

2.3 Illustration of the convolution process in a neural network, using a filter of size 3×3 on an input feature of size 5×5, demonstrating how feature maps are generated. The blue grid represents the feature map from the input layer, while the grey area shows the filter or kernel sliding across this input feature map, capturing local patterns through weighted sums at each position. As the filter moves, it generates activations that form a new feature map in the next layer, shown progressively in green at each step. This convolution extracts spatial features from the input, creating a localised feature representation in the subsequent layer. Extracted from Dumoulin and Visin (2016). . . 20

2.4 PCA analysis of the MSE between MNIST and Gaussian noise. PCA decomposes the error into its principal components, where each component is an eigenvector of the covariance matrix of the error. The variances are the eigenvalues of the covariance matrix, indicating the importance of each principal component in the error. The plot shows the variances of the error for different principal components, highlight-

ing the prioritisation of low-frequency resolutions in MSE. Variance increases to the right. 34

2.5 The figure depicts the 1D Ackley function (Ackley, 1987), a non-convex objective with multiple local minima and a global minimum at $x = 0$. Local minima (orange) and the global minimum (red) are marked on the primary plot, while the inset zooms in on a local minimum near $x = 2.5$, depicting a narrower optimisation search space that local optimisation algorithms may explore. Under poor optimisation conditions (e.g. initialisations and regularisations), local optimisation algorithms, such a gradient-based ones, may be trapped in local minima, failing to reach the optimal solution. Black arrows indicate the direction of gradient descent at different points along the function. 40

2.6 Schematic representation of a Variational Autoencoder (VAE). The encoder network maps the input data $\mathbf{x}$ to the parameters of the approximate posterior that defines latent distribution $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, from which a sample $\mathbf{z}$ is drawn. The decoder network then reconstructs the input data from the latent sample. Extracted directly from Kebaili et al. (2023) — notation in the figure differs slightly from that used in the text. 47

2.7 Schematic representation of a Generative Adversarial Network (GAN). The generator network G maps a noise vector $\mathbf{z}$ to a data sample $G(\mathbf{z})$, while the discriminator network D evaluates whether a given sample is real or generated. The adversarial training process iteratively updates both networks to improve the quality of generated samples and the discriminator's ability to distinguish them from real data. Extracted directly from Kebaili et al. (2023) — notation in the figure differs slightly from that used in the text. 52

2.8 Evolution of the sampling process, represented by blue-dotted particles over time under Langevin dynamics, as described by Equation (2.50). Each panel represents successive timesteps, showing particles converging towards regions of high probability density. The vector field (black arrows) denotes the gradient of the log-density function,

directing particles towards modes of the target distribution. Red contours illustrate the energy landscape, with particles initially dispersed and progressively concentrating in lower-potential regions. Noise, modelled by the term $\sqrt{2\eta}\epsilon_k$, introduces stochasticity, ensuring exploration of the state space. Image adapted from Song (2021) 59

2.9 Progressive transformation of an initial unimodal Gaussian distribution into a bimodal target distribution over 1000 iterations, employing a cosine annealing schedule to control the iterative degradation. This process reflects the underlying principles of score-based and diffusion models, where probability distributions are iteratively refined by perturbing initial samples towards a more complex target distribution. The evolving distribution (solid line) increasingly aligns with the target distribution (dashed line), demonstrating the gradual improvement in density estimation characteristic of generative models that leverage forward and reverse diffusion processes. Image inspired by Song (2021). 60

2.10 Illustration of the forward (solid right arrows) and reverse processes (dashed left arrows) in Denoising Diffusion Probabilistic Models (DDPMs). The forward process gradually adds Gaussian noise to the data, transforming it into a standard Gaussian distribution, here represented by z. The reverse process, parameterised by a neural network, iteratively denoises the noisy data back to the original data distribution. Extracted directly from Kebaili et al. (2023) — notation in the figure differs slightly from that used in the text. 63

2.11 Transformer model architecture proposed byVaswani et al. (2017). The figure illustrates the transformer model architecture, comprising an encoder (left) and a decoder (right). The encoder consists of N identical layers, each containing a multi-head self-attention mechanism and a feed-forward network, both followed by residual connections and layer normalisation. Input embeddings are combined with positional encodings before being processed by the encoder. The decoder, similarly composed of N layers, includes an additional masked multi-head attention mechanism to prevent positions from attending

to subsequent positions. The final output probabilities are produced through a linear transformation followed by a softmax function. . . . 82

2.12 Comparison of the spatial resolution of a breast ultrasound image reconstructed using ray-based tomography (a, c) and FWI (b, d). (c) and (d) are zoomed in versions of (a) and (b), respectively. The FWI recovered image shows a higher level of detail and resolution, capturing fine anatomical structures that are not visible in the ray-based image. Modified after Calderon Agudo et al. (2018). 95

2.13 A series of shot gathers illustrating the FWI process. A starting model lacking the complex heterogeneities of the ground truth model is iteratively updated until the simulated data closely matches the observed data. The images depict one shot gather of the (a) observed data, (b) modelled data of the starting model, and (c) modelled data of the final model recovered by FWI. The shot gathers are represented as an amplitude map of the waveforms recorded over time (y-axis) at each receiver location (x-axis). Modified after Warner et al. (2013) 97

2.14 Cycle-skipping in FWI. The inversion process is susceptible to cycle-skipping when the phase of the synthetic data is misaligned with the observed data by more than half a wavelength. The solid black line indicates a monochromatic waveform with a period T as a function of time. The upper dashed line shows the modelled monochromatic waveforms with a time delay exceeding $T/2$. In this scenario, FWI will adjust the model so that the $(n+1)^{\text{th}}$ cycle of the modelled waveforms aligns with the n^{th} cycle of the observed waveform, misrepresenting the kinematics of the system. In the lower example, FWI modifies the model such that the modelled and recorded n^{th} cycles remain in-phase, as the time delay is less than $T/2$. Figure extracted from (Virieux and Operto, 2009). 109

3.1 Top: visual illustration of two-dimensional Wiener filters that match MNIST sample $\mathbf{y}$ to random noise $\mathbf{x}$. Bottom: matching $\mathbf{y}$ to $\mathbf{x}$ where $\mathbf{x} = \mathbf{y}$, demonstrating that the Wiener filter when $\mathbf{x} = \mathbf{y}$ is given by the Dirac delta function. During Wiener Loss optimisation, the whitening matrix $\mathbf{W}$ iteratively penalises the residuals between the

Wiener filter and zero-lag spike, driving it towards a Direct delta function, and $\mathbf{x}$ and $\mathbf{y}$ towards identity. 123

3.2 Average gradient of 2D-AWI and 2D-Wiener Loss objectives for a Variational Autoencoder (VAE) model successfully trained with both methods. The model gradients under the AWI objective increase at near the minimum, while model gradients under the Wiener Loss decrease, making the latter more befitting to most standard deep learning optimisers. 128

3.3 Comparison of and estimation of the loss landscapes of a VAE model successfully trained to reconstruct MNIST digits (LeCun et al., 1998) with 2D-AWI (left) and 2D-Wiener Loss (right). For convergence, the 2D-model required a Delta-like penalty function ($\mathbf{T} \approx \boldsymbol{\delta}$), whereas the Wiener Loss achieved superior results with $\mathbf{W} = \boldsymbol{I}$. The steep gradients required in AWI create challenging conditions for optimisation, while the loss landscape of the model trained with 2D-Wiener Loss resembles the smooth surface of the MSE. The visualisation, achieved with the method described in Li et al. (2018), represents a two-dimensional parametrisation of a the optimised multi-million parameter model. 128

3.4 Identity (top) and Laplace (bottom) penalty functions along with their corresponding whitening matrices ($\mathbf{W}$) is illustrated. Both approaches assume decorrelation in the residuals, but show different self-correlations. The Identity function has unit variance and suppresses residuals uniformly, whereas the Laplace function imposes higher variance at far offsets, emphasising near-zero offset residual suppression. 133

3.5 Reconstruction of MNIST digits using a variational autoencoder (VAE) from the validation batch. The top row represents the reconstruction using Mean Squared Error (MSE) loss, the middle row shows the reconstruction using Wiener Loss, and the bottom row displays the ground truth (GT) images. The implementation of the Wiener Loss demonstrates superior performance in recovering accurate and high-frequency details, as evidenced by the sharper edges of the digits compared to the MSE counterpart. 136

3.6 Frequency domain representations of the reconstructed MNIST digits in Figure 3.5. The top row displays the real Fast Fourier Transform (FFT) of the digits reconstructed using Mean Squared Error (MSE) loss, the middle row shows the real FFT using Wiener Loss, and the bottom row presents the real FFT of the ground truth (GT) images. The Wiener Loss captures a wider range of frequencies than the MSE loss, and is a more accurate reflection of the GT spectrum. 137

3.7 Progressive reconstructions of MNIST digits using MSE (top) and Wiener Loss (bottom) models, highlighting the network's initial biases towards low frequencies. Columns represent progressive epochs from left to right. The MSE-based model quickly produces recognisable but blurry digit reconstructions, lacking finer details and high-frequency components even at convergence. In contrast, the Wiener Loss model initially shows significant amplitude mismatches due to prioritising phase alignment but rapidly fits low-to-mid frequencies, gradually improving towards higher frequencies that the MSE fails to achieve. 138

3.8 Evolution of the Wiener filters for a batch of 64 reconstructed MNIST digits (validation set) across different epochs. The filters have been flattened for better visualisation. Initially, the model focuses on suppressing non-zero lag energy to align the phases of the data at different frequencies. As the training stabilises, it begins to address the amplitude of the spike and further suppresses these non-zero lag energy values, demonstrating a refined convergence towards accurate frequency alignment and amplitude correction over time. 139

3.9 Effect of varying λ and γ on MNIST digit reconstructions using the Wiener Loss, demonstrated with samples from the validation dataset. The left panel illustrates the effect of varying λ while keeping γ fixed at 0.25, showing that low λ values hinder convergence due to numerical instabilities, while high λ values lead to blurry reconstructions. The right panel demonstrates the effect of varying γ with λ fixed at 7, indicating that low γ values slow convergence and reduce high-frequency data fitting, whereas high γ values disrupt convergence entirely. . . 140

3.10 Reconstruction of CelebA (Liu et al., 2015) images using a variational autoencoder (VAE) trained with different loss functions. The first row shows the ground truth, or target, image. Each following row represents the model trained with different losses (LPIPS, Wiener Loss, AAE, FFL, and MSE), and each column shows a different sample from the validation set. The Wiener Loss model outperforms all other models in terms of resolution, except for LPIPS. However, the Wiener Loss model demonstrates superior fidelity in preserving the original image details compared to all other models. 143

3.11 Evolution of the 2D trainable penalty matrix for the Wiener Loss experiment in reconstructing CelebA (Liu et al., 2015) images, illustrated in Figure 3.10. The rows represent different colour channels (R: top, G: middle, B: bottom), while the columns correspond to epochs of training. Colours indicate the percentage drop in values of the penalty matrix from its initial state. As training progresses, penalisation intensifies around the zero-lag axes and the vicinity of the zero-lag point, reflecting the model's efforts on recovering high-frequency components. 144

3.12 Comparison of Wiener filters (bottom) between random Gaussian noise and low-passed versions of itself, at different frequencies (top). The level of high and middle-frequency degradation increases to the right. The low frequency images are generated using a Butterworth low-pass filter of order 2, and the frequencies are normalised in the range [-0.5, 0.5]. The top row displays the filtered images at different frequency caps, while the bottom row shows the normalised 2D Wiener filters between the original and low-passed images, zoomed at zero lag for better visualisation. The blurrier the image, the more concentrated the filter energy becomes at the zero-lag axes and at the radial vicinity of the zero-lag point. 145

3.13 Outputs of a variational autoencoder (VAE) trained to impute heavily corrupted T1-weighted MRI images, with the corruption shown in the third row. The first row presents the ground truth (GT) images from the validation set, and the second row provides a zoomed version of the GT for better visualisation. Subsequent rows display the VAE out-

puts using different loss functions: LPIPS, Wiener Loss, Adversarial Autoencoder (AAE), Focal Frequency Loss (FFL), Mean Squared Error (MSE), and a simple bicubic interpolation for benchmarking. The MSE and FFL models show minimal differences and fail to recover fine details, resulting in blurry outputs. While the LPIPS model excels in capturing high-frequency details, the Wiener Loss outperforms it in preserving the fidelity of finer anatomical structures such as the ventricles and fissures of the brain, which appear brighter in T1-weighted images. 149

3.14 Optimised 2D penalty matrix of the final training epoch for the Wiener Loss experiment in the imputation of MRI scan images, as illustrated in Figure 3.13. Colours indicate the percentage drop in values of the penalty matrix from its initial state. The evolution of the penalty closely resembles the one presented in Figure 3.10, for the face reconstruction experiment, although for this example the percentage drop of $\sim 50\%$ is much larger. The cross-like structure at the zero-lag axes and the ring-like structures surrounding the zero-lag point correspond to the near-zero energy in the filters of the low-passed images shown in Figure 3.12. This highlights the penalty's substantial effort to recover high-frequency components. 150

3.15 Schematic illustrating that the Wiener filter between a signal and a translated copy of itself results in a Dirac delta function, translated in the same manner. This property is exploited to formulate a translation invariant version of the Wiener Loss. 153

3.16 Image classification task to use the MNIST training data (left) to predict the affNIST testing data (right). A KNN classifier with the Manhattan distance metric achieves and accuracy of **14%**, whereas a KNN classifier with the proposed translation invariant metric $\mathcal{L}_{\text{TI}}$ achieves an accuracy of **61%** 154

3.17 Confusion matrices for comprehensive multi-class performance evaluation of the baseline KNN classifier (left), vs the translation invariant KNN (right). Performance is substantially improved using the proposed translation invariant metric $\mathcal{L}_{\text{TI}}$. The Wiener filters in Equation

3.12 were computed in the Fourier domain in the `reverse` mode, with a 2-dimensional filter twice the size of the input image, and absolute pre-whitening $\lambda = 100$. Extracted from Cruz et al. (2023a) 154

3.18 Latent samples generated using the Wiener diffusion, visualised using the decoder, and an illustration of the diffusion process alongside the 30 closest Wiener filters to the sample $\mathbf{x}_t$ (as defined by 3.15) for MNIST (LeCun et al., 1998) (top) and CelebA (Liu et al., 2015) (bottom). 156

4.1 While standard diffusion models perform perceptual and semantic compressions simultaneously within the latent variables of the Markov chain, in latent diffusion models these tasks are separated: a first-stage training of an autoencoder performs perceptual compression, leaving the diffusion model to only perform the semantic compression. Image extracted from Kreis et al. (2023). 165

4.2 Schematics of a standard variational autoencoder, where the latent sample $\mathbf{z}$ is drawn from the latent prior modelled by the probabilistic encoder $\mathcal{E}_\theta(\mathbf{x})$ for an image $\mathbf{x}$, parameterised as a Gaussian with mean μ and standard deviation σ. The latent space is regularised through the Kullback-Leibler (KL) divergence to a standard Gaussian distribution (Equation 2.31). This regularisation, combined with the perceptual compression of the image, stabilises the learned diffusion process in the second stage of LDM training. The latent sample is subsequently mapped back to the data space by the probabilistic decoder $\mathcal{D}_\phi(\mathbf{z})$, which computes the posterior and reconstructs the perceptual details of the image. 166

4.3 Schematics of a latent diffusion model as proposed by Rombach et al. (2022). A sample is encoded to its latent representation by the trained probabilistic encoder, serving as the first sample of the diffusion process chain. The latent sample is then progressively degraded towards Gaussian noise through the forward diffusion process (Equation 2.59), following the stochastic differential (SDE) trajectories. A regularised latent space ensures smoothness in the probability flows that define these trajectories. For image generation, only the reverse diffusion

process is used by applying the reverse Markov Chain of the SDE to random gaussian noise (Equation 2.91), which is then fed to the trained decoder for perceptual upsampling towards the data space. Adapted from Kreis et al. (2023). 167

4.4 Dive into the denoising attention U-Net (Ronneberger et al., 2015) that is trained to predict the reverse diffusion process of the forward SDE. The model takes as input the sample degraded by $\boldsymbol{\epsilon}$ at time step t, defined by the noise scheduler, along with the sinusoidal positional embedding vector of the associated time step. The model then learns to predict the degradation noise $\tilde{\boldsymbol{\epsilon}} = \boldsymbol{\epsilon}_\psi(\mathbf{z}_t, t)$, at each t, by minimising the Euclidean distance $||\boldsymbol{\epsilon} - \tilde{\boldsymbol{\epsilon}}||_2^2$. Once trained, the model's predicted noise is subsequently used to estimate the previous sample of the diffusion process. Figure adapted (Kreis et al., 2023) and (Rombach et al., 2022). 170

4.5 T1-weighted MRI volumetric samples from the Human Connectome Project (HCP) Young Adult database (Van Essen et al., 2012), viewed in the mid-plane of the axial (top), coronal (middle), and sagittal (bottom) axes. 171

4.6 Volumetric speed of sound (SoS) samples, measured in m/s, from the dataset released by Strong (2023). Viewed in the mid-plane of the axial (top), coronal (middle), and sagittal (bottom) axes of the brain. 172

4.7 Density histograms of 128 volumetric samples of the (a) T1-weighted MRI HPC (Van Essen et al., 2012) dataset, (b) converted and min-max normalised speed of sound dataset by Strong (2023), and (c) the logarithmic processed speed of sound used in this work. The logarithmic processing of the samples is a reversible operation that gently helps equalising the SoS values. 173

4.8 Volumetric speed of sound (SoS) samples, measured in m/s, from the dataset released by Strong (2023) after the preprocessing pipeline. Viewed in the mid-plane of the axial (top), coronal (middle), and sagittal (bottom) axes of the brain. The preprocessing gently boosts the values of the intra-cranial tissues while gently decreasing the values of the skull. 174

4.9 Reconstruction of axial slices with associated root mean square errors (RMSE) from (a) VAE-1, (b) VAE-2, (c) VAE-3, (d) VAE-4 and (e) is the ground truth. Reconstructions from VAE-1 lack high-frequency details as no perceptual losses are integrated durings training. Reconstructions from VAE-2 fail to converge due to the Wiener loss' inability to fit the data at high value contrasts; reconstructions from VAE-3 have high perceptuality but the tight latent space bottleneck negatively impacts the fidelity of the images; reconstructions from VAE-4 are the most accurate in the semantic and perceptual recoveries. 179

4.10 Extracted feature maps (b) of a single neuroacoustic image sample of the brain from the VGG-16 (Simonyan and Zisserman, 2014) network (a) pretrained on ImageNet (Deng et al., 2009) dataset of 1,000 classes of natural images. Despite the difference in dataset nature, the pretrained network is able to select various useful features of the image, highlighting high-frequency textures between the soft tissues and isolating outer and intra-cranial structures. The choice of visualisation of the mid-sagittal plane is arbritary, and my experiments show that the same principles hold for the mid-axial and mid-coronal planes. Neural network visualisation made with (Lenail, 2024). Figure inspired by Strong (2023). 181

4.11 Normalised latent spaces of VAE-1 (a), VAE-2 (b), VAE-3 (c) and VAE-4 (d) as described in Table 4.1 and with associated reconstructions in Figure 4.9. For the multi-channel latent spaces (a, b and d) the presented visualisation is averaged in the channel dimension. . . 182

4.12 Example of samples generated from the five different LDM models (columns) explored in this work in the mid-plane of the axial (top row), sagittal (middle row), and coronal (bottom row) axes. All models were samples with a DDIM scheduler at 25 steps, except for (e) where the samples were generated with a DDPM scheduler at 1,000 steps. While LDM-1 to 4 fail to generate structurally accurate brains, LDM-5, sampled with either DDIM or DDPM are able to construct realistic-looking neuroacoustic brain samples. 184

4.13 (a) FID and (b) sampling time (b) per number of sampling steps

averaged for 8 samples generated from LDM-5 using DDPM (△) and DDIM (○). Colors represent different anatomical mid-planes. While DDPM reaches best metrics at >750 (∼30s) sampling steps, DDIM is able to achieve similar metrics starting at 10 (∼0.6s) sampling steps, showing exceptionally better efficiency. Samples were generated with a single NVIDIA GeForce RTX 2080 Ti 11 Gb card. 187

4.14 Samples generated from LDM-5 using DDPM (a-c) and DDIM (d-f) at different number of sampling steps (columns). While sampling with DDPM can only synthetise high-quality samples at >750 steps, sampling with DDIM achieves the equivalent quality with <30 steps. 188

4.15 Examples of generated neuroacoustic brain samples from LDM-5-DDIM with 25 sampling steps for the mid-sagittal plane of the brain. 192

4.16 Examples of generated neuroacoustic brain samples from LDM-5-DDIM with 25 sampling steps for the mid-axial plane of the brain. 193

4.17 Examples of generated neuroacoustic brain samples from LDM-5-DDIM with 25 sampling steps for the mid-coronal plane of the brain. 194

5.1 Dive into the denoising attention U-Net (Ronneberger et al., 2015) that is trained to predict the reverse diffusion process of the forward SDE. The model takes as input the sample degraded by $\boldsymbol{\epsilon}$ at time step t, defined by the noise scheduler, along with the sinusoidal positional embedding vector of the associated time step, and the embedding $\tilde{\mathbf{c}} = \mathcal{T}_\chi(\mathbf{c})$ of a condition $\mathbf{c}$. The embedded condition is inserted by either concatenation to $\mathbf{z_t}$ in the channel dimension, or by cross-attention in the attention blocks. The model then learns to predict the degradation noise $\tilde{\boldsymbol{\epsilon}} = \boldsymbol{\epsilon}_\psi(\mathbf{z}_t, \tilde{\mathbf{c}}, t)$, at each t, by minimising the Euclidean distance $||\boldsymbol{\epsilon} - \tilde{\boldsymbol{\epsilon}}||_2^2$. Once trained, the model's predicted noise is subsequently used to estimate the previous sample of the diffusion process, given an unseen condition embedding $\tilde{\mathbf{c}}$. Figure adapted (Kreis et al., 2023) and (Rombach et al., 2022). 199

5.2 Hypothetical acquisition setup displayed on the mid-axial plane of a brain sample, showing the distribution of speed of sound (SoS) across the medium in m/s. The spatial grid used consists of 480×480 cells, each with a spacing of 0.5 mm, capturing detailed structural informa-

tion within the brain and skull. The yellow markers represents the receiver locations, while the purple markers indicate source positions around the perimeter, outlining the circular transducer arrangement for ultrasound acquisition. 202

5.3 Source wavelet (left) and its frequency spectrum (right) used for ultrasound modelling. The waveform is a three-cycle tone burst with a peak amplitude occurring at 500 kHz, covering a frequency bandwidth of approximately 100 kHz to 1 MHz. 203

5.4 Synthetic ultrasound shot gather generated from forward modelling on an acoustic velocity map of a brain, illustrating key features of wave propagation through brain and skull structures. The shot amplitude (colour scale) is normalised to the range ± 1 using a min-max scaling approach and visualisation is clipped to range $\pm$ 0.01 to increase contrast. Major wave components are annotated, including direct arrivals through water, near-source and far-source reflections from the skull, transmission paths through brain tissue, and guided waves through skin and subcutaneous fat. 204

5.5 Stacked shot gathers from 32 sources for a single brain sample. (a) The raw stack, displaying dominant near-source direct arrivals (propagating in water) and obscured intra-cranial transmissions. (b) The stack with processed with LSM-AGC, brightening intra-cranial transmissions, guided waves, and skull reflections while suppressing near-source direct arrivals. 206

5.6 Histograms of the normalised amplitude distributions for the stacked shot gathers. (a) Raw stack, showing a narrow distribution of values centred around zero, indicating limited amplitude variation without gain adjustments. (b) Stack processed with LSM-AGC, which results in a broader amplitude distribution with increased variability. 207

5.7 Application of PCA to the ultrasound dataset, with each sample flattened into a 1D vector. The left plot shows cumulative explained variance η^2 against the number of principal components on a logarithmic scale, showing that 512 dimensions are sufficient to explain the data ($\eta^2 = 0.998$). The images on the right depict principal com-

ponents $N \leq 50$ learned by the PCA fitting, reshaped to match the original data dimensions, showing high variability in transmissions and reflections. 209

5.8 PCA applied along the offset axis of the AGC-stack ultrasound dataset, where each trace is treated as an independent data point, reducing each sample from $\mathbb{R}^{1000\times2500}$ to $\mathbb{R}^{128\times2500}$. The left plot displays the mean of the reduced forms across all samples, with each point in the x-axis reflecting the average principal component value for a given time step. The right plot, a variance map, shows pixel-wise variance across the reduced samples, revealing areas of high variability. High variance is observed in the skull near-source reflections and guided waves of propagation through skin (time-steps <1500, ∼120 μs), and in intra-cranial transmissions around time-step 1800 (∼144 μs), corresponding to critical regions of acoustic variability, showing that the reduction provides strong statistical discrimination precisely in these essential areas. Sample-averaged cumulative explained variance $\eta^2 = 0.994$. . 211

5.9 PCA applied along the time axis of the ultrasound dataset, where each trace is treated as an independent data point, reducing each sample from $\mathbb{R}^{1000\times2500}$ to $\mathbb{R}^{1000\times128}$. The left plot displays the mean of the reduced forms across all samples, with each position reflecting the average principal component value across transducers. The right plot, a variance map, shows pixel-wise variance across the reduced samples, indicating areas of higher variability. Unlike previous methods, this approach does not reveal clear structural patterns in the mean plot, making interpretability more challenging. However, the variance map shows a higher spread of variance across different areas, which could be beneficial for statistical discrimination and feature identification. Sample-averaged cumulative explained variance $\eta^2 = 0.996$. 212

5.10 A statistical summary of the latent samples from a VAE trained with a U-Net (Ronneberger et al., 2015) architecture on ultrasound data stacks after LSM-AGC, using an L2 reconstruction loss and KL regularisation of the latent space. The mean of the latent samples (left) reveals structural patterns that reflect the preservation of spatial and

temporal information from key aspects of the stack, while the variance map (right) highlights regions of high variability, including initial skull reflections and intra-cranial transmissions. This demonstrates the model's ability to capture and differentiate structural variations in the data within a compressed space, offering a more interpretable representation than PCA-based methods. The statistics are derived from 200 latent samples from the validation set. 213

5.11 Logits of the contrastive loss for the shared embedding space of 256 AGC-stack and image samples. The matrix on the left represents data-to-image alignment, while the matrix on the right corresponds to image-to-data alignment, reflecting the two terms in Equation 5.7. High values along the diagonal represent correctly aligned data-image pairs, while low values on the off-diagonals correspond to negative pairs, indicating cross-modal alignment within the learned latent space. 217

5.12 Two-dimensional projections of the embeddings of 512 paired samples using PCA (left) and UMAP (right) to visualise the alignment process. In the top row, raw data (red) and image (blue) representations are projected into 2D space as a baseline. In the bottom row, aligned embeddings of data and images are shown, with the pairwise average Euclidean distances ($\bar{\delta}$) calculated to quantify alignment. The drastic reduction in pairwise distances in the embedding space demonstrates that the network successfully aligned data and image representations in a shared latent space. The UMAP optimisation used is parameterised a minimum distance of 0.1, 15 neighbours, Euclidean metric, 2 components, and a learning rate of 1, with 512 samples. 218

5.13 Demonstration of the continuity of the latent space in contrastive learning, where brain images with similar anatomical structures and their associated AGC-stacks are embedded closer together, while more distinct samples are positioned further apart. The figure compares three samples with varying cosine similarities (sim) to a base sample (a), displaying a high (b), medium (c), and low (d)similarity. The average relative mean squared error ($\bar{\delta}_{i,j}$) increases as the cosine sim-

ilarity decreases. This spatial arrangement within the latent space facilitates structured alignment of similar samples, where the latent disentanglement represents meaningful relationships across image and data. 219

5.14 Demonstration of the network's potential for clinical generalisation by visualising cosine similarity of embeddings under rigid transformations of translation and rotation. In (a), cosine similarity is plotted against translations, showing that similarity remains high up to approximately 30 mm, beyond which it gradually declines. In (b), cosine similarity is shown relative to rotational angles, with notable similarity retained up to ∼10 degrees. Interestingly, cosine similarity decreases linearly for rotations between 10 and 45 degrees but increases for rotations between 45 and 90 degrees, suggesting the model naturally aligns small and near-perpendicular transformations. These results suggest that the latent space continuity enables the model to maintain alignment between similar brain embeddings under moderate translations and rotations, which can be beneficial in clinical applications. 220

C.1 Visualisation of the PCA projections for both data and image embeddings across different models. The red dots represent data embeddings, while the blue dots represent image embeddings. The projections are plotted for the raw image and data (first plot) and for the embeddings from models GF-223, SG-222, WL-206, SS-205, ZM-203, DF-174, LT-154, and CR-63, with their respective average pair-wise Euclidean distances ($\bar{\delta}$) indicated in parentheses. Distances between data and image embeddings within the projections illustrate how the model encodes similarities between the two modalities. 295

C.2 Visualisation of the UMAP projections for data and image embeddings with a neighbourhood size of 15, Euclidean metric, and two components. The red dots represent data embeddings, while the blue dots represent image embeddings. The projections are plotted for the raw image and data (first plot) and for the embeddings from models GF-223, SG-222, WL-206, SS-205, ZM-203, DF-174, LT-154, and CR-63, with their respective average pair-wise Euclidean distances ($\bar{\delta}$)

indicated in parentheses. Together with the PCA visualisation, the UMAP projections provide a comprehensive view of the alignment between the embeddings from different models. 296

Chapter 1

Introduction

1.1 Imaging the Human Brain

The human brain is an organ responsible for our cognitive functions, emotions, and overall control of the body, representing one of the most intricate structures in animal biology. It has long been a subject of fascination and study, and despite extensive research, it remains marked by significant unknowns, continuing to intrigue and challenge the scientific community. Understanding its structure and function is not only vital for advancing our general knowledge of human cognition, but it has become an essential tool for neurological pathology diagnosis and treatment (Wang et al., 2020). This need has driven the development of various neuroimaging techniques, where structural information about the brain is obtained through non-invasive methods, offering unique insights into brain anatomy and activity.

Tomography is one of the closest methods we have to capturing an image of the human brain without the need for surgical intervention. Much like a camera translates light into a photograph, tomography is the process of transforming signals from the body into measurable, structured representations of its internal physical properties. In Magnetic Resonance Imaging (MRI), tomography is performed by detecting signals from protons in response to magnetic fields; in X-ray Computed Tomography (CT), by measuring x-ray attenuation across tissues; in Positron Emission Tomography (PET), by tracking radiotracers that reveal metabolic activity; and in Ultrasound Tomography (UT), by analysing propagated sound waves to measure

tissue-specific acoustic variations. However, unlike photographs, tomographic images are not direct representations of the body, but rather complex reconstructions that rely on sophisticated, physics-based approaches to interpret and process these signals into meaningful images, parameterised by the medium's physical properties. This is known as the inverse problem.

High-quality images of the brain are of essential value in clinical applications. They enable detailed visualisation of brain structures and functions, allowing for detection and characterisation of abnormalities such as tumours, lesions, and neurodegenerative diseases. Early-stage imaging enables the timely detection and monitoring of these pathologies, which can lead to better treatment planning and patient outcomes. Precise imaging also enables neurosurgeons to plan and execute procedures with greater accuracy, minimising the risk of damage to critical brain regions. Yet, no single imaging modality can comprehensively address all informational needs for understanding the brain's complexity. Neuroimaging thus encompasses a range of modalities, each with distinct advantages and constraints. In this field, MRI is considered the gold standard for brain imaging due to its superior soft tissue contrast and its ability to map both neuroanatomical structures and neuroactivity. However, MRI remains a costly and non-portable procedure, requiring longer acquisition times, and may not be suitable for patients with metal implants or those who experience claustrophobia. X-ray CT, in contrast, provides rapid high-resolution imaging of brain structures, particularly useful in emergency situations, though it is limited by lower soft tissue contrast and requires exposure of patients to ionising radiation (Salmon et al., 2015). Meanwhile, PET allows for the visualisation of metabolic processes in the brain but lacks spatial resolution compared to CT or MRI; it also involves radiation exposure, and is relatively expensive (Salmon et al., 2015). Electroencephalography (EEG) and Magnetoencephalography (MEG) are also important imaging techniques that provide high temporal resolution for studying brain activity, but are not designed to capture spatially structured details (da Silva, 2013).

In this context, ultrasound imaging stands out as a versatile complement to these traditional modalities, offering unique advantages that make it particularly suitable for fast, accessible, and safe imaging. Unlike MRI and CT, UT equipment

is portable and cost-effective, allowing it to be deployed outside specialised imaging facilities, such as in emergency departments, ambulances, and patient rooms. UT also offers fast acquisition and imaging times, making it the preferred modality for real-time imaging applications, early detection and prevention of pathologies, and in cases where immediate diagnoses and live monitoring of time-sensitive conditions are essential (e.g. surgical procedures, critical trauma cases) (Beck-Razi et al., 2007; Arnold et al., 2020). Notably, unlike CT, UT avoids the use of ionising radiation, providing a safer option for repeated imaging, especially for vulnerable populations that require frequent monitoring.

Despite its practical advantages, conventional ultrasound tomography falls short of the spatial resolution and fidelity achievable with MRI and CT and faces methodological limitations that prevent its applicability in brain imaging (von Ramm et al., 1978). The inherent complexity and non-linearity of wave propagation in ultrasound pose a formidable computational barrier to image reconstruction, making the inverse problem challenging to accurately resolve. To circumvent this, these algorithms often adopt ray-based approximations, reducing wave propagation to a simplified model that enables real-time imaging. However, it is due to these crude approximations that conventional ultrasound is limited in resolution and fidelity.

Brain imaging with conventional ultrasound is further complicated by the skull's strong acoustic contrast with soft tissues, which obscures signals from intra-cranial structures. Due to this contrast, signal responses from intra-cranial structures are predominantly represented by transmission waves, which are discarded in ray-based models (Guasch et al., 2020; Calderon Agudo, 2018; Strong, 2023). Consequently, conventional ultrasound imaging is practically unachievable for imaging the soft-tissue structures of the brain, except in infants whose skulls have not yet completely formed and have smaller acoustic contrasts.

Recent advancements in medical ultrasound tomography have introduced full-waveform inversion (FWI) as a promising technique to image the adult brain (Guasch et al., 2020; Calderon Agudo, 2018; Strong, 2023; Cueto et al., 2022; Cudeiro-Blanco et al., 2022; Bates et al., 2022). Unlike ray-based approaches, FWI utilises the complete waveform of the data, allowing the tomography process to be a more accurate

representation of the imaged medium. More specifically, FWI captures complex wave interactions, including transmission waves and phenomena such as elasticity, scattering, and attenuation, which are otherwise neglected in ray approximations but are crucial for resolving fine-scale structures (Tarantola, 1984) and intra-cranial tissues (Guasch et al., 2020). FWI has not only enabled remarkable improvements in the resolution of soft-tissue ultrasound tomography (Pratt et al., 2007; Calderon Agudo, 2018), but, for the first time, it has made brain imaging possible using ultrasound data alone (Guasch et al., 2020; Calderon Agudo, 2018; Strong, 2023). This advancement achieves resolutions comparable to MRI, positioning FWI as a promising alternative to conventional ultrasound imaging. By combining high resolution and fidelity with the inherent portability, safety, and cost-effectiveness of ultrasound, FWI has the potential to revolutionise emergency care for conditions such as stroke or traumatic brain injury (Guasch et al., 2020; Calderon Agudo, 2018; Strong, 2023).

But despite these promising developments, challenges remain in applying FWI in medical problems, especially in rapid clinical settings. FWI is known for its computational complexity, particularly when high-resolution 3D models are required. FWI is also reliant on *a priori* information for the inversion process and is sensitive to noise, data artefacts, and limitations of clinical data acquisitions (Guasch et al., 2020; Calderon Agudo, 2018; Strong, 2023). In brain imaging, where the skull's high acoustic contrast considerably weakens intra-cranial transmissions, these challenges are especially acute.

1.2 Deep Learning: from classifiers to creators

The earliest developments of neural networks date back to the 1940s, when McCulloch and Pitts (1943) introduced the first mathematical model of a neuron inspired by the brain's biological structure. This model laid the foundation for the perceptron (Rosenblatt, 1958), a simple neural network capable of learning linear decision boundaries for binary classification. However, the perceptron's limitations in handling non-linear data precipitated a period of disillusionment in the 1970s, known today as the "AI winter", wherein neural networks faced criticism for their perceived

inability to address realistically complex problems (Crevier, 1993). It was not until the 1980s that neural networks experienced a resurgence, driven by the increasing accessibility of computational resources and the re-introduction of the backpropagation algorithm by Rumelhart et al. (1986). This expansion enabled the training of a multi-layer perceptron, demonstrating that neural networks could indeed solve non-linear tasks, marking a turning point in the field. It was in the late 1990's, however, with the introduction of convolutional neural networks (CNNs) by LeCun et al. (1998), that neural networks regained real momentum. CNNs, designed to leverage spatial hierarchies in data, established new performance benchmarks in image classification and marked the beginning of modern AI.

With the concurrent and fast rise of computational power the availability of large data, neural networks grew in size and complexity, and so did their processing capabilities, defining the era of deep learning (DL). Today, deep learning stands as one of the most rapidly advancing fields in scientific research, encompassing applications that extend beyond traditional computer science domains. Beyond image and speech recognition, deep learning drives innovation in fields such as drug discovery, financial modelling, geological exploration, and even creative realms like art and music generation. This evolution, from early classification tasks to sophisticated creative applications represents a major reshaping of both scientific and cultural landscapes.

Deep learning gained prominence in the 2010s, largely due to breakthroughs in computer vision and natural language processing (NLP). Early models such as AlexNet (Krizhevsky et al., 2012) revolutionised image classification by leveraging convolutional neural networks to achieve inconceivable accuracy on the ImageNet dataset (Krizhevsky et al., 2012), inspiring subsequent developments in CNN-based architectures. VGGNet (Simonyan and Zisserman, 2014) and ResNet (He et al., 2016) continued to push accuracy through increasingly deep and complex network designs, demonstrating the ability of neural networks to capture hierarchical image features, from simple shapes to intricate textures. In NLP, recurrent neural networks (RNNs) (Rumelhart et al., 1986) and later, long short-term memory (LSTM) networks (Hochreiter and Schmidhuber, 1997), enabled language models to retain context over long sequences, paving the way for tasks like machine translation and

sentiment analysis.

The shift from neural classifiers to creators was further propelled by advances in representation learning (also known as feature learning), enabling models to abstractly encode underlying data structures in the shape of multidimensional latent variables. Autoencoders (Hinton and Zemel, 1993) were initially developed for dimensionality reduction, yet their latent representations proved invaluable in data synthesis tasks. As an extension, Variational Autoencoders (VAEs) (Kingma and Welling, 2013) introduced a probabilistic approach to generation, allowing the creation of new data points by sampling from learned distributions. By capturing complex data distributions, VAEs marked a crucial step toward the generative potential of deep learning, setting the stage for a new era of creativity in AI.

One of the most transformative advancements in deep learning emerged with the introduction of Generative Adversarial Networks (GANs) by Goodfellow et al. (2014). GANs employ a generator and a discriminator engaged in an adversarial minimax game, where the generator produces synthetic samples while the discriminator distinguishes between real and generated data. This novel framework allowed GANs to synthesise highly realistic samples, surpassing the capabilities of prior generative models like VAEs. GANs have since found widespread applications, from image synthesis to audio generation, revolutionising creative industries and establishing new benchmarks for generative artificial intelligence.

More recently, the development of Transformers (Vaswani et al., 2017) further expanded the generative landscape of deep learning. Initially designed for NLP tasks, Transformers eliminated the sequential limitations of RNNs by proposing a highly-parallelisable architecture consisting of self-attention mechanisms. This architecture enabled large-scale language models like BERT (Devlin et al., 2019a) and GPT (Radford et al., 2018) to generate coherent and contextually relevant text, transforming the filed of NLP and opening doors to creative writing and automated content generation. Notably, GPT-3 (Brown et al., 2020) demonstrated that deep learning could produce human-like text, reinforcing the potential of AI as a generative agent. Today, DL continues to evolve with advancements like diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Rombach et al., 2022), which have shown remarkable

capabilities in inductive biases, synthesising high-fidelity and high-resolution data samples.

The vast scope of deep learning research has resulted in an extraordinary volume of publications, with new breakthroughs emerging regularly. The influence of AI models is now recognised at the highest levels of scientific achievement, showcased by the 2024 Nobel Prizes in Chemistry and Physics, both awarded for advancements driven by deep learning methodologies. While still maturing as a discipline, deep learning has already reshaped how we interact with technology and perceive its possibilities, firmly establishing its role in the future of scientific innovation.

1.3 Motivation and Contributions

In light of the inherent challenges in brain ultrasound imaging and full-waveform inversion, and the proven advantages of deep learning in complex data processing, the primary goal of this thesis is to study the feasibility of deep generative models to capture the intricate relationship between ultrasound data and the physical properties of the medium that underlie them. In particular, I examine how diffusion models, known for their state-of-the-art capabilities in probabilistic modelling, can be used to retrieve high-resolution images of the brain from ultrasound data while overcoming the drawbacks of FWI. The methods proposed in this research hold the potential to drive impactful advancements in the diagnostic utility of ultrasound tomography, and greatly shift how we approach brain imaging in clinical settings.

1.3.1 Chapters Overview

This thesis is comprised of three original research chapters, each addressing a specific problem related to the main objective of this work and providing unique insights within the broader context of ultrasound brain imaging with diffusion models. Additionally, this thesis includes a comprehensive literature review of deep generative learning techniques and their associated formulations, as well as an overview of ultrasound tomography with full-waveform inversion, which serves as the conceptual benchmark for the outcomes of this work. This is presented in Chapter 2.

Chapter 3 introduces a novel approach to data comparison within the realm of convolutional analysis, specifically tailored to improve data fidelity in optimisation tasks. This chapter presents the Wiener Loss, a loss function rooted in Wiener filter theory that addresses the resolution limitations of traditional data comparison techniques like Mean Squared Error (MSE), and the instability and elevated costs of more sophisticated ones. By leveraging convolutional properties, the Wiener Loss preserves structural integrity and contextual correlations within the data, promoting full-spectrum recovery and mitigating the issues of high-frequency degradation seen in neural networks. The efficacy of the Wiener Loss is demonstrated in two key applications: data compression and medical imputation, where it notably improves fidelity in complex structures of both natural images and brain scans. The results show that Wiener Loss provides better data distribution capturing and improved perceptual quality, establishing it as a computationally efficient alternative to existing metrics. In addition to the Wiener loss, other Wiener-based are proposed, including a Wiener attention mechanism, a Wiener diffusion framework, and a discussion of how it can be reformulated to tasks that require translation invariability.

Chapter 4 explores the application of an unconditioned diffusion model for generating synthetic ultrasound neuroimaging samples, addressing the critical challenge of data scarcity in this field. Building upon prior work in medical imaging, this chapter adapts diffusion models to perform on ultrasound imaging, a domain traditionally constrained by low-resolution data due to limitations in traditional UT. By training an LDM on a small dataset (N=1,000) of acoustic brain images, this chapter demonstrates that a condition-free LDM can produce statistically accurate, structurally coherent and diverse synthetic images. This approach effectively augments the small dataset by generating on-demand samples at near real-time, while laying a robust foundation for the complex conditioning techniques explored in Chapter 5.

Chapter 5 proposes an extension of the diffusion model from Chapter 4 into a conditional framework that enables the direct reconstruction of high-resolution neuroacoustic images from ultrasound data, an ambitious approach aimed at overcoming the inherent limitations of traditional UT and FWI. By conditioning the diffusion model with embedded representations of the raw ultrasound data, the architecture

proposed in this chapter is designed to guide the generative diffusion process towards a tailored image synthesis in order to perform reliable end-to-end tomographic inversion at near-real-time speeds. In this chapter, the different strategies used to reduce the large dimensionality of the data through feature learning are discussed in depth. Although the proposed experiments could not be fully realised within the timeframe of this thesis, I outline potential avenues and adjustments that may enable its successful implementation in future work.

Finally, Chapter 6 summarises the main contributions of this thesis and discusses potential future research directions for ultrasound brain imaging with diffusion models. The chapter also reflects on the broader implications of this work for advancements in medical imaging, and the potential impact on clinical practice.

1.3.2 Publications

The first research chapter was initially presented as a poster titled *"Global Information Retrieval in Deep Learning using Wiener Filters"* at Imperial College London's *Towards the Future of AI - 2022* event, organised by the Imperial College AI Network, where it received the award for Best Poster. This work was subsequently uploaded as a full research paper to arXiv (https://arxiv.org/abs/2311.06558) and later accepted in the form of a poster and extended abstract at the Neural Information Processing Systems (NeurIPS) Medical Imaging Workshop (https://neurips.cc/virtual/2023/82490) under the title *"Convolve and Conquer: Data Comparison with Wiener Filters"*. I was invited to present this work in-person on behalf of the research team at NeurIPS. Both the poster and abstract are included at the end of this thesis document. Additionally, the implementation of the Wiener Loss has been made publicly available as an open-source library on GitHub (https://github.com/dpelacani/wienerloss).

Since completing the work presented in this thesis, I have been collaborating with colleagues on the development and testing of the Wiener Attention, a topic only briefly mentioned in Chapter 3. This work is progressing well, and we expect to release the first version on arXiv soon, with a planned submission to the International Conference on Machine Learning (ICML) 2025. We also intend to make the Wiener

Attention implementation available as an open-source library on GitHub.

The second and third research chapters have not yet been published. I aim to extend the implementation of Chapter 4 to enable 3D image synthesis, targeting a submission to the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI) 2025. Additionally, I plan to finalise the work in Chapter 5 and, if successful within the timeframe, include it in the same submission to MICCAI 2025.

1.4 Ethical Statement

As I present the work in this thesis, I wish to state clearly that it has been conducted with the sincere intention of improving human well-being and without any purposeful actions that could cause harm or negative impact. Aware that all research, particularly deep learning research, is subject to the biases inherent in data and the perspectives of the researchers, I openly invite any questions, criticisms, or concerns about my work. Should anyone identify issues or unintended consequences, I encourage them to reach out, as I am committed to maintaining transparency and addressing such matters responsibly.

On a broader level, as an AI researcher, I feel the dutiful need to appeal to those who shape the future of artificial intelligence — fellow researchers, government authorities, and corporate leaders — to carefully consider the ethical implications of their contributions in a world that is rapidly reshaping in the face of AI. The development and deployment of large-scale AI must prioritise the welfare of all living things, aiming to reduce, rather than exacerbate, and social divides. It is of profound importance to acknowledge and address the racial, ethnic, socioeconomic, and gender biases that inevitably permeate AI technologies, and to have trained experts rigorously examining and mitigating these biases. Open-source practices play a fundamental role here, enabling fair scrutiny and promoting resilience against misuse and bias reinforcement.

Governments and companies must also work together to address the large-scale social and economic changes that AI may bring, particularly in terms of job displace-

ment. Policies must be enacted to support retraining and relocation efforts, ensuring that workers' skills remain relevant, their sense of worth is preserved, and that their human rights and social conditions are not cast aside in favour of automation. Such measures are only just and humane.

Finally, I urge policymakers to place the welfare of people at the heart of an AI-driven society, that the democratisation of AI should be a shared objective, and that companies should consciously refrain from monopolistic control over widely applied AI solutions. Researchers, too, should be provided with the resources needed to pursue impactful work without the commercial pressures of capital gain as the driving factor. AI solutions should serve the common good, not become a tool for control or a mechanism for a new enslaved society. Now, with a calm sense of purpose, I humbly present this thesis.

Chapter 2

Background Theory

2.1 Generative Artificial Intelligence

Generative Artificial Intelligence (AI) is a specialised branch of AI focused on developing models that autonomously generate realistic data closely mimicking existing datasets. By employing techniques capable of producing diverse outputs such as text, images, and audio, generative AI preserves the essential characteristics of its training data while enabling applications that span creative content generation, data augmentation, and beyond. Its transformative potential is evident across industries, where it has become integral to modern technology. In entertainment, generative models drive advancements in the creation of art, music, video content, and interactive media, while in healthcare, they accelerate drug discovery and address data scarcity by augmenting datasets with synthetic alternatives.

The foundational concepts of generative AI are rooted in early theories of statistics and machine learning (ML). The primary focus was on modelling complex probability distributions of existing data for posterior sampling, particularly where traditional methods such as Maximum Likelihood Estimation (MLE), Bayesian models, and Gaussian Mixture Models (GMM) lacked the expressiveness or scalability required to represent intricate datasets. Generative modelling addresses this limitation by training a parameterised model $p_\phi(\mathbf{x})$, where ϕ represents the model parameters, to approximate the underlying data distribution such that $p_\phi(\mathbf{x}) \approx p_{\text{data}}(\mathbf{x})$.

The overarching goal is that, once trained, the model can generate samples

$\tilde{\mathbf{x}} \sim p_\phi(\mathbf{x})$ from the approximate distribution, such that they are representative of samples $\mathbf{x} \sim p_{\text{data}}(\mathbf{x})$ from the real data distribution. The optimisation process is framed as a probabilistic marginalisation of the data over a latent variable $\mathbf{z}$, sampled from a known prior distribution. The objective is then formulated as a divergence minimisation between conditional likelihood of the data given the latent variable and the marginal likelihood of the data:

$$\min_{\phi} \text{Div}\left(p_\phi(\mathbf{x} \mid \mathbf{z}) \parallel p_{\text{data}}(\mathbf{x})\right),$$

where Div is some function measuring the divergence between two distributions, and $p_\phi(\mathbf{x} \mid \mathbf{z})$ is the probabilistic transformation learned by the network.

In likelihood-based methods, the objective can also be reframed as a maximisation of the observed data under the model. This is formalised as an expectation over the data distribution:

$$\max_{\phi} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})}\left[\log p_\phi(\mathbf{x})\right],$$

where ϕ is iteratively adjusted to align $p_\phi(\mathbf{x})$ with $p_{\text{data}}(\mathbf{x})$. By optimising the log-likelihood, the model learns to synthesise new samples that preserve the structural and statistical characteristics of the training data. Overall, this mathematical reasoning underpins the diverse frameworks and training paradigms that define modern deep generative models.

In this section, I present a literature review structured to provide a comprehensive understanding of the foundations and advancements in generative AI. It begins with an exploration of the deep learning foundations, including an in-depth examination of earlier advancements in neural networks, followed by the various loss functions and optimisation techniques that enable efficient learning in high-dimensional spaces. Following this, the review dives into specific generative models, starting with Variational Autoencoders (VAEs), which employ probabilistic latent variable modelling to encode data distributions into a continuous latent space, enabling smooth interpolation and structured generation. This is followed by Generative Adversarial Networks (GANs), which leverage adversarial training between a generator and a discriminator

to synthesise highly realistic data distributions, addressing limitations in probabilistic frameworks. The discussion then turns to Diffusion Models, which iteratively refine data from random noise through a process modelled on stochastic differential equations, demonstrating exceptional performance in generating high-fidelity and diverse samples. The review further examines Transformers and Attention mechanisms, emphasising their transformative impact on generative tasks, particularly in text, image, and multimodal data generation. Additional generative frameworks, including energy-based models, autoregressive models, and flow-based models, are briefly discussed to provide a holistic overview of the field, although they are not directly related to this thesis. Finally, the review addresses common metrics and benchmarks used to evaluate the quality and performance of image generation tasks, which will serve as a foundation for the discussions in later sections.

While this review offers a comprehensive overview of foundational concepts and key milestones in generative AI, it is inherently not exhaustive. With increased accessibility to computational resources and a well-established culture of open-source contributions and publication repositories, the field of generative AI is rapidly evolving, with new research emerging at an unprecedented pace. Given the dynamic nature of this field, this literature review focuses on offering a structured understanding of the core principles and major developments within generative AI, while recognising the impossibility of covering every method of such a rapidly evolving domain.

2.1.1 Deep Learning Foundations

Deep learning is a subset of machine learning that utilises artificial neural networks to model and learn complex data representations. The foundational concepts of deep learning are rooted in the early theories of neural networks, which were inspired by the structure and function of the human brain's architecture (McCulloch and Pitts, 1943; Ivakhnenko, 1971; Rosenblatt, 1958). The development of deep learning models has been driven by advancements in computational power, the availability of large-scale datasets, and the development of efficient training algorithms (Rumelhart et al., 1986).

Neural Networks

Neural networks have been historically present in the study of artificial intelligence, originating from the 1940s when McCulloch and Pitts (1943) introduced a simplified computational model of a neuron. This model laid the foundation for the development of artificial neural networks, which gained popularity in the 1950s and 1960s with the arrival of the binary classifier model Perceptron by Rosenblatt (1958). However, the initial enthusiasm faded due to the limitations in computational power and the inability of early models to solve non-linear problems. The resurgence occurred in the 1980s with the development of multi-layer perceptrons and backpropagation algorithms capable of modelling non-linear problems, marking the beginning of modern neural network research (Rumelhart et al., 1986).

Early neural networks were designed to mimic the structure and function of the human brain, consisting of nodes, or neurons, arranged in layers, and interconnected through weighted connections (McCulloch and Pitts, 1943; Ivakhnenko, 1971; Rosenblatt, 1958). Feedforward neural networks (FFNs) are the simplest form of modern neural networks, consist of an input layer, one or more hidden layers, and an output layer. FFNs are formulated such that the connections between the nodes do not form a cycle, and the information flows in one direction, from input to output (Figure 2.1).

The connections between nodes are represented by weights, which are learned during the training process, and mimic the activation behaviour of brain synapses under different stimuli. The output of each layer is computed by applying the weighted sum of the inputs from the previous layer to a non-linear activation function, introducing the ability to model non-linear relationships in the data and capture complex patterns. For every i-th node of each j-th layer, the following holds:

$$h_i^{(j)} = \sigma\left(\sum_{k=1}^{n_{j-1}} w_{ik}^{(j)} h_k^{(j-1)} + b_i^{(j)}\right), \tag{2.1}$$

where σ represents the activation function, $w_{ik}^{(j)}$ is the weight connecting the k-th neuron in the $(j-1)$-th layer to the i-th neuron in the j-th layer, $h_k^{(j-1)}$ is the output of the k-th neuron in the $(j-1)$-th layer, and $b_i^{(j)}$ denotes the bias term for the i-th

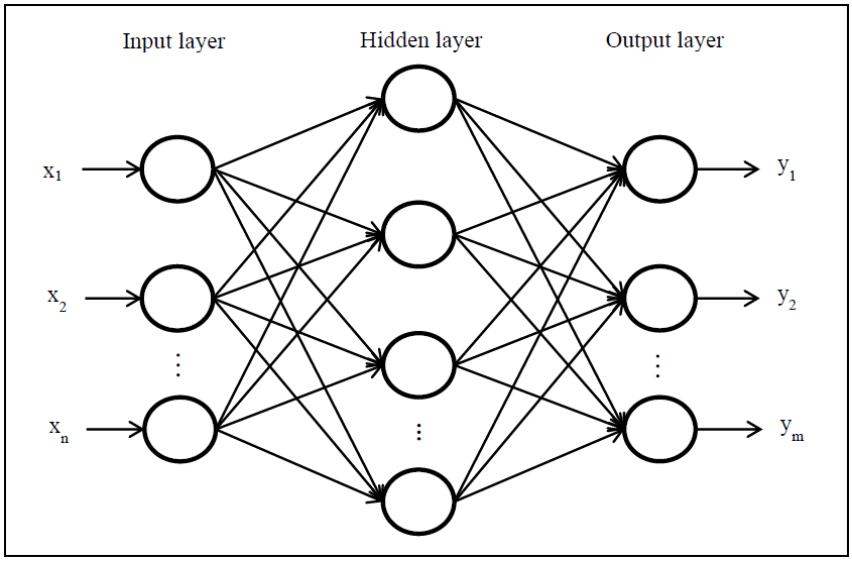


Figure 2.1: Illustration of a simple feedforward neural network architecture with one input, one hidden and one output layer. Each node (circles) in the previous layer is connected to every node in the subsequent layer through a weighted sum, where weights (lines) are learned during training. Image obtained from Ahmadian and Khanteymoori (2015)

neuron in the j-th layer. The term n_{j-1} represents the number of neurons present in the $(j-1)$-th layer.

In practice, Equation 2.1 is reformulated in matrix form to compute the output of the entire layer simultaneously, improving computational efficiency. The output of the j-th layer is given by:

$$\mathbf{h}^{(j)} = \sigma\left(\mathbf{W}^{(j)}\mathbf{h}^{(j-1)} + \mathbf{b}^{(j)}\right), \tag{2.2}$$

where $\mathbf{h}^{(j)}$ is the output vector of that represents the j-th hidden layer, $\mathbf{W}^{(j)} \in \mathbb{R}^{n_j \times n_{j-1}}$ is the weight matrix connecting the $(j-1)$-th layer to the j-th layer, where n_{j-1} is the number of neurons in the $(j-1)$-th layer and n_j is the number of neurons in the j-th layer, and $\mathbf{b}^{(j)}$ is the bias vector for the j-th layer. The activation function σ is applied element-wise to the weighted sum of the inputs. The collection of sequential operations defined by Equations 2.1 and 2.2 for all model layers is termed the *forward pass* of the network. FNNs are characterised by their simplicity and were among the first models capable of learning non-linear functions, providing crucial insights into the learning dynamics and optimisation techniques essential for

more complex architectures (Rumelhart et al., 1986).

Deep Neural Networks

The Universal Approximation Theorem proposes that a simple neural network with a single hidden layer containing a sufficient number of neurons can approximate any continuous function given sufficient data and appropriate activation functions (Cybenko, 1989; Hornik, 1991), although no current method has been proposed to obtain such optimal parameters. In compensation, the pursuit of deeper networks, with additional layers and neurons, has become a prevalent approach in the field (LeCun et al., 2015; Krizhevsky et al., 2012; Simonyan and Zisserman, 2014). These deeper networks, commonly referred to as deep learning models, are able to capture more intricate patterns and dependencies within data than their shallower counterparts, with the hope to approximate complex functions to a sufficient degree of accuracy.

Non-linearity

The manifold hypothesis posits that high-dimensional real-world data lies on lower-dimensional manifolds embedded within the higher-dimensional space (Goodfellow et al., 2016; Fefferman et al., 2016). This hypothesis implies that complex datasets can be effectively represented by these lower-dimensional structures. Each layer of the neural network performs a linear transformation that attempts to capture the underlying structure of the data, aligning the data more closely with these intrinsic manifolds (Olah, 2014). Linear transformation alone, however, are insufficient to model non-linear data structures and thus cannot fully reveal the representations of these manifolds. To address this, activation functions such as sigmoid, hyperbolic tangent, and rectified linear units (ReLU) (Nair and Hinton, 2010) are applied following these transformations to introduce non-linearity, allowing the network to capture intricate structures by transforming the data manifold into linearly separable features (Figure 2.2).

ReLU activation is a piecewise non-linear function that has been widely adopted in deep learning due to its simplicity, computational efficiency, and ability to promote sparsity by allowing unbounded positive outputs while zeroing out negative inputs

(Nair and Hinton, 2010; Glorot et al., 2011; Ramachandran et al., 2017). This sparsity reduces computational load, acts as a form of regularisation, and facilitates efficient gradient propagation through active neurons during training Glorot et al. (2011). However, the hard threshold at zero can lead to the "dying ReLU" problem, where too many neurons become inactive due to zero gradients for negative inputs, leading to vanishing gradient issues Maas et al. (2013); Bengio et al. (1994). Leaky ReLU, a piecewise linear variant, addresses this limitation by introducing a small, non-zero gradient for negative values, thereby improving gradient flow while also limiting neuron inactivity (Maas et al., 2013). Building on this, smoother activation functions such as the Sigmoid Linear Unit (SiLU), also known as Swish (Elfwing et al., 2018), along with Gaussian Error Linear Units (GeLU) (Hendrycks and Gimpel, 2016) and Mish (Misra, 2019), have been proposed. These functions provide smooth ReLU-like transformations that allow continuous gradient propagation while retaining the sparsity-promoting behaviour of ReLU, thereby supporting more stable optimisation and allowing the modelling of complex, non-linear data structures. Consequently, these smoother activations have become standard in many modern deep learning architectures (Rombach et al., 2022; Devlin et al., 2019b; Misra, 2019; Shazeer, 2020).

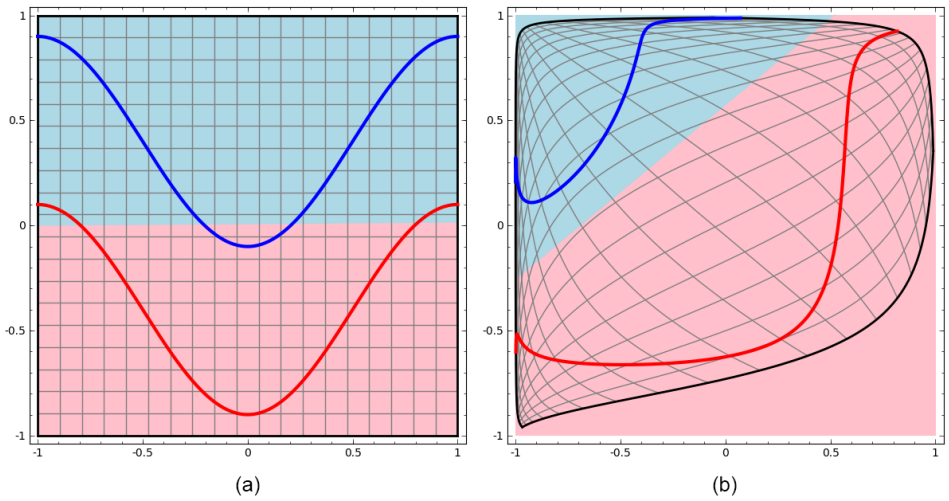


Figure 2.2: (a) Decision boundary of a simple neural network classifier with no hidden layers or activation functions. (b) Manifold representation of the data learned through hidden layers and activation functions in a neural network. The network learns to transform the data into a representation that is more linearly separable. Images obtained directly from Olah (2014)

Convolutional Neural Networks

Convolutional neural networks (CNNs), introduced in LeCun et al. (1989), rose as an evolution of FFNs, marking a great advancement in the field by introducing convolutional operations to neural networks. Convolution operations address the limitations of FFNs in processing grid-like data structures such as images (LeCun et al., 1998) by leveraging the spatial hierarchies within the data. The equivalent of hidden layers in CNNs are termed *feature maps*, or *activation maps*, which are obtained by convolving the output of the previous layer by a number of arbitrarily-sized filters to achieve the next forward state (Figure 2.3).

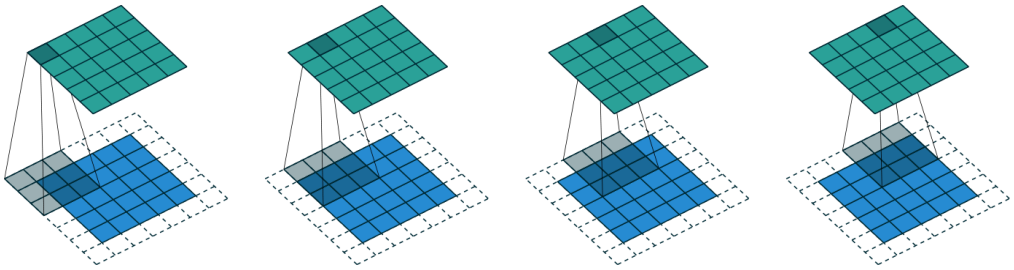


Figure 2.3: Illustration of the convolution process in a neural network, using a filter of size 3×3 on an input feature of size 5×5, demonstrating how feature maps are generated. The blue grid represents the feature map from the input layer, while the grey area shows the filter or kernel sliding across this input feature map, capturing local patterns through weighted sums at each position. As the filter moves, it generates activations that form a new feature map in the next layer, shown progressively in green at each step. This convolution extracts spatial features from the input, creating a localised feature representation in the subsequent layer. Extracted from Dumoulin and Visin (2016).

The feature maps of each layer are sequentially obtained from the previous layer in the following form:

$$\mathbf{H}_k^{(j)} = \sigma\left(\sum_{i=1}^{M} \mathbf{W}_{ik}^{(j)} * \mathbf{H}_i^{(j-1)} + \mathbf{b}_k^{(j)}\right), \tag{2.3}$$

where $\mathbf{H}_k^{(j)}$ is the k-th feature map of the j-th layer, $\mathbf{W}_{ik}^{(j)}$ is the filter with trainable weights connecting the i-th feature map of the $(j-1)$-th layer to the k-th feature map of the j-th layer, $\mathbf{H}_i^{(j-1)}$ is the i-th feature map of the $(j-1)$-th layer, and $\mathbf{b}_k^{(j)}$ is the bias term for the k-th feature map of the j-th layer. The term M represents the number of feature maps present in the $(j-1)$-th layer, and the activation function

σ is also applied element-wise to the weighted sum of the inputs.

The network is trained when the weights of the matrix that define the filter $\mathbf{W}_{ik}^{(j)}$ are optimised. Its dimensionality depends on the number of input channels, the number of output channels, and the size of the convolutional kernel, such that $\mathbf{W}^{(j)} \in \mathbb{R}^{N\times M\times K\times K}$, where N is the number of filters, M is the number of input channels, and K is the size of the (here 2D) convolutional kernel.

Due to their convolutional architecture, CNNs effectively capture spatial hierarchies in data, enabling the network to learn both local and global patterns more efficiently than FFNs. One notable advantage of CNNs is their ability to leverage convolutional and pooling operations to progressively reduce the spatial dimensions of input data. Additionally, CNNs employ a weight-sharing mechanism in convolutional layers, which drastically reduces the number of trainable parameters compared to FNNs for the same input size. This not only improves computational efficiency and memory usage but also enhances the model's ability to generalise by focusing on relevant spatial features, making CNNs particularly well-suited for tasks involving gridded data such as images.

Since their release, CNNs have demonstrated superior performance in image and video recognition tasks, exemplified by their success in computer vision benchmarks (Krizhevsky et al., 2012; Szegedy et al., 2016; He et al., 2016), and have been widely adopted in the field. LeNet-5 (LeCun et al., 1998) was one of the first successful applications of CNNs in handwritten digit recognition. AlexNet (Krizhevsky et al., 2012) advanced the field by using deeper architectures and GPU acceleration to win the ImageNet (Deng et al., 2009) challenge of natural images classification. Inception networks (Szegedy et al., 2016) introduced novel concepts like inception modules to handle varying filter sizes. ResNet (He et al., 2016), introduced residual connections which allows inputs to bypass one or more layers, mitigating the vanishing gradient problem (Hochreiter et al., 1998), and VGG networks (Simonyan and Zisserman, 2014) emphasised depth of convolutional layers to reach competitive performance.

Model Training

Training of a neural network is fundamentally an optimisation problem aimed at approximating complex non-linear functions. Training of neural networks is typically achieved through the backpropagation algorithm (Rumelhart et al., 1986), where a measure of error, termed the *loss function*, is minimised for each input-target pair in a gradient-based optimisation process (LeCun et al., 1998). Backpropagation facilitates the efficient computation of the gradient of the loss function with respect to the model parameters (weights and biases) by employing a series of chain rules from calculus. These gradients are then used to update the parameters in the direction that reduces the loss function value, iteratively improving the model's performance (see Algorithm 1). Once the model has been trained, it can then be used to make inferences on new, unseen data.

In practice, backpropagation is facilitated by computational graphs, which provide a structured representation of the sequence of operations and dependencies within a neural network. These graphs break down complex computations into simpler, interconnected nodes, enabling the efficient application of the chain rule during the backward pass. By mapping out dependencies, computational graphs allow for accurate gradient computation with respect to each parameter, tracing the path from the output back to the input. Automatic differentiation tools, available in modern deep learning frameworks, make use of computational graphs to enable rapid experimentation and development of deep-learning models. These frameworks leverage GPU acceleration and parallelisation to perform large-scale computations efficiently, speeding up the training process (Abadi et al., 2015; Paszke et al., 2019; Baydin et al., 2017).

Generalisation

Generalisation in neural networks refers to the ability of a model to perform well across a variety of new, unseen datasets, rather than just on the data it was trained on (Goodfellow et al., 2016). This characteristic is critical in real-world applications, where the ability to predict or classify accurately outside of controlled conditions can

Algorithm 1 Training a Simple CNN Model

Require: Training data (X, y), Number of epochs N, Batch size B, Learning rate α

Initialise the CNN model parameters (weights and biases)
for epoch $= 1$ to N **do**
  for each mini-batch (X_b, y_b) in the training data **do**
    Forward pass: Compute the output predictions $\hat{y}_b$ from X_b using the CNN
    Compute the loss $\mathcal{L}(\hat{y}_b, y_b)$ (e.g., cross-entropy loss)
    Backward pass: Compute the gradients of the loss $\nabla_\theta \mathcal{L}$ with respect to model parameters
    Update model parameters using some form of gradient descent, e.g.:

$$\theta \leftarrow \theta - \alpha \cdot \nabla_\theta \mathcal{L}(\theta)$$

  end for
  Optionally, evaluate the model on the validation set for hyperparameter tuning
end for
return Trained CNN model

mark the difference between a useful model and a flawed one. Inadequate generalisation can significantly impact such models; for instance, a neural network trained to detect abnormalities in medical scans might fail to adequately perform when exposed to scans from different demographics or different imaging equipment than those it was trained on, leading to potential misdiagnoses (Suzuki, 2017; Nichyporuk et al., 2022; Vakanski and Xian, 2021; Samala et al., 2020).

Bias and variance are fundamental concepts that explain the different types of error a model might exhibit. Bias is the expected difference between the estimator's predictions and the true value, while variance measures the variability of the estimator's predictions across different data samples (Goodfellow et al., 2016). In machine learning applications, bias results from a model's assumptions being too simplistic: high bias models are unable to capture the complex patterns in the dataset, resulting in underfitting. Variance, conversely, occurs when a model is excessively sensitive to small fluctuations in the training set, capturing noise as if it were signal, which leads to overfitting (James et al., 2013; Goodfellow et al., 2016). The capacity of the model, which refers to its ability to fit a wide variety of functions, influences both bias and variance. Models with high capacity, if not properly regulated, can easily overfit by memorising features specific to the training data (James et al., 2013),

thus failing to generalise to new data. Conversely, models with insufficient capacity might not capture sufficient complexity, leading to high bias. The goal is to find the optimal model capacity that minimises both bias and variance, achieving the best generalisation performance (Domingos, 2000; Goodfellow et al., 2016).

The generalisation error, which measures the discrepancy between the performance of the model on new data compared to the training data (Goodfellow et al., 2016; James et al., 2013), is a crucial metric for assessing a model's effectiveness in practical applications (James et al., 2013). Strategies to minimise this error involve regularisation techniques of various natures, such as proper segmentation of data into training, validation, and testing sets, cross-validation, hyperparameter tuning, and data augmentation. Discussions over generalisation errors tend to focus on the overfitting problem, since underfitting can be simply addressed by increasing the model's complexity.

The division of data into training, validation, and testing sets helps to mitigate overfitting and underfitting by allowing for the independent evaluation of model performance. The training set is used to fit the model, the validation set to tune the hyperparameters, and the testing set to provide an unbiased evaluation of a final model fit to the training dataset (James et al., 2013; Bergstra and Bengio, 2012). Hyperparameters, which are all parameters that define the model's architecture and training process except the trainable weights of the model, influence the behaviour of the training algorithm and include choices such as the learning rate, the complexity of the model, or the penalty terms used in regularisation. Cross-validation involves performing different trainings for different subsets of training and validation splits, and combines the outcomes by average, allowing for a more robust generalisation (Arlot and Celisse, 2010). This technique helps ensure that the model is not tuned to only a specific part of the training data.

Regularisation techniques are employed to the model weights in order to improve a model's generalisation by discouraging complex models in various ways. L1 regularisation (or Lasso) introduces a penalty equivalent to the absolute value of the magnitude of the coefficients, leveraging its sparse-promoting nature that forces the model to learn only the most important features from the data (Shen et al., 2022;

Goodfellow et al., 2016). This feature selection approach essentially reduces the complexity of the model by reducing coefficients of less important features to zero (Goodfellow et al., 2016). L2 regularisation (or Ridge) introduces a penalty equivalent to the square of the magnitude of the coefficients, and like L1 it helps control overfitting by penalising large coefficients. However, unlike L1 regularisation, which can completely eliminate the influence of some neurons, L2 regularisation merely reduces their contribution. This approach means that L2 regularisation does not carry out feature selection but ensures that every feature has some degree of influence on the model, which is beneficial when all features are considered important (Goodfellow et al., 2016).

Dropout, another regularisation strategy, involves randomly deactivating a subset of neurons during each training step (Srivastava et al., 2014). This effectively means that the network trains on a different subset of neurons at each step, which prevents co-adaptation of features and forces the network to learn more general information that is useful across different subsets. This method of feature selection is conceptually similar to the sparsity-promoting approach of L1 regularisation, but it operates through a stochastic process that randomly deactivates neurons during training, rather than explicitly minimising the magnitude of coefficients.

Another important regularisation strategy is batch normalisation. During training, batch normalisation calculates the mean and variance of the activations for each mini-batch. It then standardises the activations by subtracting the mean and dividing by the standard deviation, reducing internal covariate shift. After standardisation, it applies a scaling factor and a shift parameter, which are learned during training, thus ensuring that the activations can still represent a wide range of values (Ioffe and Szegedy, 2015). Batch normalisation has been shown to accelerate training of deep neural networks by smoothing the optimisation landscape, resulting in more stable gradients, and allowing for higher learning rates to achieve convergence (Luo et al., 2018; Santurkar et al., 2018). In addition to stabilising the training process, batch normalisation also acts as a regularisation technique by introducing noise to the training process due to the variability in the batch statistics (mean and variance), which are computed from mini-batches of data (Luo et al., 2018; Santurkar

et al., 2018; Ioffe and Szegedy, 2015; Goodfellow et al., 2016). This makes the model less dependent on specific feature representations and prevents it from overfitting. (Santurkar et al., 2018).

One drawback of batch normalisation is that its efficacy is reduced when the batch size is small, as the statistics calculated for each mini-batch are less representative of the overall dataset. This can lead to adverse training convergences and poor generalisation. To address this issue, two derivative techniques have been proposed. The first is group normalisation, which divides the channels of the input into groups and standardises them separately. This allows for more stable statistics to be calculated, even with small batch sizes (Wu and He, 2018). The second technique is layer normalisation, which calculates the mean and variance of the activations across all neurons in a layer for each individual input sample, rather than across a mini-batch, making it particularly effective for stabilising training in scenarios where small batch sizes or single-sample training are required (Ba et al., 2016).

Early stopping is another regularisation technique that helps prevent overfitting by stopping the training process when the model's performance on a validation set begins to degrade (Prechelt, 1998). At that point, it is understood that the model is merely memorising the training data and is no longer generalising well to unseen data.

Data augmentation is another widely employed regularisation technique in deep learning serving to artificially increases the size and diversity of the training dataset by applying random but realistic transformations to the training images (e.g. rotations, shifts, elastic transformations). This process helps the model generalise better by exposing it to a more representative variability seen in the real-world without requiring further data acquisition (Krizhevsky et al., 2012; Shorten and Khoshgoftaar, 2019; Wei and Zou, 2019; Shorten and Khoshgoftaar, 2019; Frid-Adar et al., 2018; Pinaya et al., 2023). A popular approach for data augmentation is training generative models that are capable of modelling the given data distribution and generating new samples that fall in close proximity to it. This approach has been shown to be effective in improving the generalisation of deep learning models, particularly in scenarios where the original training dataset is limited (Shorten and Khoshgoftaar,

2019; Wei and Zou, 2019; Frid-Adar et al., 2018; Pinaya et al., 2023; Shin et al., 2018).

Another regularisation technique that has been shown to improve generalisation is noise robustness. Noise robustness involves introducing variability to the training process by deliberately adding noise to inputs or outputs, which encourages the model to learn to disregard minor variations or perturbations in the data (Bishop, 1995). This has been shown to be equivalent to training the model with L2 regularisation (Bishop, 1995).

Bagging ensemble methods, such as Random Forests and Gradient Boosting Machines, improve generalisation by training multiple instances of a model on different subsets of the training data and combining their weights to make a final, combined model prediction. In fact, dropout regularisation can be seen as a form of ensemble method, where the model is trained on different subsets of neurons at each iteration. This ensemble approach has been shown to increase the robustness and generalisability of models (Breiman, 1996; Friedman, 2001).

Supervised and Unsupervised Learning

Supervised and unsupervised learning represent two fundamental paradigms in machine learning, differing primarily in the nature of the labelled data available during the training process. The distinction between these methods is particularly significant when analysed from a statistical perspective, as the statistical assumptions and goals diverge between the two tasks (Goodfellow et al., 2016).

In supervised learning, the training dataset consists of input-output pairs, where the inputs, $\mathbf{x}$, are observed along with corresponding outputs, or labels, $\mathbf{y}$. The primary objective is to learn a function $f : \mathbf{x} \rightarrow \mathbf{y}$ that maps inputs to outputs based on the labelled data. Statistically, supervised learning can be interpreted as an exercise in function approximation (regression) when $\mathbf{y}$ is continuous, and classification when $\mathbf{y}$ is categorical (Goodfellow et al., 2016; Murphy, 2012). The underlying assumption is that the training data is representative of the distribution from which future test data will be drawn.

From a statistical perspective, supervised learning involves estimating the con-

ditional probability distribution $p(\mathbf{y} \mid \mathbf{x})$. This estimation process typically relies on maximising the likelihood of the observed data under the model, which may be achieved through various optimisation techniques, such as gradient descent. The choice of model (e.g., linear regression, decision trees, neural networks) affects the complexity and flexibility of the function f, and hence, the ability of the model to generalise to unseen data (Bishop and Nasrabadi, 2006).

In machine learning, the most fundamental supervised learning tasks are classification and regression. Consider a linear regression model where the objective is to predict a continuous outcome $\mathbf{y}$ based on a set of predictors $\mathbf{x}$. The model assumes a linear relationship of the form:

$$\mathbf{y} = \mathbf{x}^\top \beta + \epsilon, \tag{2.4}$$

where β is the vector of coefficients to be estimated, and ϵ represents the error term, assumed to follow a zero-centered normal distribution. The parameters β are estimated by minimising the sum of squared errors, a method known as ordinary least squares, which implicitly maximises the likelihood under the assumption of normally distributed errors (Hastie et al., 2009). In linear regression, this maximisation is simple enough to be performed analytically, but more complex models require iterative optimisation algorithms to meaningful solutions.

In classification tasks, particularly binary classification, logistic regression is a widely-used method. Here, the output y is categorically binary, and the model estimates the probability that $y = 1$ given $\mathbf{x}$ as:

$$p(y = 1 \mid \mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{x}^\top \beta)}, \tag{2.5}$$

where, the coefficients β are estimated by maximising the likelihood of the observed labels y given the predictors $\mathbf{x}$, which often requires iterative optimisation algorithms due to the non-linearity of the logistic function (Goodfellow et al., 2016). The model is then used to predict the probability of $y = 1$ for new observations, and a threshold is applied to convert these probabilities into binary predictions (e.g. $p > 0.5$ predicts class 1 but otherwise 0). Other supervised learning tasks include object detection,

image segmentation, and natural language processing, among others (Redmon et al., 2016; Lin et al., 2017a; Long et al., 2015; Ronneberger et al., 2015; Vaswani et al., 2017; Devlin et al., 2019a), where the conceptual paradigm of predicting a label based on a predictor input data remains the same.

In contrast, unsupervised learning involves learning patterns from data that does not contain annotated labels. The goal is to infer the underlying structure or distribution in the data, which is often achieved by modelling the marginal distribution $p(\mathbf{x})$ rather than the conditional distribution $p(\mathbf{y} \mid \mathbf{x})$. This is particularly useful for tasks such as clustering, density estimation, dimensionality reduction, and data synthesis, where the absence of labelled data prevents the direct application of supervised methods (Bishop and Nasrabadi, 2006). From a statistical perspective, unsupervised learning often relies on models that make assumptions about the underlying data distribution, such as Gaussian mixture models in clustering, or methods that attempt to capture the variance in the data, such as Principal Component Analysis (PCA) (Jolliffe, 2002).

Clustering is a popular example of unsupervised learning aims to partition the data into groups (clusters) such that data points within a cluster are more similar to each other than to those in other clusters. A common approach is the K-means algorithm, which assumes that each cluster can be represented by its centroid mean, and the objective is to minimise the within-cluster variance (MacQueen, 1967):

$$\min_{\{\mu_k\}} \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in C_k} \|\mathbf{x}_i - \mu_k\|^2, \tag{2.6}$$

where μ_k is the mean of cluster C_k computed for k numbers of clusters. This optimisation is typically solved iteratively, by alternating between assigning data points to the nearest cluster centroid and updating the cluster means (Murphy, 2012).

Dimensionality reduction is another common unsupervised learning task that aims to reduce the number of features in the data while preserving as much information as possible. PCA is a popular statistical method used for dimensionality reduction. It identifies the directions (principal components) that maximise the variance in the data. Mathematically, PCA involves finding the eigenvectors and eigenval-

ues of the covariance matrix of $\mathbf{x}$. The eigenvectors correspond to the directions of maximum variance and the eigenvalues indicate the magnitude of variance along these directions. The data can then be projected onto the N-dimensional subspace defined by the eigenvectors associated with the highest N eigenvalues, reducing the dimensionality while preserving as much data variance as can be contained in these N dimensions (Jolliffe, 2002).

Generative models are another important class of unsupervised learning models that aim to model the underlying distribution of the data. These models can be used to generate new samples that are similar to the training data, or to perform tasks such as anomaly detection or data denoising. Popular generative models include Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), and Denoising Diffusion Probabilistic Models (DDPM) which have been used in a wide range of applications, including image generation, unpaired domain transfer, super-resolution, and data augmentation (Isola et al., 2017; Pinaya et al., 2022; Goodfellow et al., 2014, 2016; Saharia et al., 2022; Dalca et al., 2019). These are explored in detail later in this chapter.

Other types of learning include semi-supervised learning, where the model is trained on a combination of labelled and unlabelled data, and reinforcement learning, where the model learns to make decisions by interacting with an environment and receiving rewards or penalties based on its actions. These models have been used in a wide range of applications, including robotics, game playing, and natural language processing (Sutton and Barto, 2018; Sutton et al., 2000; Mnih et al., 2013).

2.1.2 Loss Functions and Optimisation Algorithms

Deep learning is inherently an optimisation problem that seeks to approximate highly complex, non-linear functions through the iterative refinement of model parameters. These models, structured as neural networks, aim to map inputs to desired outputs with minimal error. This minimisation process is governed by a loss function, which provides a scalar value that quantifies the misfit of what the the model is trying to learn. Backpropagation calculates the gradient of the loss function with respect to each trainable parameter in the network, using the chain rule of calculus to propagate

the error from the output layer back through the hidden layers, to the input layer. These gradients are then employed, predominantly, in gradient-based optimisation algorithms to update the network's parameters in a direction that minimises the loss value, thereby improving the model's predictive performance over successive iterations (Goodfellow et al., 2016)

The critical role of loss functions in deep learning extends beyond error calculation; they shape the entire optimisation landscape, influencing how a model learns and generalises (Li et al., 2018). The loss function directs the parameter updates in a manner that should, ideally, lead to convergence to the global or a sufficiently optimal local minimum of the loss landscape (Li et al., 2018). From an optimisation perspective, loss functions that are smooth, differentiable, and convex are desirable, as they stabilise and facilitate iterative optimisation in the landscape, and lead to better generalisation outcomes (Boyd and Vandenberghe, 2004; Li et al., 2018; Nocedal and Wright, 2006).

Deep learning tasks are diverse and encompass a broad range of applications, each with distinct requirements for loss functions. In regression tasks, where the goal is to predict continuous outputs, loss functions like Mean Squared Error (MSE) and Mean Absolute Error (MAE) are commonly employed. These functions quantify the difference between predicted and target continuous values, with MSE being particularly sensitive to outliers due to its quadratic nature. In classification tasks, where the objective is to predict discrete class labels, the *Cross-Entropy Loss* is typically used. This loss function is grounded in probabilistic principles, penalising incorrect classifications based on the predicted probability distribution. For sequence prediction tasks, often seen in natural language processing and time-series forecasting, the *Negative Log-Likelihood* (NLL) loss is frequently utilised (Bishop and Nasrabadi, 2006; Goodfellow et al., 2016). Object detection, a task that requires both classification and localisation, employs compound loss functions that combine aspects of cross-entropy and regression losses (Ren et al., 2015). Reinforcement learning, another distinct area, typically employs reward-based losses, where the objective is to maximise cumulative rewards of an agent in an environment over time (Sutton and Barto, 2018; Sutton et al., 2000).

In image generation, the focus of this thesis, the choice of loss function directly impacts the quality, realism, and fidelity of the generated images (Rombach et al., 2022; Jiang et al., 2021; Zhang et al., 2018b). Image comparison tasks require loss functions that can quantify the difference between two images, treated as continuous data structures, assessing their similarity or dissimilarity. One of the simplest yet effective loss functions in image generation tasks is the MAE, also known as the L1 norm. It is mathematically defined as:

$$\mathcal{L}_{MAE}(\mathbf{x}, \mathbf{y}) = \frac{1}{n} \sum_{i=1}^{N} |x_i - y_i|, \tag{2.7}$$

where x_i and y_i are the i-th index of the true and predicted values, respectively, and n is the number of features in the data. The MAE computes the average of the absolute differences between the predicted and true values, providing a measure of the average intensity difference between two images. When MAE is used as a loss function, it implicitly assumes that the residuals $\epsilon_i = y_i - \hat{y}_i$ are Laplace-distributed, which makes the model more robust to outliers. In other words, large deviations do not disproportionately influence the overall loss, making MAE particularly useful in situations where data contains significant noise or outliers.

MAE's primary advantage in optimisation is its ability to produce sparse solutions, which is desirable in high-dimensional spaces where interpretability and feature selection are important (Tibshirani, 1996). Sparse solutions reduce the risk of overfitting by focusing on the most relevant features and ignoring irrelevant ones, although overly-sparse models may end up suffering from underfitting (Shen et al., 2022; Yan et al., 2010).

In contrast, the MSE loss, also known as the L2 norm, is defined as:

$$\mathcal{L}_{MSE}(\mathbf{x}, \mathbf{y}) = \frac{1}{n} \sum_{i=1}^{N} (x_i - y_i)^2, \tag{2.8}$$

where x_i and y_i are the true and predicted values, respectively, and n is the number of features in the data. MSE computes the average of the squared differences between the predicted and true values, penalising larger errors more severely than MAE. Statistically, MSE is related to the Gaussian (or normal) distribution by assuming

that the residuals are normally distributed with zero mean. When the underlying noise in the data is Gaussian, MSE provides the maximum likelihood estimate (MLE) of the parameters. MSE is also smooth and differentiable everywhere (Nocedal and Wright, 2006), which makes it particularly amenable to gradient-based optimisation methods. The smoothness ensures that the gradient exists at all points, leading to more straightforward convergence properties.

However, MSE is more sensitive to outliers than the MAE, as the squared term amplifies the impact of large errors on the overall loss value. This can lead to suboptimal performance when the dataset contains noisy or extreme values, as the model may focus too heavily on minimising the impact of outliers at the expense of generalisation.

Both MAE and MSE are element-wise loss functions that compare the intensity values of corresponding elements in two signals. These loss functions are computationally efficient and effective for quantifying overall differences between images. However, they often produce blurry results when applied to image reconstruction tasks because they treat each element independently, ignoring spatial correlations and context. This lack of spatial awareness causes high-frequency details, such as edges and fine textures, to be averaged out, resulting in smooth but less sharp images (Wang and Bovik, 2009). In the case of MSE, the blurriness is further exacerbated by its quadratic formulation, which disproportionately penalises large deviations. Since these large deviations are often associated with prominent low-frequency components, such as smooth transitions and broad features, the model tends to overfit to minimise such errors, neglecting high-frequency details crucial for perceptual quality. This prioritisation of low-frequency resolutions is demonstrated in Figure 2.4, where a principal component analysis (PCA) of the MSE between MNIST (LeCun et al., 1998) and gaussian noise shows that lower-frequencies are represented by higher variances, and in turn, higher priority in the error propagation. As a result, models optimised under MSE tend to generate outputs that capture low-frequency information accurately but fail to maintain sharp, high-frequency features, leading to a loss of detail and the characteristic blurriness in the generated images (Wang and Bovik, 2009).

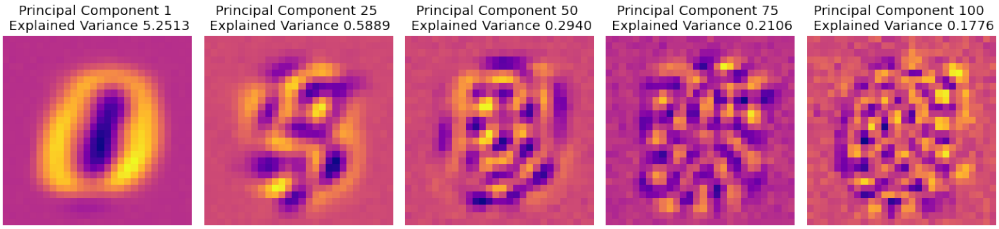


Figure 2.4: PCA analysis of the MSE between MNIST and Gaussian noise. PCA decomposes the error into its principal components, where each component is an eigenvector of the covariance matrix of the error. The variances are the eigenvalues of the covariance matrix, indicating the importance of each principal component in the error. The plot shows the variances of the error for different principal components, highlighting the prioritisation of low-frequency resolutions in MSE. Variance increases to the right.

This limitation has motivated the development of perceptual loss functions, which aim to capture high-level features and spatial correlations in images to generate more visually appealing and realistic outputs (Pappas et al., 2000; Johnson et al., 2016; Zhang et al., 2018a). One such perceptual metric is the Structural Similarity Index (SSIM) — and its extension, Multi-Scale Structural Similarity Index Measure (MS-SSIM) — which quantifies the perceptual similarity between two images by considering changes in structural information such as luminance, contrast, and general structure (Wangz et al., 2004). The SSIM is mathematically defined as:

$$\text{SSIM}(\mathbf{x}, \mathbf{y}) = \frac{(2\mu_{\mathbf{x}}\mu_y + C_1)(2\sigma_{\mathbf{x}y} + C_2)}{(\mu_{\mathbf{x}}^2 + \mu_y^2 + C_1)(\sigma_{\mathbf{x}}^2 + \sigma_y^2 + C_2)}$$

where $\mu_{\mathbf{x}}$ and μ_y are the mean intensities, $\sigma_{\mathbf{x}}$ and σ_y are the variances, and $\sigma_{\mathbf{x}y}$ is the covariance of images $\mathbf{x}$ and y, respectively. Constants C_1 and C_2 are used to stabilise the division.

As a loss function, SSIM is particularly advantageous over MAE and MSE because it measures the perceptual similarity between images with statistical descriptors rather than element-wise differences. This makes SSIM more aligned with human visual perception, effectively capturing spatial dependencies and high-level textures that element-wise norms fail to address, resulting in visually superior image reconstructions (Wang et al., 2004; Zhao et al., 2016). MS-SSIM is an extension of

SSIM that evaluates the structural similarity at multiple scales or resolutions. This approach allows MS-SSIM to capture more detailed information about the image structure at different levels, making it more robust and reliable for assessing image quality.

While both losses offer significant advantages in preserving perceptual quality by focusing on structural similarities, it also presents several limitations that impact its effectiveness as a loss function in deep image generation tasks. Both SSIM and MS-SSIM are highly sensitive to changes in luminance and contrast, which can lead to undervalue of images that differ slighly in those parameters despite having great perceptual compatibility. Specifically, they assume that structural similarity is maintained if local mean and variance are preserved, which is not always true in practical scenarios (Nilsson and Akenine-Möller, 2020). In addition, their reliance on local statistics, such as means and variances, makes it vulnerable to image noise and artefacts, and can lead to inaccurate quality assessments. This reliance also makes SSIM and MS-SSIM less robust in scenarios where images have local variations or local distortions, or where there is little variation in pixel intensities (low variance) (Nilsson and Akenine-Möller, 2020; Pambrun and Noumeir, 2015). This can result in poor convergence and inferior model performance, particularly in tasks involving fine-grained image details or complex textures (Nilsson and Akenine-Möller, 2020).

The Focal Frequency Loss (FFL) (Jiang et al., 2021) is an advanced loss function designed to address the low-frequency bias inherent in MSE, particularly within deep image generation tasks. Unlike conventional element-wise losses that operate in the spatial domain, FFL operates in the frequency domain, targeting discrepancies between generated and target images across various frequency bands. Formally, it is defined as:

$$\mathcal{L}_{\text{FFL}}(\mathbf{x}, \mathbf{y}) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \mathbf{w}(u, v) |F_{\mathbf{x}}(u, v) - F_{\mathbf{y}}(u, v)|^2, \tag{2.9}$$

where M and N denote the dimensions of the frequency domain representation. The FFL is computed by performing the Hadamard product (element-wise multiplication) between the spectrum weight matrix $\mathbf{w}(u, v)$ and the squared differences between

the Fourier coefficients of the true and generated images, $\mathbf{x}$ and $\mathbf{y}$, respectively. The weight matrix is formulated to emphasize discrepancies in the frequency domain by focusing on frequencies where the predicted and target images differ the most (Jiang et al., 2021), and is defined as:

$$\mathbf{w}(u, v) = |F_{\mathbf{x}}(u, v) - F_{\mathbf{y}}(u, v)|^{\alpha}, \tag{2.10}$$

where $F_{\mathbf{x}}(u, v)$ and $F_{\mathbf{y}}(u, v)$ are the Fourier transform coefficients of the reference (target) image and the generated (predicted) image, respectively, at frequency coordinates (u, v). The parameter α serves as a scaling factor for flexibility, with $\alpha = 1$ typically used in experiments. The weight matrix $\mathbf{w}(u, v)$ is further normalised to the range $[0, 1]$, where the value 1 corresponds to the frequency with the greatest loss, and lower values correspond to less significant frequencies.

The resulting loss function focuses on those frequencies that contribute most to the perceptual differences between the images, thereby guiding the model to correct these specific discrepancies and ultimately improving the visual quality of the generated images (Jiang et al., 2021). However, while FFL is adept at preserving high-frequency information, it may not fully capture other perceptual qualities, requiring integration with other reconstructive loss functions to comprehensively address the multifaceted requirements of deep image generation tasks (Jiang et al., 2021).

Deep perceptual losses, on the other hand, have gained prominence for their ability to align generated images more closely with human visual perception (Zhang et al., 2018a; Dosovitskiy and Brox, 2016; Pappas et al., 2000; Johnson et al., 2016). They operate on higher-level features extracted from pre-trained convolutional neural networks, leveraging the network's ability to capture complex patterns and textures that are more aligned with human perception (Dosovitskiy and Brox, 2016).

Mathematically, if $\phi_l(\mathbf{x})$ and $\phi_l(\mathbf{y})$ represent the feature maps at layer l of a trained network for the generated image $\mathbf{x}$ and the target image $\mathbf{y}$, respectively, the perceptual loss can be defined as:

$$\mathcal{L}_{\text{perceptual}}(\mathbf{x}, \mathbf{y}) = \sum_{l} \|\phi_l(\mathbf{x}) - \phi_l(\mathbf{y})\|_2^2 \tag{2.11}$$

This loss function measures the Euclidean distance between the feature representations of the generated and target images across several layers l of the network (Johnson et al., 2016). By focusing on these higher-level abstractions rather than element-by-element differences, perceptual losses help to preserve the global structure and finer details of the image.

One of the key advantages of perceptual losses is their ability to maintain the semantic content and stylistic features of images. For example, in style transfer applications, perceptual losses can be used to ensure that the generated image retains the content structure of the original image while adopting the style of a reference image (Johnson et al., 2016). This is achieved by comparing the content and style features separately, often using different layers of the CNN. The lower layers, which capture more detailed information, are typically used for content preservation, while higher layers, which capture more abstract features, are used for style transfer (Johnson et al., 2016).

A particularly popular perceptual loss is the Learned Perceptual Image Patch Similarity (LPIPS) metric (Zhang et al., 2018b). LPIPS extends the concept of perceptual loss by employing deep features from multiple layers of a pre-trained network and weighting these features based on their contribution to perceptual similarity. The LPIPS metric is defined as:

$$\text{LPIPS}(\mathbf{x}, \mathbf{y}) = \sum_{l} \frac{1}{H_l W_l} \sum_{h=1}^{H_l} \sum_{w=1}^{W_l} \| w_l \odot (\phi_l(\mathbf{x})_{hw} - \phi_l(\mathbf{y})_{hw}) \|_2^2, \tag{2.12}$$

where $\phi_l(\mathbf{x})$ and $\phi_l(\mathbf{y})$ represent the feature maps of images x and y extracted from the l-th layer of a pre-trained network, with H_l and W_l denoting the height and width of the feature map at layer l. The learned weights w_l, applied to each feature channel, adjust the importance of different feature activations to better align with human perceptual judgments. The squared Euclidean distance, $\| \cdot \|_2^2$, is used to measure the difference between corresponding feature activations, while the element-wise multiplication, $\odot$, allows for weighting these differences by w_l (Zhang et al., 2018b). This approach allows LPIPS to more accurately reflect human judgment of

image quality in general.

However, similar to other perceptual losses, LPIPS requires significant computational resources due to the need for forward passes through deep networks. Additionally, the reliance on pre-trained weights introduces a dependency on the specific dataset and task, which can limit generalisability across different applications. Perceptual losses also depend heavily on the choice of pre-trained networks and the specific layers used for feature extraction. Different networks or layers can yield different perceptual qualities, which are directly related to the dataset they were trained on and the hyperparameters that defined the pre-training. If the pre-trained kernels do not extract useful features for the new task, then the perceptual loss may not be effective. Finally, similar to the focal frequency loss, perceptual losses do not optimally perform alone and require integration with other loss functions to prevent high-frequency artifacts and hallucinations (Zhang et al., 2018b).

Adversarial losses are another class of loss functions that leverage the capabilities of neural networks to generate high-quality data by introducing a model that discriminate between real and generated data. They form the foundational concept of Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), and will be discussed in detail in Section 2.1.4

The Kullback-Leibler (KL) divergence is another widely used measure in deep generative learning, particularly in the framework of variational inference. Conceptually, KL divergence quantifies how one probability distribution diverges from another, providing a means to compare the statistical properties of two distributions. For a discrete case, the KL divergence of a distribution $q(x)$ from $p(x)$ is defined as:

$$\mathrm{KL}(q(x) \parallel p(x)) = \sum_{x \in X} q(x) \log \left(\frac{q(x)}{p(x)} \right). \tag{2.13}$$

The KL divergence is a strictly-positive value that quantifies the expected log difference between probabilities assigned by $q(x)$ and $p(x)$, with a lower value indicating greater similarity between the distributions. It is important to note that the KL divergence is asymmetric, such that $\mathrm{KL}(q \parallel p) \neq \mathrm{KL}(p \parallel q)$. This asymmetry places greater emphasis on regions where $q(x)$ has higher density for Equation 2.13,

making it particularly sensitive to inaccuracies in these areas. The KL divergence plays a crucial role in the context of variational inference, which is discussed in depth in Section 2.1.3.

Optimisation Algorithms

Optimisation algorithms are integral to the training of deep neural networks, serving as the mechanisms through which models learn by minimising a given loss function. While the loss function itself defines the optimisation landscape — a high-dimensional space where each axis corresponds to a model parameters — the role of the optimiser is to effectively navigate this landscape, iteratively moving the model towards a point of minimal loss. This process is non-trivial due to the highly non-convex nature of loss surfaces in deep learning, often characterised by numerous local minima, saddle points, and flat regions, which complicate the convergence to a global minimum (Nocedal and Wright, 2006; Horst et al., 2000).

Local optimisation methods traverse this complex landscape in search of an optimal solution, although by design they can not discern between local and global minima (Figure 2.5) (Horst et al., 2000). Global optimisation methods aim to mitigate this problem by exploring a larger search space in the optimisation landscape, but for the most part, they are largely computationally intractable, and therefore unscalable and not suitable for deep learning tasks. Deep learning optimisation, and non-convex optimisations in general (these can be used interchangeably throughout this subsection), try to achieve a balance between the two by using local optimisation methods that are computationally efficient and can find solutions which are a sufficiently good approximation of the global minimum (LeCun et al., 2015; Goodfellow et al., 2016).

Stochastic Gradient Descent (SGD) is one of the most fundamental optimisation algorithms used in deep learning (Rumelhart et al., 1986). It is a local optimisation algorithm, mathematically expressed as:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}(\theta_t) \tag{2.14}$$

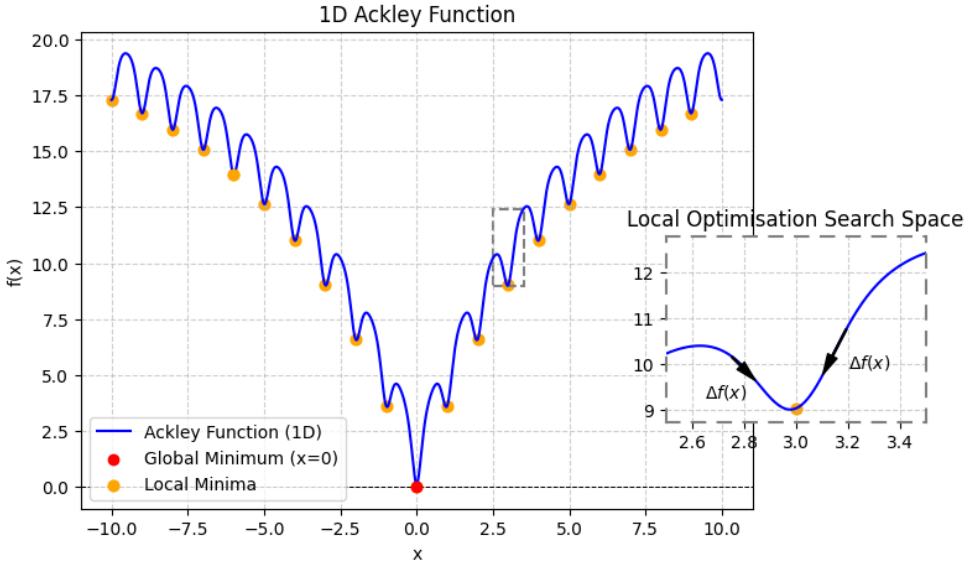


Figure 2.5: The figure depicts the 1D Ackley function (Ackley, 1987), a non-convex objective with multiple local minima and a global minimum at $x = 0$. Local minima (orange) and the global minimum (red) are marked on the primary plot, while the inset zooms in on a local minimum near $x = 2.5$, depicting a narrower optimisation search space that local optimisation algorithms may explore. Under poor optimisation conditions (e.g. initialisations and regularisations), local optimisation algorithms, such a gradient-based ones, may be trapped in local minima, failing to reach the optimal solution. Black arrows indicate the direction of gradient descent at different points along the function.

where θ_t represents the model parameters at iteration t, η is the learning rate, and $\nabla_\theta \mathcal{L}(\theta_t)$ denotes the gradient of the loss function $\mathcal{L}$ with respect to θ_t. As the name implies, this formulation drives the model parameters towards the direction of descent that minimises the loss function. Each model update is performed for a random subset, termed mini-batch, of the training data until the model has converged to a local solution.

This *stochastic* nature of SGD introduces noise into the parameter updates, which can both help and hinder the optimisation process. While this stochasticity may allow the algorithm to escape local minima, it also results in high variance in the updates, which can slow convergence, especially in the presence of mini batches that are not an accurate representation of the whole dataset.

SGD with momentum aims to address the variance and oscillatory behaviour in SGD by incorporating a *velocity* term, which is a dynamically-updated running

average of past gradients (Rumelhart et al., 1986) and is used instead of position to update the parameters (Tieleman and Hinton, 2012). The velocity terms is dynamically updated as:

$$v_{t+1} = \beta v_t + \nabla_\theta \mathcal{L}(\theta_t), \tag{2.15}$$

and the parameter update with the velocity term as:

$$\theta_{t+1} = \theta_t - \eta v_{t+1}, \tag{2.16}$$

where β is a hyperparameter controlling the contribution of previous gradients to the current update. The momentum term helps to accelerate updates in the direction of consistent gradients and dampen oscillations, particularly in the directions where the gradients exhibit high variance (Kingma and Ba, 2014). While this improves convergence speed and stability, it does not resolve issues related to poor scaling of the learning rate across different dimensions of the parameter space, leading to suboptimal performance in scenarios where the loss surface is anisotropic.

RMSprop (Root Mean Square Propagation) introduces an adaptive learning rate mechanism by normalising the gradient updates using an exponentially decaying average of squared past gradients (Tieleman and Hinton, 2012):

$$s_{t+1} = \rho s_t + (1-\rho)\nabla_\theta \mathcal{L}(\theta_t)^2 \tag{2.17}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{s_{t+1} + \epsilon}} \nabla_\theta \mathcal{L}(\theta_t), \tag{2.18}$$

where ρ is the decay rate, and ϵ is a small constant added for numerical stability. The first equation computes an exponentially decaying average of the squared gradients, which acts as an estimate of the second moment (variance) of the gradients. The second equation updates the model parameters using the gradient, but with a learning rate scaled by the square root of the running average of the squared gradient, meaning that parameters with frequently large gradients receive smaller updates, and parameters with smaller gradients receive relatively larger updates (Tieleman and Hinton, 2012).

RMSprop effectively mitigates the issue of learning rate scaling by adapting it for each parameter individually based on the historical magnitude of its gradients. However, RMSprop is limited by its sensitivity to the choice of hyperparameters, particularly the decay rate and the learning rate , and by its lack of bias correction of gradients (Kingma and Ba, 2014). They can also struggle optimising landscapes that exhibits sharp changes in curvature or have sparse gradients, and it may underperform in scenarios with non-stationary objectives (Kingma and Ba, 2014).

Adam (Adaptive Moment Estimation) (Kingma and Ba, 2014) addresses these limitations by combining the benefits of both momentum and RMSprop by maintaining two moving averages: the first for the gradients (Equation 2.19), akin to momentum, and the second for the squared gradients (Equation 2.20), similar to RMSprop (Kingma and Ba, 2014):

$$m_{t+1} = \beta_1 m_t + (1 - \beta_1)\nabla_\theta \mathcal{L}(\theta_t) \tag{2.19}$$

$$s_{t+1} = \beta_2 s_t + (1 - \beta_2)\nabla_\theta \mathcal{L}(\theta_t)^2 \tag{2.20}$$

$$\hat{m}_{t+1} = \frac{m_{t+1}}{1 - \beta_1^{t+1}}, \quad \hat{s}_{t+1} = \frac{s_{t+1}}{1 - \beta_2^{t+1}} \tag{2.21}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{s}_{t+1}} + \epsilon}\hat{m}_{t+1}, \tag{2.22}$$

where the terms β_1 and β_2 are hyperparameters controlling the decay rates of the moving averages, and $\hat{m}_{t+1}$ and $\hat{s}_{t+1}$ are bias-corrected estimates of the first and second moments, respectively, shown in Equation 2.21. The weights are then updated according to Equation 2.22.

Adam's adaptive learning rates, combined with momentum-like behaviour, make it particularly effective in practice, offering robust performance across a wide range of tasks and architectures (Kingma and Ba, 2014). Its ability to handle sparse gradients and maintain stability in the face of non-stationary objectives has contributed to its widespread adoption as the *de facto* optimiser in deep learning. Nonetheless, Adam is not without limitations; it has been observed to converge to suboptimal minima in specific settings where regularisation is critical or when the learning rate is not carefully tuned (Reddi et al., 2019).

Beyond these key optimisers, several other algorithms have been proposed to address specific challenges in deep learning. AdaGrad introduces element-wise scaling of the learning rate based on the cumulative sum of squared gradients, but it suffers from an ever-decreasing learning rate, which can stall optimisation (Duchi et al., 2011). Nadam, a variant of Adam, incorporates Nesterov momentum, which theoretically provides faster convergence but may be more sensitive to hyperparameter tuning (Dozat, 2016). AMSGrad aims to resolve Adam's convergence issues by modifying the second moment estimates to ensure they do not increase, thus preventing the adaptive learning rates from vanishing too rapidly (Reddi et al., 2019). Similarly, AdaMax modifies Adam's update rule to improve stability in the presence of large gradients (Kingma and Ba, 2014). Lastly, LAMB (Layer-wise Adaptive Moments) is designed for large-batch training, particularly effective in the context of training large-scale language models, by adapting the learning rate at the layer level, addressing issues of poor conditioning across different network depths (You et al., 2020).

2.1.3 Variational Autoencoders

Variational inference, a method rooted in Bayesian statistics, provides an efficient alternative to computationally expensive techniques like Markov Chain Monte Carlo (MCMC) for approximating intractable posterior distributions (Goodfellow et al., 2016). It reformulates inference as an optimisation problem by iteratively learning a variational distribution $q_\theta(\mathbf{z} \mid \mathbf{x})$ that approximates the intractrable true posterior $p_\phi(\mathbf{z} \mid \mathbf{x})$. This approach enables scalable inference for large datasets and has become fundamental in training latent variable models.

Traditional autoencoders, composed of an encoder $\mathcal{E}_\theta$ and a decoder $\mathcal{D}_\phi$, learn to map input data $\mathbf{x}$ to a latent representation $\mathbf{z} = \mathcal{E}_\theta(\mathbf{x})$ and reconstruct it as $\tilde{\mathbf{x}} = \mathcal{D}_\phi(\mathbf{z})$. While effective for dimensionality reduction and data compression, these deterministic mappings are ill-suited for generative tasks, as they lack a probabilistic framework to model the underlying data distribution.

Variational Autoencoders (VAEs) address this limitation by introducing a probabilistic latent variable model trained via variational inference (Kingma and Welling,

2013). In a VAE, the encoder $q_\theta(\mathbf{z} \mid \mathbf{x})$ maps $\mathbf{x}$ to the parameters of a latent distribution, parametereised as a multi-variate Gaussian of a mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}^2$. A sample $\mathbf{z}$ is then drawn from this distribution, $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$, and passed to the decoder $p_\phi(\mathbf{x} \mid \mathbf{z})$, which reconstructs $\tilde{\mathbf{x}}$. This framework allows VAEs to generate new data samples by sampling latent variables from a known prior $p(\mathbf{z})$ and passing them through the decoder to retrieve its data-space representation. The probabilistic framework not only introduces generative capabilities to autoencoders, but also allows for smooth interpolation in the latent space, enabling the generation of diverse and coherent samples (Goodfellow et al., 2016; Kingma and Welling, 2013).

Probabilistic Framework of VAEs

In any probabilistic MLE framework, the objective is to maximise the likelihood of generating real samples $\mathbf{x}$, which is equivalent to minimising the negative log-likelihood of the data distribution:

$$\phi^* = \arg\min_\phi - \sum_{i=1}^{n} \log p_\phi(\mathbf{x}^{(i)}), \tag{2.23}$$

where $p_\phi(\mathbf{x})$ is the likelihood of the data under the parameterised model distribution p_ϕ. For models with latent variables $\mathbf{z}$, the likelihood $p_\phi(\mathbf{x})$ is expressed as a marginalisation over the latent space:

$$p_\phi(\mathbf{x}) = \int p_\phi(\mathbf{x} \mid \mathbf{z})\, p(\mathbf{z})\, d\mathbf{z}, \tag{2.24}$$

where $p_\phi(\mathbf{x} \mid \mathbf{z})$ is the likelihood of the observed data given the latent variable $\mathbf{z}$ and $p(\mathbf{z})$ is the prior distribution over the latent space. The integral sums over all possible configurations of $\mathbf{z}$, weighting each configuration of the latent variables by its prior probability $p(\mathbf{z})$. Conceptually, this term quantifies how well the latent variable $\mathbf{z}$ explains the observed data $\mathbf{x}$, under parameterisation by ϕ.

The need for integrating over all possible values of $\mathbf{z}$ to evaluate $p_\phi(\mathbf{x})$, makes computing this marginal likelihood intractable for high-dimensional latent spaces. The key insight of VAEs is to approximate this intractable integral by introducing a variational distribution $q_\theta(\mathbf{z} \mid \mathbf{x})$ that serves as a tractable approximation to the

true posterior $p_\phi(\mathbf{z} \mid \mathbf{x})$ (Kingma and Welling, 2013), which in turn, as will be seen in the following derivations, enables the estimation of the likelihood $p_\phi(\mathbf{x})$ without the high dimensionality.

The variational distribution is parameterised by θ and is optimised to match the true posterior as closely as possible. By multiplying and dividing by $q_\theta(\mathbf{z} \mid \mathbf{x})$ within the integral in Equation 2.24, the marginal likelihood can be rewritten as:

$$p_\phi(\mathbf{x}) = \int q_\theta(\mathbf{z} \mid \mathbf{x}) \frac{p_\phi(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z})}{q_\theta(\mathbf{z} \mid \mathbf{x})} \, d\mathbf{z}. \tag{2.25}$$

Taking the logarithm of the marginal likelihood and substituting the above expression, yields:

$$\log p_\phi(\mathbf{x}) = \log \int q_\theta(\mathbf{z} \mid \mathbf{x}) \frac{p_\phi(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z})}{q_\theta(\mathbf{z} \mid \mathbf{x})} \, d\mathbf{z}. \tag{2.26}$$

Using Jensen's inequality (Jensen, 1906), which states that $\log \mathbb{E}[f(x)] \geq \mathbb{E}[\log f(x)]$, we derive a tractable lower bound for $\log p_\phi(\mathbf{x})$:

$$\log p_\phi(\mathbf{x}) \geq \int q_\theta(\mathbf{z} \mid \mathbf{x}) \log \frac{p_\phi(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z})}{q_\theta(\mathbf{z} \mid \mathbf{x})} \, d\mathbf{z}. \tag{2.27}$$

Factoring out $p_\phi(\mathbf{x} \mid \mathbf{z})$ from the logarithm with $\log ab = \log a + \log b$, and using the property $\log \frac{a}{b} = -\log \frac{b}{a}$, we obtain:

$$\log p_\phi(\mathbf{x}) \geq \int q_\theta(\mathbf{z} \mid \mathbf{x}) \left[\log p_\phi(\mathbf{x} \mid \mathbf{z}) - \log \frac{q_\theta(\mathbf{z} \mid \mathbf{x})}{p(\mathbf{z})} \right] d\mathbf{z}. \tag{2.28}$$

Expanding the integral, we have:

$$\log p_\phi(\mathbf{x}) \geq \int q_\theta(\mathbf{z} \mid \mathbf{x}) \log p_\phi(\mathbf{x} \mid \mathbf{z}) \, d\mathbf{z} - \int q_\theta(\mathbf{z} \mid \mathbf{x}) \log \frac{q_\theta(\mathbf{z} \mid \mathbf{x})}{p(\mathbf{z})} \, d\mathbf{z}, \tag{2.29}$$

where the first term represents the expected log-likelihood of the data $\mathbf{x}$ given the latent variable under the approximate posterior $q_\theta(\mathbf{z} \mid \mathbf{x})$, quantifying how well the latent variable $\mathbf{z}$, sampled from the approximate posterior, explains the observed data $\mathbf{x}$ when passed through the probabilistic decoder $p_\phi(\mathbf{x} \mid \mathbf{z})$. The second term is the KL divergence (Equation 2.13) between the approximate posterior $q_\theta(\mathbf{z} \mid \mathbf{x})$

and the prior $p(\mathbf{z})$. In full, Equation 2.29 is known as the Evidence Lower Bound (ELBO), and its minimisation leads to maximising the marginal likelihood of the data under the model (Kingma and Welling, 2013). Minimising the ELBO is the driving force behind VAEs, compactly expressed as:

$$\mathcal{L}_{VAE}(\mathbf{x};\theta,\phi) = -\mathbb{E}_{q_\theta(\mathbf{z}|\mathbf{x})}\left[\log p_\phi(\mathbf{x} \mid \mathbf{z})\right] + \mathrm{KL}\left(q_\theta(\mathbf{z} \mid \mathbf{x}) \parallel p(\mathbf{z})\right). \tag{2.30}$$

The expectation term of Equation 2.30 is often referred to as the *reconstruction term* in the VAE framework, and is the same term minimised by non-probabilistic autoencoders, usually measured under an element-wise norm like MSE. In turn, the KL divergence measures the "distance" between the learned approximate posterior $q_\theta(\mathbf{z} \mid \mathbf{x})$ and the prior $p(\mathbf{z})$, ensuring that these two distributions are close to each other. This term acts as a regulariser, structuring the latent space by encouraging it to conform to a distribution of our choice that supports interpolation and generalisation (Goodfellow et al., 2016; Kingma and Welling, 2013). In summary, these terms ensure both generative performance and latent space regularity.

The choice of prior $p(\mathbf{z})$ in VAEs is typically a unit Gaussian distribution $\mathcal{N}(\mathbf{0},\mathbf{I})$, as this facilitates the analytical computation of the KL divergence when the approximate posterior also takes the form of a Gaussian. For a Gaussian prior $p(\mathbf{z}) = \mathcal{N}(\mathbf{0},\mathbf{I})$ and approximate posterior $q_\theta(\mathbf{z} \mid \mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_\theta(\mathbf{x}), \boldsymbol{\sigma}_\theta(\mathbf{x})^2\mathbf{I})$ modelled as a diagonal Gaussian, the closed form of the KL divergence is obtained by inserting their respective probability density functions in Equation 2.13, yielding:

$$\mathrm{KL}\left(q_\theta(\mathbf{z} \mid \mathbf{x}) \parallel p(\mathbf{z})\right) = \frac{1}{2}\sum_{j=1}^{J}\left(1 - \log(\boldsymbol{\sigma}_\theta(\mathbf{x})_j^2) + \boldsymbol{\mu}_\theta(\mathbf{x})_j^2 + \boldsymbol{\sigma}_\theta(\mathbf{x})_j^2\right) + C, \tag{2.31}$$

where J is the dimensionality of the latent space and C are constants independent of the parameterisation. This tractable form ensures that the KL divergence term in the VAE objective can be computed efficiently, making the estimation of $p_\phi(\mathbf{x})$ scalable and practical for training.

The VAE is constructed such that the encoder network outputs the parameters of the approximate posterior, $\boldsymbol{\mu}_\theta(\mathbf{x})$ (mean) and $\boldsymbol{\sigma}_\theta(\mathbf{x})$ (standard deviation), which are used to evaluate the KL divergence. The latent variable $\mathbf{z}$ is sampled using the reparameterisation trick:

$$\mathbf{z} = \boldsymbol{\mu}_\theta(\mathbf{x}) + \boldsymbol{\sigma}_\theta(\mathbf{x}) \odot \boldsymbol{\epsilon}, \tag{2.32}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\odot$ denotes element-wise multiplication. This reformulation makes the sampling operation differentiable, enabling backpropagation and gradient-based optimisation to maximise the ELBO (Kingma and Welling, 2013). The sample $\mathbf{z}$ is then passed to the decoder network, which reconstructs the input data $\tilde{\mathbf{x}}$ to evaluate the reconstruction term of the lower bound.

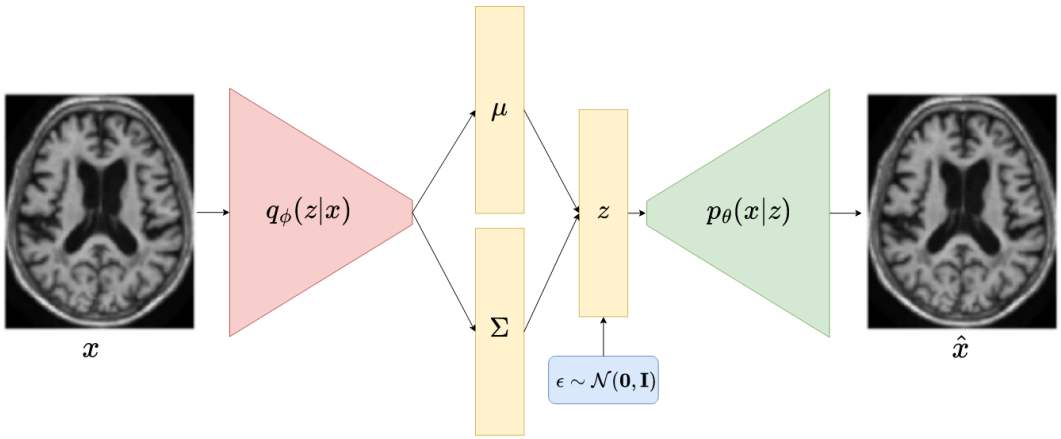


Figure 2.6: Schematic representation of a Variational Autoencoder (VAE). The encoder network maps the input data $\mathbf{x}$ to the parameters of the approximate posterior that defines latent distribution $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, from which a sample $\mathbf{z}$ is drawn. The decoder network then reconstructs the input data from the latent sample. Extracted directly from Kebaili et al. (2023) — notation in the figure differs slightly from that used in the text.

While the assumption of a Gaussian posterior with a diagonal covariance matrix simplifies computation, it limits expressiveness by preventing the model from capturing correlations between latent dimensions. Similarly, the Gaussian prior $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ enforces an isotropic latent space, which supports smooth interpola-

tion but may misalign with the data's true posterior. These design choices highlight the trade-off between computational tractability and the ability to model complex data distributions in VAEs.

The schematic representation of a VAE is shown in Figure 2.6 and its training algorithm is outlined in Algorithm 2. Generating new samples from a trained VAE involves sampling latent variables from the prior distribution $p(\ddagger)$ and passing them through the decoder network to generate new data points, where its latent space can be explored to manipulate the generative process.

Algorithm 2 Training Algorithm for VAEs

Input: Training data distribution $q(\mathbf{x})$
Output: Encoder $\mathcal{E}_\theta$ and Decoder $\mathcal{D}_\phi$ networks
Initialise encoder parameters θ and decoder parameters ϕ
repeat
  for number of training iterations **do**
    Step 1: Encode the data sample
    Sample $\mathbf{x} \sim q(\mathbf{x})$
    Obtain latent parameters:

$$\boldsymbol{\mu}_\theta, \boldsymbol{\sigma}_\theta \leftarrow \mathcal{E}_\theta(\mathbf{x}) \tag{2.33}$$

    Sample latent variable:

$$\mathbf{z} = \boldsymbol{\mu}_\theta(\mathbf{x}) + \sigma_\theta(\mathbf{x}) \odot \boldsymbol{\epsilon} \tag{2.34}$$

    Step 2: Decode and reconstruct the data

$$\tilde{\mathbf{x}} \leftarrow \mathcal{D}_\phi(\mathbf{z}) \tag{2.35}$$

    Step 3: Compute the total loss

$$\mathcal{L} = ||\mathbf{x} - \tilde{\mathbf{x}}||_2^2 + \mathrm{KL}(q(\mathbf{z} \mid \mathbf{x}) \parallel p(\mathbf{z})) \tag{2.36}$$

    Step 4: Update model parameters

$$\theta, \phi \leftarrow \theta, \phi - \alpha \nabla_{\theta,\phi} \mathcal{L} \tag{2.37}$$

  end for
until convergence of encoder and decoder

VAEs in Practice

VAEs have been widely applied across various domains, including dimensionality reduction, data generation, and the manipulation of latent spaces (Kingma and

Welling, 2013; Higgins et al., 2017; Berthelot et al., 2018; Rezende et al., 2014). However, a common issue with VAEs, and variational generative models in general, is posterior collapse, where the encoder neglects the latent variables, resulting in poor generative performance. This issue is particularly prevalent when the decoder is overly powerful or the prior is overly simplistic or weak, leading the model to maximise the likelihood of the data without leveraging the latent space effectively (Razavi et al., 2019a; Lucas et al., 2019). Additionally, VAEs often struggle to generate high-quality, sharp images on large datasets compared to more sophisticated generative models, primarily due to the smoothness imposed by the Gaussian regularisation of the latent space and the limited expressiveness of the approximate posterior (Goodfellow et al., 2016; Bond-Taylor et al., 2021; Zhao et al., 2017). These limitations, however, can be mitigated by augmenting the reconstruction loss with perceptual losses (Rombach et al., 2022). Furthermore, the KL divergence term can sometimes dominate the training process, resulting in suboptimal latent spaces that fail to effectively capture the underlying data structure (Kingma and Welling, 2013).

To address these challenges, various VAE variants have been proposed. β-VAEs introduces a hyperparameter β that adjusts the balance between the reconstruction loss and the KL divergence, promoting the learning of disentangled latent representations (Higgins et al., 2017). Vector Quantised VAEs (VQ-VAEs) replace the continuous latent space with a discretised representation, demonstrating effectiveness in tasks like discrete data generation and speech synthesis (Razavi et al., 2019b). Very deep VAEs utilise hierarchical latent variable models to increase the expressiveness of the latent space, enabling the generation of more complex and diverse data (Child, 2020).

2.1.4 Generative Adversarial Networks

Generative Adversarial Networks (GANs) have emerged as a groundbreaking framework in deep learning, revolutionising generative modelling techniques. Introduced by Goodfellow et al. (2014), GANs represent a departure from traditional generative models by employing an adversarial training framework that fundamentally redefined data generation (Goodfellow et al., 2014). Unlike VAEs and MCMC methods that

relied on statistical assumptions and required expensive computations, GANs leverage a competitive interaction between two neural networks to directly sample from the underlying distribution of the training data, eliminating the need for explicit density estimation (Salimans et al., 2016).

At the core of GANs lies the adversarial training mechanism, which establishes a dynamic interplay between a generator network G and a discriminator network D. The generator learns to map samples from a latent variable $\mathbf{z} \sim p_z(\mathbf{z})$, drawn from a prior distribution (commonly Gaussian or uniform), to a data space, producing synthetic samples $G(\mathbf{z}) \sim p_G(\mathbf{x})$. The discriminator, on the other hand, maps data samples to a probability, $D(\mathbf{x})$, which represents its confidence that a given sample $\mathbf{x}$ originates from the true data distribution $p_{\text{data}}(\mathbf{x})$ rather than the generator's distribution $p_G(\mathbf{x})$. This adversarial framework creates a minimax game, where the generator aims to produce data that maximises the discriminator's error while driving $p_G(\mathbf{x}) \approx p_{\text{data}}(\mathbf{x})$, while the discriminator seeks to correctly distinguish between real and generated samples. This interplay is defined by the adversarial loss given by:

$$\min_G \max_D \mathcal{L}_{\text{adv}} = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))], \tag{2.38}$$

where the generator and discriminator co-train iteratively as described in Algorithm 3 and depicted in Figure 2.7.

The first term $\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x})]$ of Equation 2.38 represents the expectation over all real data samples $\mathbf{x}$ drawn from the true data distribution $p_{\text{data}}(\mathbf{x})$, where $D(\mathbf{x})$ denotes the discriminator's estimate of the probability that the sample $\mathbf{x}$ is a real data point rather than one generated by the model. The logarithm of $D(\mathbf{x})$ is employed to reward the discriminator for correctly classifying real data samples as genuine, with the expectation encouraging accurate identification over the entire dataset. The second term $\mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))]$ corresponds to the expectation over the latent variable $\mathbf{z}$, which is sampled from a prior distribution $p_z(\mathbf{z})$. Here, $G(\mathbf{z})$ refers to the generator's output, which aims to produce data samples that resemble those from the real data distribution. $D(G(\mathbf{z}))$ represents the discrim-

Algorithm 3 Training Algorithm for GANs — extracted from Goodfellow et al. (2014)

Input: Training data distribution $p_{\text{data}}(\mathbf{x})$, noise distribution $p_z(\mathbf{z})$
Output: Generator G and Discriminator D networks
Initialise generator parameters θ_g and discriminator parameters θ_d
repeat
  for number of training iterations **do**
    Step 1: Train the Discriminator D
    for number of discriminator steps k **do**
      Sample m noise samples $\{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}\} \in p_z(\mathbf{z})$
      Sample m real data samples $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\} \in p_{\text{data}}(\mathbf{x})$
      Compute the discriminator loss:

$$\mathcal{L}_D = -\frac{1}{m}\sum_{i=1}^{m}\left[\log D(\mathbf{x}^{(i)}) + \log(1 - D(G(\mathbf{z}^{(i)})))\right] \tag{2.39}$$

      Update the discriminator by ascending the gradient:

$$\theta_d \leftarrow \theta_d + \alpha\nabla_{\theta_d}\mathcal{L}_D \tag{2.40}$$

    end for
    Step 2: Train the Generator G
    Sample m noise samples $\{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}\} \in p_z(\mathbf{z})$
    Compute the generator loss:

$$\mathcal{L}_G = -\frac{1}{m}\sum_{i=1}^{m}\log D(G(\mathbf{z}^{(i)})) \tag{2.41}$$

    Update the generator by descending the gradient:

$$\theta_g \leftarrow \theta_g - \alpha\nabla_{\theta_g}\mathcal{L}_G \tag{2.42}$$

  end for
until convergence of the generator and discriminator

inator's estimate of the probability that the generated sample $G(\mathbf{z})$ is real, and the logarithm of $1 - D(G(\mathbf{z}))$ penalises the generator for producing samples that the discriminator can easily distinguish from real data. This encourages the generator, which minimises the loss, to improve the quality of its outputs, making them increasingly indistinguishable from real data and thus more likely to fool the discriminator. At the Nash equilibrium of this game (Osborne and Rubinstein, 1994), the generator has ideally learned to generate data that is indistinguishable from the real data distribution, and the discriminator is no longer able to discern generated samples from real ones with any better accuracy than random guessing. This equilibrium

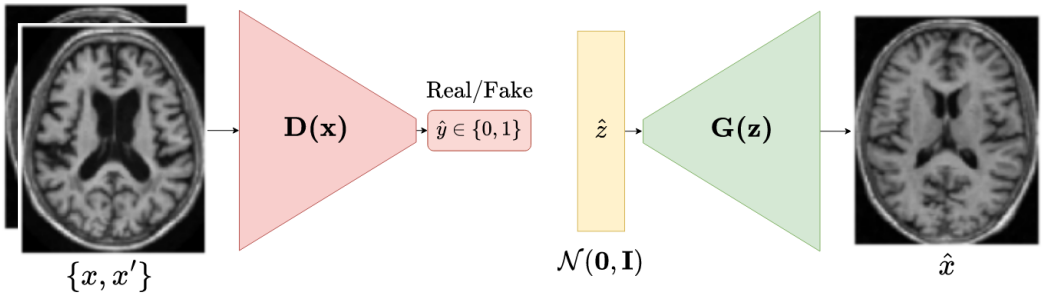


Figure 2.7: Schematic representation of a Generative Adversarial Network (GAN). The generator network G maps a noise vector $\mathbf{z}$ to a data sample $G(\mathbf{z})$, while the discriminator network D evaluates whether a given sample is real or generated. The adversarial training process iteratively updates both networks to improve the quality of generated samples and the discriminator's ability to distinguish them from real data. Extracted directly from Kebaili et al. (2023) — notation in the figure differs slightly from that used in the text.

represents the point where the generator and discriminator are optimally balanced, and the model's performance stabilises (Goodfellow et al., 2014; Weng, 2017).

In a statistical framework, when the discriminator is at its optimal performance, minimising the adversarial loss reduces to minimising the Jensen-Shannon (JS) divergence between the generated data distribution $p_g(\mathbf{x})$ and the real data distribution $p_{\text{data}}(\mathbf{x})$. At each training iteration, given a fixed generator G, the optimal discriminator D^* can be derived by maximising Equation 2.38 pointwise for each input sample. Taking the derivative of Equation 2.38 with respect to $D(\mathbf{x})$ and setting it to zero yields:

$$D^*(\mathbf{x}) = \frac{p_{\text{data}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_G(\mathbf{x})}. \tag{2.43}$$

Substituting $D^*(\mathbf{x})$ back into the Equation 2.38 gives the generator's objective under the assumption of an optimal discriminator. The generator minimises:

$$\mathcal{L}_G = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}\left[\log \frac{p_{\text{data}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_G(\mathbf{x})}\right] + \mathbb{E}_{\mathbf{x}\sim p_G(\mathbf{x})}\left[\log \frac{p_G(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_G(\mathbf{x})}\right], \tag{2.44}$$

which can be written as the JS divergence between the two distributions $p_{\text{data}}(\mathbf{x})$ and $p_G(\mathbf{x})$ (Goodfellow et al., 2014):

$$\mathcal{L}_G = -\log(4) + 2 \cdot \text{JS}(p_{\text{data}}(\mathbf{x}) \| p_g(\mathbf{x})), \tag{2.45}$$

with JS divergence is defined as:

$$\begin{aligned} \text{JS}(p_{\text{data}}(\mathbf{x}) \| p_g(\mathbf{x})) = & \frac{1}{2} \text{KL} \left(p_{\text{data}}(\mathbf{x}) \| \frac{p_{\text{data}}(\mathbf{x}) + p_g(\mathbf{x})}{2} \right) \\ & + \frac{1}{2} \text{KL} \left(p_g(\mathbf{x}) \| \frac{p_{\text{data}}(\mathbf{x}) + p_g(\mathbf{x})}{2} \right) \end{aligned} \tag{2.46}$$

where $\text{KL}(p \,||\, \frac{p+q}{2})$ and $\text{KL}(p \,||\, \frac{p+q}{2})$ are the KL divergences between the two distributions p and q and their average, respectively. The JS divergence, although not differentiable, is symmetric and bounded, with a value of 0 when the two distributions are identical. Thus, minimising the adversarial loss with an optimal discriminator is equivalent to minimising the JS divergence between the true data distribution $p_{\text{data}}(\mathbf{x})$ and the generator's distribution $p_G(\mathbf{x})$. At the Nash equilibrium, when $p_G(\mathbf{x}) = p_{\text{data}}(\mathbf{x})$, the Jensen-Shannon divergence reaches its minimum value of zero, implying that the generator has perfectly learned to replicate the true data distribution.

Adversarial losses, particularly when applied to GANs, have revolutionised deep image generation tasks by introducing a powerful framework for producing highly realistic and detailed images. This approach addresses the limitations of traditional loss functions by driving the generation of images to closely mimic the complex characteristics of real-world data, modelling the data distribution indirectly with equal focus in all perceptual qualities (Goodfellow et al., 2014; Heusel et al., 2017; Bińkowski et al., 2018; Wang et al., 2018; Isola et al., 2017; Arjovsky et al., 2017; Mao et al., 2017).

GANs in Practice

GANs have found widespread application in various fields, particularly in the synthesis of data across different modalities such as images, text, and audio. They have proven instrumental in generating realistic images, which has been leveraged in areas ranging from artistic creation to medical imaging (Goodfellow et al., 2014; Alqahtani et al., 2021; Kazeminia et al., 2020). Another prominent application of GANs is domain transfer, where data is converted from one domain to another, such as translating photographs into artistic renditions. Super-resolution, which involves generating high-resolution images from low-resolution counterparts, has also seen significant advancements through GANs in applications like satellite imagery and medical diagnostics (Alqahtani et al., 2021; Kazeminia et al., 2020). GANs are also employed in interpolation tasks, generating intermediate frames in sequences of images or videos, and in data augmentation, where they enhance the diversity of training datasets, thereby improving the performance of machine learning models (Goodfellow et al., 2014; Salimans et al., 2016).

Despite their successes, GANs are accompanied by notable challenges and limitations. One of the primary issues with GANs is mode collapse, where the generator learns to produce a limited set of samples that fool the discriminator, rather than capturing the full diversity of the data distribution (Creswell et al., 2018). Despite its statistical framework, it is stated by the adversarial loss that its only goal is to fool the discriminator, not to capture the full distribution of the data. This can lead to a lack of variety in the generated images and a failure to reproduce the distribution data to a sufficient degree (Goodfellow et al., 2014). Additionally, GANs require careful hyperparameter tuning and architectural choices to properly balance the work of the generator vs. the discriminator and achieve Nash equilibrium (Creswell et al., 2018), a task that is particularly difficult to achieve. The stability of the adversarial training process can be highly sensitive to the choice of learning rates, optimisers, and network architectures, making it challenging to apply in practice. Furthermore, GANs are known to suffer from issues such as vanishing gradients, occurring when the discriminator becomes excessively adept at distinguishing real from generated

data, leading to insufficient gradient information for updating the generator, stagnating the training process (Goodfellow et al., 2014; Weng, 2017). Altogether, these issues can hinder their performance and limit the applicability of GANs in real-world scenarios (Creswell et al., 2018; Salimans et al., 2016).

Addressing these limitations has been a primary focus of GAN research, resulting in the development of various techniques to stabilise training and enhance performance, including alternative loss functions and architectural modifications (Arjovsky and Bottou, 2017; Salimans et al., 2016). Wasserstein GANs (WGANs) (Arjovsky et al., 2017) and Least Squares GANs (LSGANs) (Mao et al., 2017) are notable examples of such advancements, offering smoother loss functions compared to the discontinuous JS divergence. These approaches aim to improve stability and convergence while mitigating challenges such as mode collapse and vanishing gradients (Arjovsky et al., 2017; Mao et al., 2017). WGANs leverage the Wasserstein distance as the loss function, whereas LSGANs employ the least squares loss (or MSE) loss.

The impressive capabilities of the GAN framework have driven the development of numerous variants. For instance, Conditional GANs (cGANs) extend the basic GAN framework by incorporating additional information, such as class labels, into the generation process, enabling the creation of class-specific data (Mirza and Osindero, 2014). StyleGANs have advanced state-of-the-art image synthesis by allowing detailed control over the latent space to manipulate the style of generated images, making them highly valuable in industries requiring photorealistic image generation (Karras et al., 2020). Variational Autoencoder GANs (VAE-GANs) combine the probabilistic framework of VAEs with adversarial training to overcome the lack of expressiveness of VAEs and produce higher-quality samples (Larsen et al., 2016). CycleGANs have transformed image-to-image translation tasks by enabling domain conversion without requiring paired datasets, greatly impacting applications in style transfer and domain adaptation (Zhu et al., 2017). InfoGANs, meanwhile, enhance interpretability and control in generative processes by maximising mutual information between a subset of latent variables and the generated data, promoting disentangled, meaningful representations (Chen et al., 2016).

The adaptability of GANs is further demonstrated by the ease with which ad-

versarial training can be integrated into diverse network architectures. By adding a discriminator network and an adversarial loss function, many modern models harness the power of GANs (Larsen et al., 2016; Rombach et al., 2022), establishing adversarial training as a foundational technique in the machine learning community.

2.1.5 Diffusion Models

Diffusion models have recently emerged as a robust framework in generative modelling, marking a significant shift from earlier probabilistic and adversarial architectures. Originally formalised by Sohl-Dickstein et al. (2015) and later refined by Ho et al. (2020), these models rely on a sequential denoising process to transform simple Gaussian noise into complex, data-like samples. Unlike GANs and VAEs, which utilise adversarial training or probabilistic latent variable approximations, respectively (Goodfellow et al., 2014; Kingma and Welling, 2013), diffusion models progressively add noise to data in a forward process and then reverse this corruption through a parameterised and iterative denoising process (Ho et al., 2020; Song and Ermon, 2019). By circumventing adversarial training, diffusion models avoid the instability and mode collapse issues prevalent in GANs; and by not imposing a regularised latent space, they are free from the assumptions and limited expressiveness associated with VAEs. Instead, the iterative noise-removal approach enables diffusion models to capture intricate data distributions with high fidelity, employing a sequence of Markov steps that can effectively approximate the data manifold (Ho et al., 2020; Podell et al., 2023).

Diffusion models are inspired by non-equilibrium thermodynamics, where systems evolve towards a target distribution through gradual perturbations that increase entropy over time (Sohl-Dickstein et al., 2015). The forward process transforms structured data into Gaussian noise through stochastic perturbations governed by continuous-time SDEs, progressively degrading the data distribution into a high-entropy Gaussian state (Song et al., 2021b). The forward diffusion process is formally described by the forward SDE:

$$\mathrm{d}\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}_t, \tag{2.47}$$

where $\mathbf{x}_t$ represents the data state at time t, $\mathbf{f}(\mathbf{x}_t, t)$ is the drift term, $g(t)$ is the diffusion term, and $\mathbf{w}_t$ is the Wiener process representing Brownian motion. Equation 2.47 can be understood by its components. The drift term acts on $\mathbf{x}_t$ to apply a steady deterministic shift, moving data from a structured state towards Gaussian noise; and the diffusion term controls the level of randomness added at each step, which, governed by the Wiener process, introduces natural variability to the equation through stochasticity (Oksendal, 2013). Together, the drift and diffusion terms jointly transform the data towards Gaussian noise as $t \to T$.

The goal of a diffusion model is to learn the reverse process that reconstructs the data by approximating the reverse-time dynamics of the forward SDE. For a stochastic process described by a forward SDE, the reverse-time process is defined by modifying the drift term to include a corrective term proportional to the gradient of the log-likelihood of the data distribution (Anderson, 1982; Song et al., 2021b; Sohl-Dickstein et al., 2015). Consequently, the reverse-time process is itself another SDE, enabling the iterative recovery of structured data from noise. In combination with the Fokker-Planck equation defined as:

$$\frac{\partial p_t(\mathbf{x})}{\partial t} = -\nabla \cdot (\mathbf{f}(\mathbf{x}, t) p_t(\mathbf{x})) + \frac{1}{2} g(t)^2 \nabla^2 p_t(\mathbf{x}), \tag{2.48}$$

which describes how the data distribution evolves as a function of drift and diffusion terms, the reverse SDE takes the form (Song et al., 2021b; McAllester, 2023):

$$\mathrm{d}\mathbf{x}_t = \left[\mathbf{f}(\mathbf{x}_t, t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)\right] \mathrm{d}t + g(t)\, \mathrm{d}\mathbf{w}_t, \tag{2.49}$$

where $p_t(\mathbf{x}_t)$ is the probability density of the corrupted sample $\mathbf{x}_t$ at time t, and $\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)$ is the score function. Equation 2.49 can also be understood by its components. The backward drift term $\mathbf{f}(\mathbf{x}_t, t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)$ acts as a force that guides the reverse process towards the original data distribution by combining the original forward drift term and the score function, the latter of which provides the gradient direction towards regions of higher probability in the data distribution (Oksendal, 2013; Song et al., 2021b). The term $g(t)\, \mathrm{d}\mathbf{w}_t$ represents the Gaussian noise added at each step, which, similarly to the forward SDE, introduces stochasticity

into the reverse process, allowing for stability and variability in the denoising trajectory. Without this term, the reverse process would deterministically map noise to data in the form of an ordinary differential equation (ODE) (Oksendal, 2013; Song et al., 2021b; Ho et al., 2020). Together, these terms allow the reverse-time SDE to reconstruct structured data from Gaussian noise, effectively generating samples by following the reverse diffusion pathway back to the initial data distribution (Ho et al., 2020).

Score-Based Generative Modelling

In theory, if the gradient of the log-likelihood of the data is known, Equation 2.49 can be approximated using *Langevin dynamics* to sample from the data distribution in discrete steps. Langevin dynamics provides a discrete approximation to the continuous reverse SDE and mathematically operates as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \eta \nabla_{\mathbf{x}} \log p(\mathbf{x}_k) + \sqrt{2\eta}\, \boldsymbol{\epsilon}_k, \tag{2.50}$$

where η is a small step size (analogous to $\mathrm{d}t$ in the SDE), $\nabla_{\mathbf{x}} \log p(\mathbf{x}_k)$ is the score function, providing the gradient direction at the current step k, and $\boldsymbol{\epsilon}_k$ is standard Gaussian noise sampled independently at each step. In this formulation, the term $\eta \nabla_{\mathbf{x}} \log p(\mathbf{x}_k)$ leverages the score function to direct the sample towards high-density regions of the data distribution, similar to gradient descent (Figure 2.8). The term $\sqrt{2\eta}\, \boldsymbol{\epsilon}_k$ adds stochastic noise, akin to the Brownian motion component $g(t)\, \mathrm{d}\mathbf{w}_t$ in the reverse SDE, stabilising the sampling process and ensuring sampling variety (Song and Ermon, 2019). It is important to note that convergence of Langevin dynamics to the target distribution depends on a sufficiently small step size η and the number of steps taken.

While Langevin dynamics may appear as a crude approximation to the continuous reverse SDE, it proves effective for several reasons. First, the score function guidance at each step provides a strong directional cue toward higher-probability regions, reducing reliance on a finely tuned continuous path. Additionally, using a small step size η ensures that each update closely follows the behaviour of the continuous re-

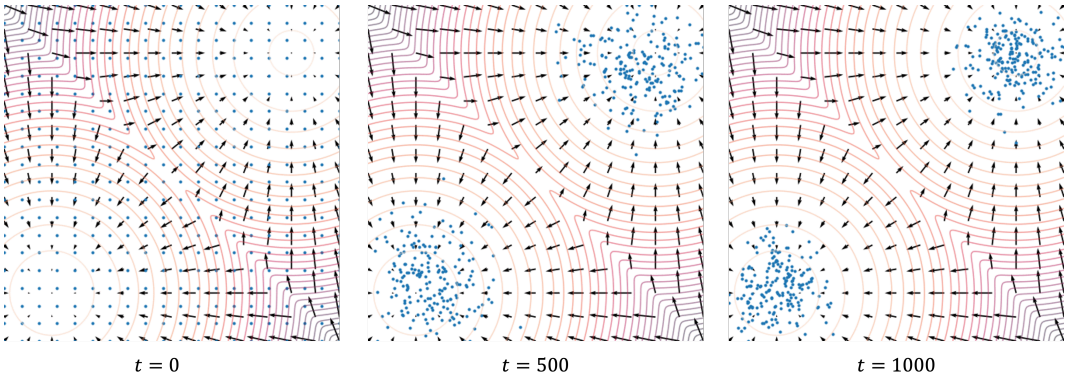


Figure 2.8: Evolution of the sampling process, represented by blue-dotted particles over time under Langevin dynamics, as described by Equation (2.50). Each panel represents successive timesteps, showing particles converging towards regions of high probability density. The vector field (black arrows) denotes the gradient of the log-density function, directing particles towards modes of the target distribution. Red contours illustrate the energy landscape, with particles initially dispersed and progressively concentrating in lower-potential regions. Noise, modelled by the term $\sqrt{2\eta}\boldsymbol{\epsilon}_k$, introduces stochasticity, ensuring exploration of the state space. Image adapted from Song (2021)

verse SDE, allowing the discrete steps to approximate the trajectory accurately with sufficient iterations. However, excessively small step sizes can lead to slower convergence, requiring a careful balance to maintain efficiency. Furthermore, the iterative addition of noise facilitates exploration of the probability landscape, preventing the sampler from becoming trapped in local modes (Song and Ermon, 2019). *Annealed Langevin dynamics*, which gradually reduces the noise level, is often employed to enable broad exploration initially and fine-grained adjustments later, further stabilising the sampling process (Song and Ermon, 2019). Conceptually, score-based generative models, and diffusion models more broadly, iteratively denoise pure noise through a series of steps, effectively "diffusing" the noise distribution towards the real data distribution (Figure 2.9).

In practice, the score function is intractable to compute directly because it requires knowledge of the true data distribution $p_t(\mathbf{x}_t)$. Song and Ermon (2019) instead proposes that a neural network can be used to approximate the score function $\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)$, allowing for the reverse SDE path to be resolved and to generate new samples from the data distribution. This approach is known as score-based generative modelling. In Song and Ermon (2019), data degradation is implemented

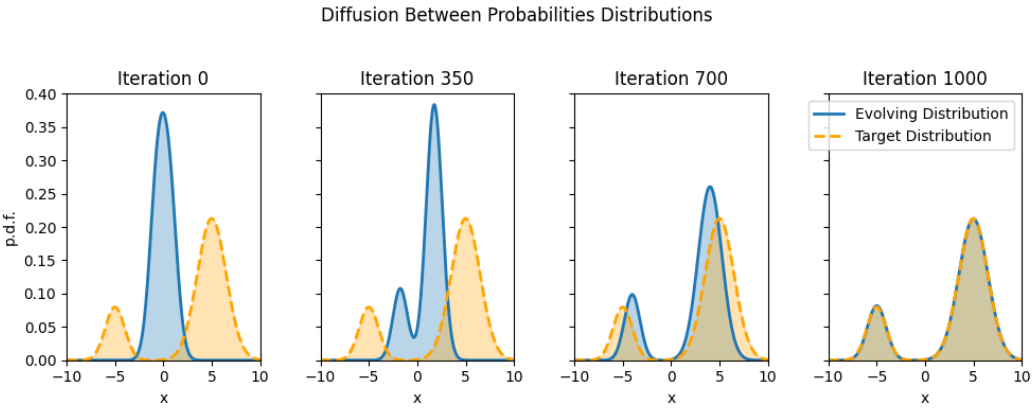


Figure 2.9: Progressive transformation of an initial unimodal Gaussian distribution into a bimodal target distribution over 1000 iterations, employing a cosine annealing schedule to control the iterative degradation. This process reflects the underlying principles of score-based and diffusion models, where probability distributions are iteratively refined by perturbing initial samples towards a more complex target distribution. The evolving distribution (solid line) increasingly aligns with the target distribution (dashed line), demonstrating the gradual improvement in density estimation characteristic of generative models that leverage forward and reverse diffusion processes. Image inspired by Song (2021).

through a continuous multi-scale noise schedule, where Gaussian noise is added at various continuous noise levels $\{\sigma_t\}_{t=1}^{T}$, producing a range of progressively noisier distributions $p(\mathbf{x}|\sigma_t)$ as σ_t increases. Each noise level σ_t represents a different degree of noise variance, where the sample is perturbed following:

$$\mathbf{x}_{\sigma_t} = \mathbf{x}_0 + \sigma_t \cdot \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, I). \tag{2.51}$$

The objective function in score-based generative modelling is thus formulated as a denoising score-matching loss, where the model, parameterised by ψ, is trained by minimising the expected squared error between the predicted score and the true score across a range of noise scales (Song and Ermon, 2019):

$$\mathcal{L}_{SBGM}(\psi) = \mathbb{E}_{\sigma_t}\left[\lambda(\sigma_t), \mathbb{E}_{\mathbf{x}_{0,\epsilon}} \left\|\mathbf{s}_\psi(\mathbf{x}_{\sigma_t}, \sigma_t) + \nabla \mathbf{x} \log p(\mathbf{x}_{\sigma_t}|\mathbf{x}_0)\right\|_2^2\right], \tag{2.52}$$

where $\lambda(\sigma_t)$ is a weighting function that can be chosen to balance the contributions from different noise levels σ_t (Song and Ermon, 2019).

For a Gaussian distribution $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, {\sigma_t}^2\mathbf{I})$, with mean $\boldsymbol{\mu}$ and variance ${\sigma_t}^2\mathbf{I}$, the score function is given by:

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}) = -\frac{\mathbf{x} - \boldsymbol{\mu}}{\sigma_t^2}. \tag{2.53}$$

In the case of Equation 2.51, $\mathbf{x}_{\sigma_t}$ is centered around the original data $\mathbf{x}_0$ with variance σ_t^2, and the score function is given by:

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}_{\sigma_t}|\mathbf{x}_0) = -\frac{\mathbf{x}_{\sigma_t} - \mathbf{x}_0}{\sigma_t^2} = -\frac{\sigma_t \cdot \boldsymbol{\epsilon}}{\sigma_t^2} = -\frac{\boldsymbol{\epsilon}}{\sigma_t}, \tag{2.54}$$

yielding the revised and tractable loss function:

$$\mathcal{L}_{SBGM}(\psi) = \mathbb{E}_{\sigma_t}\left[\lambda(\sigma_t), \mathbb{E}_{\mathbf{x}_{0,\boldsymbol{\epsilon}}}\left\|\mathbf{s}_\psi(\mathbf{x}_{\sigma_t}, \sigma_t) - \frac{\boldsymbol{\epsilon}}{\sigma_t}\right\|_2^2\right]. \tag{2.55}$$

This approach of learning the score function across multiple noise scales provides flexibility and stability, allowing the model to produce high-quality samples by leveraging the trained scores to iteratively denoise through Langevin dynamics (Equation 2.50), without fixed timesteps (Song and Ermon, 2019). However, achieving stability and convergence in score-based generative modelling requires careful tuning of the noise schedule and accurate approximation of the score function. If noise levels are too low at certain scales, the model may not sufficiently explore the data space, limiting the model's exposure to diverse representations of the data distribution (Song and Ermon, 2019). Conversely, overly high noise levels can obscure data structure by excessively corrupting the samples, making it challenging for the model to identify meaningful patterns within the data (Song and Ermon, 2019).

Denoising Diffusion Probabilistic Models

In another approach, Sohl-Dickstein et al. (2015) proposed the framework of *Denoising Diffusion Probabilistic Models* (DDPMs), which was further refined by Ho et al. (2020) to improve sample quality and training efficiency. Unlike score-based generative modelling, which models the underlying data distribution by learning the score function, DDPMs parameterise the reverse-time dynamics of the forward SDE by

learning the transition probabilities of intermediate noisy latent variables. DDPMs formalises the following form of an SDE:

$$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\mathbf{w}_t, \tag{2.56}$$

where $-\frac{1}{2}\beta(t)\mathbf{x}_t$ serves as a drift term that gradually decays the data signal over time, and $\sqrt{\beta(t)}\,\mathrm{d}\mathbf{w}_t$ is the diffusion term that introduces Gaussian noise with instantaneous variance $\beta(t)\,\mathrm{d}t$. This SDE is discretised to approximate the continuous process as a stepwise diffusion process parameterised by an also discretised noise schedule $\{\beta_t\}_{t=1}^{T}$, using Euler-Maruyama as follows:

$$\mathbf{x}_t \approx \mathbf{x}_{t-1} + \left(-\frac{1}{2}\beta_t\mathbf{x}_{t-1}\right)\Delta t + \sqrt{\beta_t \Delta t}\,\boldsymbol{\epsilon} \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}), \tag{2.57}$$

taking $\Delta t = 1$:

$$\mathbf{x}_t \approx \left(1 - \frac{1}{2}\beta_t\right)\mathbf{x}_{t-1} + \sqrt{\beta_t}\,\boldsymbol{\epsilon} \tag{2.58}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$ is Gaussian noise, and $\mathbf{x}_t$ is the intermediate noisy latent variable of the diffusion process at timestep t. Using the approximation $\sqrt{1-\beta_t} \approx 1 - \frac{1}{2}\beta_t$ for small β_t under a first order Taylor series expansion, the update step in DDPMs simplifies to:

$$\mathbf{x}_t = \sqrt{1-\beta_t}\,\mathbf{x}_{t-1} + \sqrt{\beta_t}\,\boldsymbol{\epsilon} \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}) \tag{2.59}$$

The scaling factor $\sqrt{1-\beta_t}$ approximates the continuous drift term in the forward SDE, and ensures that, for a monotonically increasing β_t each timestep gradually corrupts the data signal, pushing $\mathbf{x}_t$ towards a standard Gaussian distribution as $t \to T$ (Figure 2.10).

Cumulative Variance Alignment with the Continuous Process The choice of $\sqrt{1-\beta_t}$ is also essential for aligning the cumulative variance (noise) with that of the continuous diffusion process. At any small time interval $\mathrm{d}t$, the variance introduced by the noise term $\sqrt{\beta(t)}\,\mathrm{d}\mathbf{w}_t$ is approximately $\beta(t)\,\mathrm{d}t$, as per Brownian motion

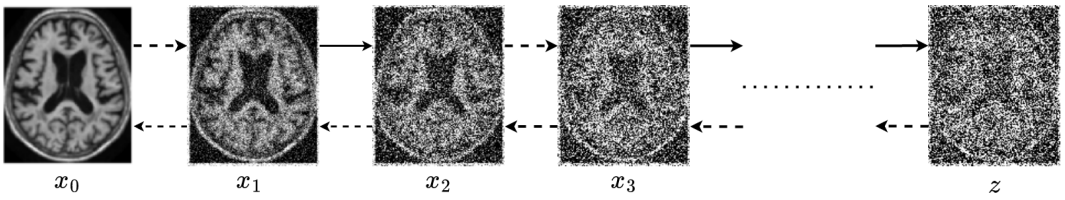


Figure 2.10: Illustration of the forward (solid right arrows) and reverse processes (dashed left arrows) in Denoising Diffusion Probabilistic Models (DDPMs). The forward process gradually adds Gaussian noise to the data, transforming it into a standard Gaussian distribution, here represented by z. The reverse process, parameterised by a neural network, iteratively denoises the noisy data back to the original data distribution. Extracted directly from Kebaili et al. (2023) — notation in the figure differs slightly from that used in the text.

properties. Thus, for the continuous SDE in Equation 2.56, the cumulative variance over time is defined by the integral of the noise variance $\beta(t)$ over all timesteps (Oksendal, 2013):

$$\mathrm{Var}(\mathbf{x}_T) = \int_0^T \beta(t)\,\mathrm{d}t, \tag{2.60}$$

In the discrete form, the variance of each timestep sample can be described following Equation 2.59 as:

$$\mathrm{Var}(\mathbf{x}_t) = \mathrm{Var}(\sqrt{1-\beta_t}\,\mathbf{x}_{t-1} + \sqrt{\beta_t}\,\boldsymbol{\epsilon}_t) = (1-\beta_t)\,\mathrm{Var}(\mathbf{x}_{t-1}) + \beta_t. \tag{2.61}$$

To express $\mathrm{Var}(\mathbf{x}_t)$ in terms of $\mathrm{Var}(\mathbf{x}_0)$, we can expand Equation 2.61 recursively (i.e., express $\mathrm{Var}(\mathbf{x}_t)$ as a function of $\mathrm{Var}(\mathbf{x}_{t-1})$, $\mathrm{Var}(\mathbf{x}_{t-1})$ as a function of $\mathrm{Var}(\mathbf{x}_{t-2})$, and so on) to obtain the discrete cumulative variance:

$$\mathrm{Var}(\mathbf{x}_t) = (1-\beta_{t-1})(1-\beta_{t-2})\cdots(1-\beta_0)\,\mathrm{Var}(\mathbf{x}_0) + \sum_{s=0}^{t-1}\beta_s \prod_{j=s+1}^{t-1}(1-\beta_j), \tag{2.62}$$

which can be compactly written as:

$$\mathrm{Var}(\mathbf{x}_t) = \prod_{s=0}^{t-1}(1-\beta_s)\,\mathrm{Var}(\mathbf{x}_0) + \sum_{s=0}^{t-1}\beta_s \prod_{j=s+1}^{t-1}(1-\beta_j), \tag{2.63}$$

where the first term, $\prod_{s=0}^{t-1}(1-\beta_s)\operatorname{Var}(\mathbf{x}_0)$, represents the initial variance $\operatorname{Var}(\mathbf{x}_0)$, scaled by the cumulative product of $(1-\beta_s)$ factors, which decays the initial signal over time. The second term, $\sum_{s=0}^{t-1}\beta_s\prod_{j=s+1}^{t-1}(1-\beta_j)$, accumulates the noise contributions at each timestep. Each β_s term represents the variance added by noise at timestep $0 \leq s \leq t-1$, scaled by the product of subsequent $(1-\beta_j)$ terms that account for the residual effect of this noise up to timestep t by gradually reducing the influence of earlier noise contributions over time. As $t \to T$ for $0 < \beta_1, \beta_2, \ldots, \beta_T < 1$, $\prod_{s=0}^{T-1}(1-\beta_s) \to 0$, the effect of the initial variance vanishes:

$$\lim_{t\to T}\operatorname{Var}(\mathbf{x}_t) = \lim_{t\to T}\cancel{\left(\prod_{s=0}^{t-1}(1-\beta_s)\operatorname{Var}(\mathbf{x}_0)\right)} + \lim_{t\to T}\left(\sum_{s=0}^{t-1}\beta_s\prod_{j=s+1}^{t-1}(1-\beta_j)\right), \quad (2.64)$$

leaving only the cumulative noise contributions:

$$\lim_{t\to T}\operatorname{Var}(\mathbf{x}_t) = \lim_{t\to T}\left(\sum_{s=0}^{T-1}\beta_s\prod_{j=s+1}^{T-1}(1-\beta_j)\right) \approx 1. \quad (2.65)$$

This aligns with the continuous process, where the variance of $\mathbf{x}_T$ is defined purely by the cumulative inserted noise, approaching that of a standard Gaussian of variance equal to 1.

Probabilistic Framework of DDPMs In the context of Denoising Diffusion Probabilistic Models (DDPMs), we define the following probability terms:

- $q(\mathbf{x}_0)$: the distribution of the original, uncorrupted data;
- $p_\psi(\mathbf{x}_0)$: the modelled distribution of the original, uncorrupted data, approximated by a neural network parameterised by ψ;
- $p_\psi(\mathbf{x}_T) \approx q(\mathbf{x}_T)$: the prior distribution of $\mathbf{x}_T$ after applying the forward diffusion process through T steps, commonly chosen to approximate a standard Gaussian distribution;
- $q(\mathbf{x}_t|\mathbf{x}_{t-1})$: the conditional distribution of latents $\mathbf{x}_t$ given $\mathbf{x}_{t-1}$ in the forward process;

- $q(\mathbf{x}_{0:T}|\mathbf{x}_0)$: the joint distribution over all steps from $\mathbf{x}_0$ to $\mathbf{x}_T$ in the forward process, representing the sequential degradation;
- $p_\psi(\mathbf{x}_{t-1}|\mathbf{x}_t)$: the conditional distribution of latents $\mathbf{x}_{t-1}$ given $\mathbf{x}_t$ in the reverse process, learned via a neural network parameterised by ψ;
- $p_\psi(\mathbf{x}_{T:0}|\mathbf{x}_t)$: the joint distribution over all steps from $\mathbf{x}_T$ to $\mathbf{x}_0$ in the reverse process, representing the sequential denoising;
- $\boldsymbol{\mu}_\psi(\mathbf{x}_t, t)$: the mean of the Gaussian distributions at each timestep t of the reverse process, approximated by a neural network parameterised by ψ;
- σ_t^2: the variance of the Gaussian distributions at each timestep t of the reverse process, which can be a fixed or learned parameter (often fixed).

In the probabilistic formulation of DDPMs (Sohl-Dickstein et al., 2015; Ho et al., 2020), the forward process is viewed as a Markov chain that transforms the data distribution $q(\mathbf{x}_0)$ into a standard Gaussian distribution $q(\mathbf{x}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ through the following series of transitions:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \tag{2.66}$$

which can be understood as a single intermediate degradation step where $\mathbf{x}_t$ is sampled from a Gaussian distribution with mean $\mathbf{x}_{t-1}$ scaled down by $\sqrt{1-\beta_t}$, and variance $\beta_t\mathbf{I}$ (equivalent to Equation 2.59). The complete forward process that defines the sequence of noisy data distributions from $\mathbf{x}_0$ to $\mathbf{x}_T$ can then be expressed as the joint distribution over all intermediate steps, representing the full Markov process of data degradation, formalised as:

$$\begin{aligned} q(\mathbf{x}_{0:T}|\mathbf{x}_0) &= q(\mathbf{x}_0)q(\mathbf{x}_1|\mathbf{x}_0)q(\mathbf{x}_2|\mathbf{x}_1)\cdots q(\mathbf{x}_T|\mathbf{x}_{T-1}) \\ &= \cancel{q(\mathbf{x}_0)} \prod_{t=1}^{T-1} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \\ &= \prod_{t=1}^{T-1} \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \end{aligned} \tag{2.67}$$

noting that the probability distribution of the data is equal to 1.

DDPMs aim to learn the reverse process that transitions Gaussian noise back to the data distribution $p(\mathbf{x}_0)$ (Sohl-Dickstein et al., 2015). This reverse process is also modelled as a Markov chain, wherein each intermediate step gradually removes noise from $\mathbf{x}_T$ in the direction of the original data distribution. This is formalised as the conditional probability of $\mathbf{x}_{t-1}$ given $\mathbf{x}_t$:

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t) := q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0), \tag{2.68}$$

where the right-hand term is the true posterior, conditioned on the clean data point $\mathbf{x}_0$, representing the "ideal" path from $\mathbf{x}_t$ back to $\mathbf{x}_{t-1}$. However, as $\mathbf{x}_0$ is not available during sampling, the conditional probability is reformulated with the unknown $\mathbf{x}_0$ replaced by a neural network estimate $\mathbf{x}_\psi(\mathbf{x}_t, t)$, as a function of its parameters ψ and the current data state $\mathbf{x}_t$, such that:

$$p_\psi(\mathbf{x}_{t-1}|\mathbf{x}_t) \approx q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_\psi(\mathbf{x}_t, t)). \tag{2.69}$$

For Gaussian processes with small β_t values, where the diffusion process is approximately continuous, it holds that the functional forms of $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ and $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ are identical (Sohl-Dickstein et al., 2015). This is because small noise increments produce nearly linear transitions at each step, making it feasible to approximate both forward and reverse processes with Gaussian distributions in closed, tractable form. Thus, Equation 2.69 is posed similarly to Equation 2.66 as:

$$p_\psi(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\psi(\mathbf{x}_t, t), \sigma_t^2 \mathbf{I}), \tag{2.70}$$

where $\boldsymbol{\mu}_\psi(\mathbf{x}_t, t)$ and σ_t^2 are the mean and variance, respectively, of each timestep of the reverse diffusion process, as modelled by the neural network. Similar to the forward process (Equation 2.66), the mean $\boldsymbol{\mu}_\psi(\mathbf{x}_t, t)$ varies at each timestep, although here it progressively guides $\mathbf{x}_t$ closer to the original data distribution in an iterative *denoising* process. The neural network is trained to minimise a loss function that aligns it with the true posterior distribution such that, at the end of the reverse chain, $\boldsymbol{\mu}_\psi(\mathbf{x}_1, 1)$ resembles the clean data state. On the other hand, the variance σ_t^2

introduces controlled randomness at each step, providing stability and variability, similar to score-based models, preventing deterministic collapse. This formulation means that, in order to sample from the original distribution, the network only needs to learn the mean and covariance of this diffusion kernel (Sohl-Dickstein et al., 2015; Ho et al., 2020). Although the variance σ_t^2 can be a learned parameter, it is often fixed as a hyperparameter (Ho et al., 2020).

Akin to the forward chain, the complete reverse process that transitions $\mathbf{x}_T$ back to $\mathbf{x}_0$ can be expressed as the joint distribution over all reverse steps to achieve the modelled distribution of the data:

$$\begin{aligned} p_\psi(\mathbf{x}_{T:0}|\mathbf{x}_T) &= p_\psi(\mathbf{x}_T)p_\psi(\mathbf{x}_{T-1}|\mathbf{x}_T)p_\psi(\mathbf{x}_{T-2}|\mathbf{x}_{T-1})\cdots p_\psi(\mathbf{x}_0|\mathbf{x}_1) \\ &= p(\mathbf{x}_T)\prod_{t=1}^{T} p_\psi(\mathbf{x}_{t-1}|\mathbf{x}_t) \\ &= p(\mathbf{x}_T)\prod_{t=1}^{T} \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\psi(\mathbf{x}_t, t), \sigma_t^2\mathbf{I}) \\ &= p_\psi(\mathbf{x}_0) \approx q(\mathbf{x}_0), \end{aligned} \tag{2.71}$$

where $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; 0, \mathbf{I})$ represents the initial standard Gaussian distribution of $\mathbf{x}_T$. Note that the formulation of Equation 2.71 is the equivalent to $p_\psi(\mathbf{x}_0)$, which in turn is how the real data distribution $q(\mathbf{x}_0)$ is modelled.

Forward Sampling Reparameterisation A common manipulation in DDPMs is to express $\mathbf{x}_t$ from Equation 2.59 recursively as a function of $\mathbf{x}_0$ and a normal noise term $\boldsymbol{\epsilon}$ (Ho et al., 2020). Let $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ be the cumulative product of α_t up to timestep t. Then, we can write $\mathbf{x}_t$ as:

$$
\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon} \\
&= \sqrt{\alpha_t}\left(\sqrt{\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_{t-1}}\boldsymbol{\epsilon}\right) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon} \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\boldsymbol{\epsilon} \\
&= \sqrt{\alpha_t\alpha_{t-1}\alpha_{t-2}}\mathbf{x}_{t-3} + \sqrt{1-\alpha_t\alpha_{t-1}\alpha_{t-2}}\boldsymbol{\epsilon} \\
&\vdots \\
&= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon},
\end{aligned}
\tag{2.72}
$$

meaning that, with this formulation, the forward Markov chain can be sampled at any arbitrary timestep t for a given initial state $\mathbf{x}_0$ and the noise $\boldsymbol{\epsilon}$ without the need to sample all intermediate steps. This is then formalised as the distribution of $\mathbf{x}_t$ given $\mathbf{x}_0$ as:

$$
q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right). \tag{2.73}
$$

Sampling in the Reverse Process Starting from the forward process in Equation 2.59, we rearrange to isolate the previous data state $\mathbf{x}_{t-1}$ as a function of the current state $\mathbf{x}_t$ and the noise $\boldsymbol{\epsilon}$ (Weng, 2021):

$$
\mathbf{x}_{t-1} = \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t - \sqrt{\beta_t}\,\boldsymbol{\epsilon}\right), \tag{2.74}
$$

Since the network predicts the noise $\boldsymbol{\epsilon}_\psi(\mathbf{x}_t, t) \approx \boldsymbol{\epsilon}$ (Ho et al., 2020), we can substitute $\boldsymbol{\epsilon}_\psi(\mathbf{x}_t, t)$ into Equation 2.74 to obtain:

$$
\mathbf{x}_{t-1} \approx \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t - \sqrt{\beta_t}\,\boldsymbol{\epsilon}_\psi(\mathbf{x}_t, t)\right), \tag{2.75}
$$

which represents the deterministic mapping of the Markov process backward in time from $\mathbf{x}_t$ to $\mathbf{x}_{t-1}$ based on the predicted noise $\boldsymbol{\epsilon}_\psi(\mathbf{x}_t, t)$ from the network.

Alternatively, we can rearrange Equation 2.72 to isolate $\mathbf{x}_0$ directly in terms of $\mathbf{x}_t$ and the noise:

$$\mathbf{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t}\, \boldsymbol{\epsilon} \right), \tag{2.76}$$

and the network approximation is given by:

$$\mathbf{x}_0 \approx \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t}\, \boldsymbol{\epsilon}_\psi(\mathbf{x}_t, t) \right), \tag{2.77}$$

although this formulation is not directly used for sampling due to instabilities, which arise from taking a large step that violates the assumptions of Equation 2.69.

Parameterising the Mean The DDPM neural network is parameterised to predict the noise $\boldsymbol{\epsilon}_\psi(\mathbf{x}_t, t) \approx \boldsymbol{\epsilon}$ added at each step of the forward process, rather than predicting the mean of the reverse process (Equation 2.70) directly, as Ho et al. (2020) empirically demonstrated that this approach simplifies the training objective and enhances stability. Using this noise prediction, the mean of the reverse process can be derived as follows.

We can use Bayes' theorem to obtain the reverse step $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ from the forward step defined in Equation 2.66 (Weng, 2021).

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0) q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)}, \tag{2.78}$$

and since $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)$, representing the forward process step is not dependent on $\mathbf{x}_0$, we can simplify Equation 2.78 to:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = q(\mathbf{x}_t|\mathbf{x}_{t-1}) \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)}. \tag{2.79}$$

This formulation can be expanded by their Gaussian definitions in Equations 2.66 and 2.73, yielding:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \propto \exp \begin{pmatrix} -\frac{1}{2\beta_t} \|\mathbf{x}_t - \sqrt{\alpha_t}\, \mathbf{x}_{t-1}\|_2^2 \\ -\frac{1}{2(1 - \bar{\alpha}_{t-1})} \|\mathbf{x}_{t-1} - \sqrt{\bar{\alpha}_{t-1}}\, \mathbf{x}_0\|_2^2 \\ +\frac{1}{2(1 - \bar{\alpha}_t)} \|\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\, \mathbf{x}_0\|_2^2 \end{pmatrix}, \tag{2.80}$$

where the exponential properties $\exp(a)\exp(b) = \exp(a+b)$ and $\frac{\exp(a)}{\exp(b)} = \exp(a-b)$ are used. Expanding the $||\cdot||_2^2$ terms:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \propto \exp\left(\begin{array}{l} -\frac{1}{2\beta_t}\left(\mathbf{x}_t^\top \mathbf{x}_t - 2\sqrt{\alpha_t}\,\mathbf{x}_t^\top \mathbf{x}_{t-1} + \alpha_t\,\mathbf{x}_{t-1}^\top \mathbf{x}_{t-1}\right) \\ -\frac{1}{2(1-\bar{\alpha}_{t-1})}\left(\mathbf{x}_{t-1}^\top \mathbf{x}_{t-1} - 2\sqrt{\bar{\alpha}_{t-1}}\,\mathbf{x}_{t-1}^\top \mathbf{x}_0 + \bar{\alpha}_{t-1}\,\mathbf{x}_0^\top \mathbf{x}_0\right) \\ +\frac{1}{2(1-\bar{\alpha}_t)}\left(\mathbf{x}_t^\top \mathbf{x}_t - 2\sqrt{\bar{\alpha}_t}\,\mathbf{x}_t^\top \mathbf{x}_0 + \bar{\alpha}_t\,\mathbf{x}_0^\top \mathbf{x}_0\right) \end{array}\right). \tag{2.81}$$

We then isolate any terms that do not depend on $\mathbf{x}_{t-1}$ and represent them as a constant C:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \propto \exp\left(\begin{array}{l} -\frac{1}{2}\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right)\mathbf{x}_{t-1}^\top \mathbf{x}_{t-1} \\ +\left(\frac{\sqrt{\alpha_t}}{\beta_t}\,\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\,\mathbf{x}_0\right)^\top \mathbf{x}_{t-1} \\ -\frac{1}{2\beta_t}\mathbf{x}_t^\top \mathbf{x}_t \\ -\frac{1}{2(1-\bar{\alpha}_{t-1})}\,\bar{\alpha}_{t-1}\,\mathbf{x}_0^\top \mathbf{x}_0 \\ +\frac{1}{2(1-\bar{\alpha}_t)}\left(\mathbf{x}_t^\top \mathbf{x}_t - 2\sqrt{\bar{\alpha}_t}\,\mathbf{x}_t^\top \mathbf{x}_0 + \bar{\alpha}_t\,\mathbf{x}_0^\top \mathbf{x}_0\right) \end{array}\right), \tag{2.82}$$

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \propto \exp\left(\begin{array}{l} -\frac{1}{2}\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right)\mathbf{x}_{t-1}^\top \mathbf{x}_{t-1} \\ +\left(\frac{\sqrt{\alpha_t}}{\beta_t}\,\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\,\mathbf{x}_0\right)^\top \mathbf{x}_{t-1} \\ +C \end{array}\right), \tag{2.83}$$

and by taking:

$$\boldsymbol{\mu}_{t-1} = \sigma_{t-1}^2\left(\frac{\sqrt{\alpha_t}}{\beta_t}\,\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\,\mathbf{x}_0\right) \quad \text{and} \quad \sigma_{t-1}^2 = \frac{1}{\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}} = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t, \tag{2.84}$$

we can express the reverse step posterior as:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \propto \exp\left(-\frac{1}{2\sigma_{t-1}^2}\left(\mathbf{x}_{t-1} - \boldsymbol{\mu}_{t-1}\right)^\top\left(\mathbf{x}_{t-1} - \boldsymbol{\mu}_{t-1}\right)\right), \tag{2.85}$$

which is the canonical form of a Gaussian distribution. Therefore, the expression in Equation 2.84 represents the mean $\boldsymbol{\mu}_{t-1}$ and variance σ_{t-1}^2 of the reverse step as a

function of the current state $\mathbf{x}_t$ and the initial state $\mathbf{x}_0$. A more compact form of the mean is given by expanding the σ_{t-1}^2 term:

$$\begin{aligned}\boldsymbol{\mu}_\psi(\mathbf{x}_t, t) &= \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t\left(\frac{\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0\right) \\ &= \left(\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\right)\mathbf{x}_t + \left(\frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t}\right)\mathbf{x}_0.\end{aligned} \tag{2.86}$$

where $\beta_t = 1 - \alpha_t$ was used. However, since the initial state $\mathbf{x}_0$ is unknown during sampling, we can use Equation 2.76 to rewrite the mean:

$$\begin{aligned}\boldsymbol{\mu}_\psi(\mathbf{x}_t, t) &= \left(\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\right)\mathbf{x}_t + \left(\frac{\sqrt{\bar{\alpha}_{t-1}}\,(1-\alpha_t)}{1-\bar{\alpha}_t}\right)\frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}\right) \\ &= \left(\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} + \frac{\sqrt{\bar{\alpha}_{t-1}}\,(1-\alpha_t)}{1-\bar{\alpha}_t}\frac{1}{\sqrt{\bar{\alpha}_t}}\right)\mathbf{x}_t - \frac{\sqrt{\bar{\alpha}_{t-1}}\,(1-\alpha_t)\sqrt{1-\bar{\alpha}_t}}{1-\bar{\alpha}_t}\boldsymbol{\epsilon},\end{aligned} \tag{2.87}$$

which algebraically simplifies to (Weng, 2021):

$$\boldsymbol{\mu}_\psi(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}\right), \tag{2.88}$$

and approximated by the network with $\boldsymbol{\epsilon}_\psi(\mathbf{x}_t, t) \approx \boldsymbol{\epsilon}$ as:

$$\boldsymbol{\mu}_\psi(\mathbf{x}_t, t) \approx \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\psi(\mathbf{x}_t, t)\right). \tag{2.89}$$

This expression represents the mean of the reverse process as a function of the current state $\mathbf{x}_t$ and the predicted noise component $\boldsymbol{\epsilon}_\psi(\mathbf{x}_t, t)$. So, in summary, the following holds:

$$\mathbf{x}_{t-1} \sim \mathcal{N}\left(\mathbf{x}_{t-1}; \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\psi(\mathbf{x}_t, t)\right), \sigma_t^2\mathbf{I}\right), \tag{2.90}$$

which is sampled via:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\psi(\mathbf{x}_t, t)\right) + \sigma_t\zeta, \tag{2.91}$$

where σ_t is the standard deviation of the reverse process at timestep t defined by Equation 2.84, and $\zeta \sim \mathcal{N}(0, \mathbf{I})$ is the stochastic term. This formulation aligns

with the reverse Markov process defined by Ho et al. (2020), enabling the model to reconstruct $\mathbf{x}_0$ by iteratively denoising from $\mathbf{x}_T \sim \mathcal{N}(\mathbf{x}_T; 0, \mathbf{I})$ down to $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ using the noise predicted from the trained network at each timestep (Ho et al., 2020). Unlike the formulation of 2.75, this formulation is a more accurate representation of $\mathbf{x}_{t-1}$ as a function of $\mathbf{x}_t$ because it accounts for the cumulative noise added at each step of the forward process.

Note that the form of Equation 2.91 closely resembles that of Langevin dynamics, as shown in Equation 2.50. Although DDPMs only indirectly model the data distribution, the concept of denoising a noisy sample towards regions of high likelihood, as in score-based models, also applies to DDPMs. The distributions of the transitional noisy latents $\mathbf{x}_{t-1}$ are conceptually aligned with the progressive transformation of a simpler distribution towards the data distribution, as depicted in Figure 2.9.

Variational Inference and Objective Function As with any generative model, training a DDPM can be posed as a likelihood maximisation of the data distribution $q(\mathbf{x}_0)$, modelled as $p_\psi(\mathbf{x}_0) \approx q(\mathbf{x}_0)$. This can be formally described as the minimisation of the cross-entropy loss of the modelled distribution as:

$$\mathcal{L}_{CE} = -\mathbb{E}_{q(\mathbf{x}_0)} \log p_\psi(\mathbf{x}_0), \tag{2.92}$$

which reads as the expectation of the negative log-likelihood of the modelled distribution over the true data distribution. As previously shown, the modelled distribution is defined by the joint distribution of the reverse process, $p_\psi(\mathbf{x}_{T:0}|\mathbf{x}_T)$, which is parameterised by the neural network. The cross-entropy loss can be expanded as:

$$\mathcal{L}_{CE} = -\mathbb{E}_{q(\mathbf{x}_0)} \log \left(\int p_\psi(\mathbf{x}_{T:0})\, d\mathbf{x}_{T:1} \right). \tag{2.93}$$

The integral term in Equation 2.93 represents the marginalisation of the joint distribution over all intermediate steps from $\mathbf{x}_T$ to $\mathbf{x}_0$, essentially describing all possible paths from the final Gaussian noise state back to the original data. Integrating over this large, interconnected sequence of possibilities across high-dimensional space is,

however, intractable. To address this, we use variational inference to approximate these steps using importance sampling, where we draw samples from a simpler distribution that approximates the true posterior and reweight these samples to approximate the integral.

We first introduce an auxiliary, or variational, distribution $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$ (Nguyen, 2023), which represents the posterior of the degraded data (Ho et al., 2020). This posterior is known (see Equation 2.67) and, therefore, easy to sample from. By multiplying and dividing the integral term in Equation 2.93 by the auxiliary distribution $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$, we can reformulate the cross-entropy loss as a function of the ratio of the joint distribution to the auxiliary distribution:

$$\mathcal{L}_{CE} = -\mathbb{E}_{q(\mathbf{x}_0)} \log \left(\int q(\mathbf{x}_{1:T}|\mathbf{x}_0) \frac{p_\psi(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \, d\mathbf{x}_{1:T} \right). \tag{2.94}$$

We note here that the integral of a function weighted by a probability distribution can be expressed, by definition, as the sample-based expectation of the integrand. For an arbitrary function $f(\mathbf{x}_{1:T})$, the following holds:

$$\mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}[f(\mathbf{x}_{1:T})] := \int q(\mathbf{x}_{1:T}|\mathbf{x}_0) f(\mathbf{x}_{1:T}) \, d\mathbf{x}_{1:T}, \tag{2.95}$$

and by taking $f(\mathbf{x}_{1:T}) = \frac{p_\psi(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}$, Equation 2.94 becomes:

$$\mathcal{L}_{CE} = -\mathbb{E}_{q(\mathbf{x}_0)} \log \left(\mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \frac{p_\psi(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right). \tag{2.96}$$

This formulation is beneficial because it allows us to rewrite the intractable integral in Equation 2.93 as an expectation, which can be approximated using sampling methods. That is, instead of computing the full integral, we can now draw samples $\mathbf{x}_{1:T}^{(i)}$ from $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$ and compute the expectation as an average over these samples:

$$\mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\frac{p_\psi(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] \approx \frac{1}{N} \sum_{i=1}^{N} \frac{p_\psi(\mathbf{x}_{0:T}^{(i)})}{q(\mathbf{x}_{1:T}^{(i)}|\mathbf{x}_0)}, \tag{2.97}$$

which represents an estimate of the total probability mass of the modelled distribu-

tion. The ratio $\frac{p_\psi(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}$ is known as the *importance weight* and serves to adjust the contributions of samples drawn from $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$ so that they can approximate the distribution $p_\psi(\mathbf{x}_{0:T})$, even though the samples themselves come from q rather than p_ψ.

Jensen's inequality (Jensen, 1906) states that for a concave function f, the function applied to an expectation is greater than or equal to the expectation of the function applied to each point individually. In our case, since log is concave, $\log \mathbb{E}[f(x)] \geq \mathbb{E}[\log f(x)]$ holds and the log of the expectation is formulated as:

$$\log \left(\mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \frac{p_\psi(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right) \geq \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \frac{p_\psi(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right], \tag{2.98}$$

which we apply to Equation 2.96 to obtain the evidence lower bound (ELBO) of the cross-entropy loss:

$$\begin{aligned} \mathcal{L}_{CE} &\leq -\mathbb{E}_{q(\mathbf{x}_0)} \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \frac{p_\psi(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] \\ &= -\mathbb{E}_{q(\mathbf{x}_{0:T})} \left[\log \frac{p_\psi(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] \\ &= \mathbb{E}_{q(\mathbf{x}_{0:T})} \left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\psi(\mathbf{x}_{0:T})} \right] = \mathcal{L}_{ELBO}. \end{aligned} \tag{2.99}$$

Like VAEs, maximisation of the likelihood of the observed data in DDPMs is made tractable via an approximate posterior distribution and the ELBO (Kingma and Welling, 2013; Ho et al., 2020). In fact, DDPMs are not too dissimilar to VAEs (Ho et al., 2020). Mathematically, both models use a sequence of latent variables to represent uncertainty and to transform an initial noise distribution into a data distribution over a series of probabilistic steps. However, while latent variables in VAEs are drawn from a single-step variational posterior, latent variables in DDPMs are represented by a sequence of intermediate noisy states, where each state $\mathbf{x}_t$ gradually transforms the observed data $\mathbf{x}_0$ into a posterior distribution. Additionally, VAE latent variables are typically low-dimensional, encoding high-level features, whereas in DDPMs, the latent variables maintain the same dimensionality as the data without a bottleneck (Kingma and Welling, 2013; Ho et al., 2020).

In order to be analytically computable, the ELBO in Equation 2.99 can be

expanded with Equations 2.67 and 2.71 to be written in terms of the joint distribution of the reverse process:

$$\mathcal{L}_{ELBO} = \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{x}_T)\prod_{t=1}^{T} p_\psi(\mathbf{x}_{t-1}|\mathbf{x}_t)}\right], \tag{2.100}$$

which, following Weng (2021) and Appendix B of Sohl-Dickstein et al. (2015), can be decomposed as:

$$\mathcal{L}_{ELBO} = \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p_\psi(\mathbf{x}_T)} + \sum_{t=2}^{T} \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\psi(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \log p_\psi(\mathbf{x}_0|\mathbf{x}_1)\right], \tag{2.101}$$

where the first term represents is the KL divergence between the model-imposed prior $p_\psi(\mathbf{x}_T)$ and the true terminal distribution of the forward diffusion process $q(\mathbf{x}_T|\mathbf{x}_0)$, quantifying how well the model's prior over the final noisy state aligned with the actual distribution of noisy data at timestep T. It can be treated as constant if $p(\mathbf{x}_T)$ is fixed; the summation term is the sum of KL divergences between the true reverse posteriors $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ and the learned reverse process $p_\psi(\mathbf{x}_{t-1}|\mathbf{x}_t)$ across timesteps, representing how well the model can approximate the reverse dynamics of the forward process; and the third term is the negative log-likelihood of recovering $\mathbf{x}_0$ from the last denoised state, measuring the model's ability to reconstruct data accurately in the final reverse step.

The loss function in DDPM maximises an approximate form of the ELBO by minimising the middle summation term only, driving the focus of the loss to the reconstruction of the noisy latent states and ensuring accurate denoising:

$$\mathcal{L}_{DDPM} = \sum_{t=1}^{T} \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\mathrm{KL}\left(q(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{x}_0) \,||\, p_\psi(\mathbf{x}_t|\mathbf{x}_{t+1})\right)\right]. \tag{2.102}$$

For two normal Gaussian distributions, the KL divergence can be computed in closed form by inserting their respective probability density functions in Equation 2.13. Considering a fixed isotropic variance $\sigma^2\mathbf{I}$ for both distributions, in multivariate form, the KL divergence is given by:

$$\mathrm{KL}\left(\mathcal{N}_1 \,||\, \mathcal{N}_2\right) = \frac{1}{2\sigma^2}\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2^2, \tag{2.103}$$

where $\mathcal{N}_1$ and $\mathcal{N}_2$ are two Gaussian distributions parameterised by means $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, respectively. Since both $q(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{x}_0)$ and $p_\psi(\mathbf{x}_t|\mathbf{x}_{t+1})$ are normally distributed (Equations 2.85 and 2.70), Equation 2.102 becomes:

$$\mathcal{L}_{DDPM} = \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{1}{2\sigma_t^2}\left\|\boldsymbol{\mu}_q(\mathbf{x}_t|\mathbf{x}_{t+1},\mathbf{x}_0) - \boldsymbol{\mu}_\psi(\mathbf{x}_t|\mathbf{x}_{t+1})\right\|_2^2\right], \tag{2.104}$$

where we assume a fixed variance σ_t per timestep, and the constants that do not influence gradients are removed. We then use Equation 2.88 to substitute the true mean $\boldsymbol{\mu}_q(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{x}_0)$ and predicted mean $\boldsymbol{\mu}_\psi(\mathbf{x}_t|\mathbf{x}_{t+1})$ and obtain the DDPM loss as a function of the true and predicted noise:

$$\mathcal{L}_{DDPM} = \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{1}{2\sigma_t^2}\left\|\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{(1-\alpha_t)}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}\right) - \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{(1-\alpha_t)}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\psi(\mathbf{x}_t,t)\right)\right\|_2^2\right], \tag{2.105}$$

which can be reduced to the following form:

$$\begin{aligned}
\mathcal{L}_{DDPM} &= \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{1}{2\sigma_t^2}\left\|\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{(1-\alpha_t)}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}\right) - \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{(1-\alpha_t)}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\psi(\mathbf{x}_t,t)\right)\right\|_2^2\right] \\
&= \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{1}{2\sigma_t^2}\left\|\cancel{\frac{1}{\sqrt{\alpha_t}}\mathbf{x}_t} - \frac{(1-\alpha_t)}{\sqrt{\alpha_t(1-\bar{\alpha}_t)}}\boldsymbol{\epsilon} - \cancel{\frac{1}{\sqrt{\alpha_t}}\mathbf{x}_t} + \frac{(1-\alpha_t)}{\sqrt{\alpha_t(1-\bar{\alpha}_t)}}\boldsymbol{\epsilon}_\psi(\mathbf{x}_t,t)\right\|_2^2\right] \\
&= \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{1}{2\sigma_t^2}\left\|\frac{1-\alpha_t}{\sqrt{\alpha_t(1-\bar{\alpha}_t)}}\left(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\psi(\mathbf{x}_t,t)\right)\right\|_2^2\right] \\
&= \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{(1-\alpha_t)^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\psi(\mathbf{x}_t,t)\right\|_2^2\right].
\end{aligned} \tag{2.106}$$

Without assuming fixed isotropic variance, σ_t^2 in the equation can be replaced by a learned covariance matrix $\|\boldsymbol{\Sigma}_\psi\|_2^2$. Further simplifying Equation 2.106 by removing the constant terms, the final form of the DDPM loss is:

$$\begin{aligned}
\mathcal{L}_{\text{DDPM}} &= \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\psi(\mathbf{x}_t,t)\right\|_2^2\right] \\
&= \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\psi(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t,t)\right\|_2^2\right],
\end{aligned} \tag{2.107}$$

which is simply computed as the mean squared error (MSE) between the true noise and the model-predicted noise across all levels of degradation. This formulation

aligns with the findings of Ho et al. (2020), which demonstrated that approximating the ELBO and assuming fixed isotropic covariance is sufficient for training denoising diffusion models. The network primarily focuses on predicting the added noise, provided that the noise degradation is calibrated such that each latent state $\mathbf{x}_t$ approximates the posterior distribution well. With this approximation, the network benefits from a simplified objective and improved stability to achieve efficient training and high sampling quality. Algorithms 4 and 5 summarise the training and sampling processes in DDPM, respectively.

Algorithm 4 Training Algorithm for DDPMs — adapted from Ho et al. (2020)

Input: Training data distribution $q(\mathbf{x}_0)$, noise schedule $\bar{\alpha}_t$
Output: Trained denoising model $\boldsymbol{\epsilon}_\psi$
Initialise model parameters ψ
repeat
  for number of training iterations **do**
    Step 1: Sample data and noise
    Sample data point $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
    Sample diffusion step $t \sim \text{Uniform}(\{1, \ldots, T\})$
    Sample noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    Step 2: Corrupt sample and compute loss on predicted noise

$$\mathcal{L} = \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\psi \left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t \right) \right\|_2^2 \tag{2.108}$$

    Step 3: Update model parameters

$$\psi \leftarrow \psi - \alpha \nabla_\psi \mathcal{L} \tag{2.109}$$

  end for
until convergence of the denoising model

Algorithm 5 Sampling Algorithm for DDPMs — adapted from Ho et al. (2020)

Input: Trained denoising model $\boldsymbol{\epsilon}_\psi$, noise schedule $\alpha_t, \bar{\alpha}_t, \sigma_t$
Output: Generated sample $\mathbf{x}_0$
Initialise: Sample noise $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ **do**
  Sample $\zeta \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else set $\zeta = 0$
  Perform denoising step:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\psi(\mathbf{x}_t, t) \right) + \sigma_t \zeta \tag{2.110}$$

end for
Return: $\mathbf{x}_0$ (final denoised sample)

Diffusion Models in Practice

In the realm of generative models, diffusion models distinguish themselves by employing a predefined posterior $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$ rather than learning it through the network. This predefined forward process allows the model to focus exclusively on capturing the reverse process, enhancing its flexibility and stability when modelling complex data distributions. This framework has demonstrated strong inductive biases for intricate data, offering improved resolution, diversity, and robustness in training compared to VAEs and GANs (Ho et al., 2020). By preserving the dimensionality of latent variables to match that of the data, diffusion models maintain a high fidelity to the original distribution, which proves especially beneficial in high-resolution image synthesis and other data-intensive tasks.

When comparing score-based models to DDPMs, each offers distinct advantages and trade-offs that arise from their differing generative approaches. Score-based models, which learn the gradient of the data distribution, possess flexibility in adapting to various noise types, making them particularly suited to dynamic or diverse datasets. This adaptability, however, comes at the cost of precise noise scheduling and often demands significant computational resources to ensure model stability (Song and Ermon, 2019).

In contrast, the sequential, stepwise approach of DDPMs with a clearly defined reverse process allow them to generate high-quality outputs with more stability and robustness (Ho et al., 2020). However, this reliance on a reverse Markov chain has been a wide topic of discussion in the field, as the extensive timesteps required in the reverse process, often reaching hundreds or even thousands, lead to impractical computational demands and slower sampling times compared to implicit generative models like GANs (Ho et al., 2020). To address these computational challenges, several methods have been proposed to optimise the efficiency of diffusion models. Distillation methods compress multiple diffusion steps into a single optimised model, effectively approximating the original diffusion reverse process in a single or limited number of steps (Meng et al., 2023; Sauer et al., 2025). Consistency models also decrease the computational time of DDPMs by approximating the reverse process

as a continuous-time trajectory using ODEs, allowing for sampling in fewer steps while maintaining stable and high-quality results (Song et al., 2023). Variational Diffusion Models (VDMs) take a variational approach to this problem, optimising the variational lower bound to to learn a more efficient reverse process and minimise the number of steps required for accurate sampling (Kingma et al., 2021). Each of these approaches contributes to balancing sample quality with computational load of diffusion models, making them increasingly practical and scalable for wide-spread adopting.

A notable advancement addressing the inefficiencies in DDPMs is the Denoising Diffusion Implicit Model (DDIM), which retains the procedural foundation of DDPMs but modifies the sampling process to accelerate generation (Song et al., 2020). While DDPMs employ a stochastic Markov chain, with each reverse diffusion timestep adding a small noise term to progressively sample from the data distribution, DDIMs adopt a deterministic, non-Markovian sampling trajectory. This reformulation allows for a controlled, predictable path in the reverse process, offering a significant reduction in the number of sampling steps needed. The DDIM reverse process is derived as a modification of Equation 2.91 and is represented by:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\, \boldsymbol{\epsilon}_\psi(\mathbf{x}_t, t) \right) + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\, \boldsymbol{\epsilon}_\psi(\mathbf{x}_t, t) + \sigma_t\, \zeta, \quad (2.111)$$

where the first term represents the predicted $\mathbf{x}_0$ scaled by a function of the cumulative noise, the second term is a directional vector pointing to $\mathbf{x}_t$ with some defined variability, where the term $\sigma_t = \eta \sqrt{\frac{1-\alpha_{t-1}}{1-\alpha_t}} \sqrt{1-\frac{\alpha_t}{\alpha_{t-1}}}$ controls the degree of stochasticity in the sampling path, and the third term adds stochasticity in the same way was DDPMs. At $\eta = 1$, Equation 2.111 reproduces DDPM sampling. However, by setting $\eta = 0$, the model enforces a fully deterministic process, ensuring that identical initial noise leads to the same generated data sample (Song et al., 2020). The non-Markovian deterministic nature of DDIMs at $\eta = 0$ achieves the same high-quality synthesis as DDPMs while streamlining the sampling process into a semi-continuous trajectory rather than a sequence of stochastic transitions, reducing sampling timesteps from around 1000 to as few as 10 or 100 (Song et al., 2020).

Crucially, sample diversity is preserved despite the deterministic trajectory, as it is entirely derived from the initial noise configuration, which allows for diverse outputs even when $\eta = 0$. DDIMs are advantageous for implementation within existing DDPM workflows, as the modifications impact only the sampling process, preserving the structure and compatibility with established DDPM architectures.

Additionally, while diffusion models eliminate the complexities of adversarial training, they introduce a new set of tuning challenges. The flexibility offered by various samplers, variance schedules, and diffusion timesteps creates a range of hyperparameters that substantially impact model performance (Song and Ermon, 2019; Ho et al., 2020). For instance, the number of timesteps plays a crucial role in detail resolution, with higher timesteps generally yielding finer image fidelity (Song et al., 2020). The choice of variance schedule, such as linear or cosine, affects the noise progression throughout the diffusion process, ultimately shaping the model's ability to denoise effectively and generate high-quality samples (Song et al., 2021b). Achieving optimal performance often requires extensive experimentation and fine-tuning, making the training process computationally demanding. Ultimately, while diffusion models excel in stability and quality, their effectiveness hinges on precise parameter selection, which is a nuanced balancing act that defines their success in generative modelling.

2.1.6 Transformers and Attention Mechanisms

Transformer models have revolutionised the field of deep learning since their introduction by Vaswani et al. in 2017 (Vaswani et al., 2017). These models were developed to address the limitations of recurrent neural networks (RNNs) (Rumelhart et al., 1986; Hochreiter and Schmidhuber, 1997), and convolutional neural networks (CNNs) (LeCun et al., 1998) in handling long-range dependencies of sequential data and to provide a heavily parallelisable framework to improve computational performances of models (Vaswani et al., 2017). Transformer models were primarily designed for natural language processing (NLP) tasks, where the input data is a one-dimensional sequence of tokens, such as words or characters. However, transformers and attention have since been adapted to various other domains, including

image and video processing (Dosovitskiy et al., 2020; Carion et al., 2020; Liu et al., 2024), speech recognition (Dong et al., 2018; Chan et al., 2016), and reinforcement learning (Parisotto and Salakhutdinov, 2018; de Choudhury et al., 2021; Kannala et al., 2018).

The architecture of transformers is based on an encoder-decoder structure. The encoder processes the input sequence, such as an English sentence, which is first mapped to a continuous representation through embeddings to generate hidden representations. The decoder utilises these hidden representations, along with embeddings of the output tokens, to produce the final output sequence in the form of token probabilities, such as the translated sentence in French. Both the encoder and decoder consist of multiple identical layers, each integrating a multi-head self-attention mechanism and a position-wise fully connected feed-forward network, with residual connections and layer normalisation applied at each layer (Figure 2.11).

Since its development, transformer models have massively evolved in the number of layers and parameters. For instance, the popular Generative Pre-trained Transformer (GPT) model that powers services like ChatGPT (Bahrini et al., 2023) and BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019a) has evolved since its launch from 110 million to estimated 1.76 trillion model parameters (Radford et al., 2021; Schreiner, 2023). But overall, the architecture remains consistent with the original design (Brown et al., 2020).

At the core of transformer models lies the multi-head attention mechanism. Attention mechanisms enable the model to assign varying importance to different parts of the input data, allowing it to capture relationships regardless of their distance within the sequence. This ability to dynamically weigh the significance of elements allows the model to understand long-range dependencies and interactions (Bahdanau et al., 2014), making it highly effective in tasks such as language translation, text summarisation, and conversational reasoning. Each self-attention layer employs multiple *attention heads* to capture diverse relationships between tokens in the sequence. Each head performs an independent attention operation, with their outputs concatenated and linearly transformed at the end of the process. The attention mechanism computes significance through *attention scores* by comparing each input token to

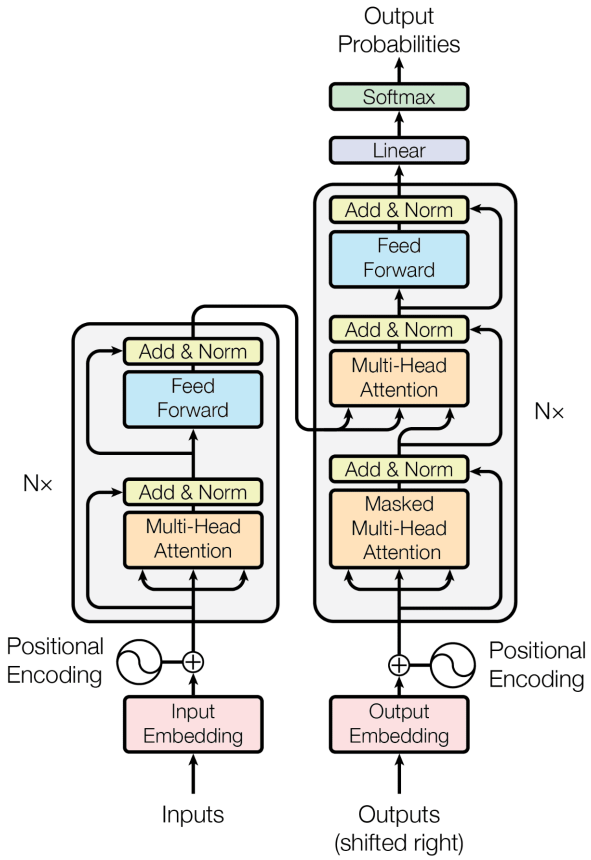


Figure 2.11: Transformer model architecture proposed byVaswani et al. (2017). The figure illustrates the transformer model architecture, comprising an encoder (left) and a decoder (right). The encoder consists of N identical layers, each containing a multi-head self-attention mechanism and a feed-forward network, both followed by residual connections and layer normalisation. Input embeddings are combined with positional encodings before being processed by the encoder. The decoder, similarly composed of N layers, includes an additional masked multi-head attention mechanism to prevent positions from attending to subsequent positions. The final output probabilities are produced through a linear transformation followed by a softmax function.

all others in the sequence, producing a weighted sum of the values based on these scores. This process is formalised as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.112}$$

Here, Q (query), K (key) and V (value) are learned vectors of dimension d_k derived from the outputs of the previous layer. Conceptually, queries represent the elements for which the model is seeking relevant information, keys represent the elements that are being compared against the queries, and values carry the actual

information or data associated with each key (Vaswani et al., 2017). The dot product QK^T is the computation of similarity scores between queries and keys, and therefore measures how much focus each query should place on the keys. These scores are transformed into probabilities through the softmax function, with the term $\sqrt{d_k}$ acting as a scaling factor that prevents excessively large values, ensuring stable gradients during training (Vaswani et al., 2017). The resulting attention weights are then applied to the values vector V, emphasising the most relevant tokens in the sequence (Vaswani et al., 2017; Bahdanau et al., 2014).

Positional encoding is also a crucial aspect of transformer models. These are added to input and output embeddings of the model to provide information about the position of each token in the sequence. This allows the model to distinguish between different positions and effectively capture the sequence's structure without relying on the order of the tokens (Vaswani et al., 2017). The powerful implication of this is that transformers can process all tokens in the sequence simultaneously, unlike RNNs which rely on sequential processing and the output of previous tokens. This independence of attention computations in transformers enables highly-scalable parallel processing, making them efficient and adaptable to modern hardware like GPUs (Vaswani et al., 2017).

In addition to being highly parallelisable, attention mechanisms capture global relationships across entire sequences, eliminating the need for convolution operations (Vaswani et al., 2017). This reduction in convolutional dependencies decreases computational complexity and improves the scalability of the model.

When queries, keys, and values are derived from the same encoder outputs, the attention mechanism is referred to as self-attention, allowing the model to capture relationships between tokens within the same sequence. Conversely, in cross-attention mechanisms, queries, keys, and values are used to integrate information from different sources, as seen in sequence-to-sequence models. In this case, queries are generated from the decoder's previous output, while keys and values are obtained from the encoder's output. The queries represent elements in the target sequence that seek relevant information from the source sequence, while the keys represent elements in the source sequence that queries are compared against to determine their relevance.

An example of this occurs in language translation, where the encoder processes the input sequence in the source language, and the decoder processes sequences in the target language. In this context, the attention mechanism incorporates the most relevant information from the source sequence into the target sequence's context, enabling the model to learn semantic relationships between the two languages (Vaswani et al., 2017). The mechanism of cross-attention is formally defined as:

$$\text{CrossAttention}(Q_D, K_E, V_E) = \text{softmax}\left(\frac{Q_D K_E^T}{\sqrt{d_t}}\right) V_E, \tag{2.113}$$

where Q_D is a vector of dimension d_t representing the queries from the decoder, and K_E, V_E are vectors of dimension d_t representing the keys and values from the encoder, respectively.

Attention mechanisms have naturally integrated into the field of generative models. Visual Transformers (ViTs) (Dosovitskiy et al., 2020) have shown that attention mechanisms can improve image processing by dividing images into patches and processing them similarly to text sequences. This approach has been shown to surpass CNNs in image classification tasks, highlighting the adaptability and potential of transformers in handling diverse data modalities (Dosovitskiy et al., 2020). In diffusion models, self-attention mechanisms through ViTs enable the capture of long-range dependencies and intricate feature interactions within the data, improving coherence and detail in generated samples (Dosovitskiy et al., 2020). For text-to-image generation, cross-attention mechanisms condition the image synthesis process on textual input (Rombach et al., 2022). The input text is encoded into a sequence of embeddings, which are used as keys and values in the cross-attention layers, while latent representations of the intermediate image states serve as queries. This architecture allows the model to dynamically attend to specific textual features during each stage of the generation process, aligning the visual output with the semantic content of the text prompt (Ramesh et al., 2021; Rombach et al., 2022; Radford et al., 2021).

In text generation models such as BERT (Devlin et al., 2019a), transformers enable bidirectional context understanding, improving performance in text genera-

tion and comprehension tasks. CLIP (Radford et al., 2021) (Contrastive Language-Image Pre-training) employs transformers to encode text and ViTs to encode images, learning a joint latent space that aligns these embeddings and enabling multi-modal reasoning. The success of these models highlights the transformative role of transformers in advancing modern generative AI.

Despite their success, transformer architectures continue to face several challenges. A primary limitation is their substantial memory requirements, driven by the attention mechanism's quadratic complexity with respect to sequence length, making them prohibitively expensive for long sequences (Vaswani et al., 2017). This high memory consumption stems from the need to store large matrices for queries, keys, and values, and to compute their pairwise interactions. To address these issues, more efficient implementations such as Flash Attention (Dao et al., 2022), Reformer (Kitaev et al., 2020), and Sparse Attention (Zaheer et al., 2020) have been developed. These methods reduce memory usage and computational overhead by optimising the algorithms and data structures used to implement Equation 2.112. However, even these improvements have not yet achieved the efficiency required for widespread accessibility. Furthermore, transformers often encounter challenges related to interpretability and transparency, as their complex, high-dimensional operations obscure the model's decision-making processes (Vig, 2019).

2.1.7 Additional Generative Frameworks

Energy-based Models Energy-based models (EBMs) represent a class of models in which the relationship between data and probability is defined through an energy function. The energy function $E(\mathbf{x})$ assigns a scalar energy value to each possible configuration of the input data $\mathbf{x}$. The probability of observing a particular configuration is governed by the Boltzmann distribution (LeCun et al., 2006):

$$p(\mathbf{x}) = \frac{\exp(-E(\mathbf{x}))}{\int \exp(-E(\mathbf{x}))},$$

where the denominator, known as the partition function, normalises the distribution. EBMs are capable of representing complex, high-dimensional distributions,

but computing the partition function is generally intractable due to the requirement to integrate over all possible configurations of the data space (LeCun et al., 2006). As a result, approximate methods such as Markov Chain Monte Carlo (MCMC) and contrastive divergence are employed to estimate the partition function and model parameters (Osogami, 2017; Hinton, 2002; LeCun et al., 2006).

EBMs offer a more flexible and interpretable framework that contrasts with approaches like VAEs and GANs (Song and Kingma, 2021). While VAEs approximate likelihoods using variational lower bounds and GANs employ adversarial training to implicitly model data distributions, EBMs directly define an energy landscape where configurations with lower energy are more probable. This explicit formulation not only enhances interpretability but also enables seamless integration with other generative models. For instance, EBMs can refine the latent space of VAEs or stabilise GAN training by constraining the generator through energy-based regularisation (Che et al., 2020; Xiao et al., 2020). However, despite their versatility, the practical deployment of EBMs is often constrained by the computational complexity of estimating and minimising the energy function, driving ongoing research into more efficient learning and inference methods (Song and Kingma, 2021; LeCun et al., 2006).

Autoregressive Models Autoregressive generative models form a class of models that learn the data distribution by decomposing it into a sequence of conditional probabilities. Given a data vector $x = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$, the joint probability can be expressed as:

$$p(\mathbf{x}) = p(\mathbf{x}_1) \cdot p(\mathbf{x}_2 \mid \mathbf{x}_1) \cdot p(\mathbf{x}_3 \mid \mathbf{x}_1, \mathbf{x}_2) \cdots p(\mathbf{x}_n \mid \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n-1}).$$

This factorisation is particularly well-suited for processing and generating sequential data, such as text, time series, and audio, by generating each element of the data vector conditioned on the preceding elements, thereby capturing dependencies incrementally. Classic examples of autoregressive models include Recurrent Neural

Networks (RNNs) (Rumelhart et al., 1986) and Long Short-Term Memory (LSTM) networks (Staudemeyer and Morris, 2019), which model sequences by maintaining hidden states that evolve at each time step. More recent models, such as PixelCNN (Van Den Oord et al., 2016) for images, GPT (Brown et al., 2020) for text, and WaveNet (Oord et al., 2016) for audio generation, extend this principle by leveraging sequential generation to synthesise complex data distributions.

While the sequential nature of autoregressive models bears some resemblance to the iterative denoising process in diffusion models, the underlying mechanisms serve different purposes. In autoregressive models, sequentiality is inherent to the structure of the data, whereas in diffusion models, the iterative process is applied to progressively refine a single data sample through multiple steps of noise removal.

Autoregressive models directly model the data distribution through conditional probabilities, enabling exact likelihood optimisation without the need for complex approximations or adversarial objectives, as required by VAEs and GANs (Bond-Taylor et al., 2021). However, the sequential generation process can be computationally intensive, especially for large datasets or long sequences, since each step depends on all previous steps, limiting the potential for parallelisation during inference (Bond-Taylor et al., 2021).

Flow-based Models Flow-based models are a class of deep generative models that define a bijective (invertible) mapping between complex data distributions and simple, tractable distributions (Dinh et al., 2014). The key idea behind flow-based models is to transform the data $\mathbf{x}$ into a latent representation $\mathbf{z}$ through a series of *invertible* transformations, each with a tractable Jacobian determinant, enabling the computation of both forward and reverse mappings (Dinh et al., 2014, 2016).

Given an invertible and differentiable transformation f_θ, parameterised by θ, the likelihood of the data can be computed using the change of variables formula:

$$p(\mathbf{x}) = p(\mathbf{z}) \left| \det \left(\frac{\partial f_\theta^{-1}(\mathbf{z})}{\partial \mathbf{z}} \right) \right|,$$

where $\mathbf{z} = f_\theta(\mathbf{x})$ is the latent variable, $p(\mathbf{z})$ represents the probability in the la-

tent space, and the determinant of the Jacobian $\det\left(\frac{\partial f_\theta^{-1}(\mathbf{z})}{\partial \mathbf{z}}\right)$ reflects the change in volume under the transformation (Dinh et al., 2016). The tractability of the Jacobian determinant is crucial, as it enables exact log-likelihood computation and, by extension, the model's interpretability. However, the efficiency of this computation varies depending on the architectural design (Papamakarios et al., 2021; Kingma and Dhariwal, 2018; Dinh et al., 2016). Prominent architectures such as NICE (Dinh et al., 2014), RealNVP (Dinh et al., 2016), and Glow (Kingma and Dhariwal, 2018) exemplify the application of invertible transformations to model complex data distributions.

Flow-based models offer a unique approach to deep generative modelling by enabling exact and tractable likelihood computation, distinguishing them from Diffusion Models, VAEs, and GANs (Dinh et al., 2014). Flow-based models optimise the data likelihood directly, supporting high-quality sample generation (Papamakarios et al., 2021; Dinh et al., 2016). The invertible structure of flow-based models not only preserves bijective mappings but also enhances the interpretability of the latent space, enabling smooth and meaningful interpolations between data points (Kingma and Dhariwal, 2018; Dinh et al., 2016).

However, the invertibility constraint in flow-based models often imposes architectural limitations that are less restrictive in other generative models. Additionally, computing the Jacobian, particularly in high-dimensional spaces, greatly increases computational and memory overhead (Papamakarios et al., 2021). Despite these challenges, flow-based models have demonstrated remarkable performance across various tasks (Dinh et al., 2014) and are increasingly being explored as alternatives to conventional inversion techniques, including recent applications in ultrasound imaging (Yin et al., 2024b).

2.1.8 Metrics and Benchmarks for Image Generation

Evaluating image generation models in deep learning is essential, as their performance determines applicability across real-world scenarios. Robust metrics are necessary to ensure generated images uphold quality, integrity, perceptual accuracy,

and stylistic consistency. However, designing a universal metric for image quality remains challenging due to the subjective nature of visual perception. Image quality depends on factors such as texture, colour, shape, and perceptual similarity, making it difficult to create a single, encapsulating and definitive metric (Ledig et al., 2017; Salimans et al., 2016; Theis et al., 2015). Different metrics are designed to assess these various dimensions, each with distinct strengths and limitations. By combining multiple metrics, a more comprehensive evaluation can be achieved, capturing the diverse aspects that define image quality (Theis et al., 2015).

In supervised learning scenarios where paired data is available (e.g. VAEs), quality evaluation is relatively straightforward. Metrics such as L1 and L2 losses, SSIM, MS-SSIM, and LPIPS are commonly used to directly measure image quality and are often combined for a comprehensive assessment of the model. These metrics, which also serve as loss functions, are discussed in detail in Section 2.1.2, and their formulations and principles are equally applicable here.

In addition to these metrics, Peak Signal-to-Noise Ratio (PSNR) is a common metric used to evaluate the fidelity of a generated image relative to its ground truth by comparing the maximum possible pixel value to the MSE. PSNR is defined as:

$$\mathrm{PSNR}(\mathbf{x}, \mathbf{y}) = 10 \cdot \log_{10} \left(\frac{\mathrm{MAX}(\mathbf{x}, \mathbf{y})^2}{\mathrm{MSE}(\mathbf{x}, \mathbf{y})} \right),$$

where MAX denotes the maximum possible pixel value of the image (typically 255 for 8-bit images), and MSE represents the Mean Squared Error between the generated and ground truth images. PSNR is computationally simple and widely applied in image generation tasks. However, like MSE, PSNR is primarily sensitive to pixel-wise differences and fails to account for perceptual quality. This reliance on low-level, element-wise comparison often results in weak correlation with human subjective evaluations, especially in contexts where perceptual fidelity is more critical than exact pixel accuracy (Blau and Michaeli, 2018; Ledig et al., 2017). As a result, PSNR is less effective for evaluating tasks such as image denoising, compression, and generation, where perceptual metrics better reflect human visual perception.

On the other hand, quantitative assessment of generated image quality becomes

significantly more challenging in the absence of ground truth labels, as is often the case with generative models that synthesise novel, unseen data from a modelled distribution (e.g. GANs and Diffusion Models). In such cases, evaluation typically relies on comparing the underlying statistical properties of the generated data to those of the training set, using metrics that measure distributional similarity or feature space alignment.

One widely used method for evaluating generative models trained with unpaired data is the Fréchet Inception Distance (FID), originally designed to assess the quality of images generated by GANs. Similar to LPIPS, FID leverages deep neural networks pre-trained on large image datasets to evaluate the perceptual similarity between datasets. However, unlike LPIPS, which minimises the distance between activations of paired data, FID measures the distance between the *distributions* of generated and real images within the feature space of a pre-trained Inception network (Heusel et al., 2017). This approach is formulated as:

$$\text{FID} = \|\mu_r - \mu_g\|_2^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2\sqrt{\Sigma_r \Sigma_g}),$$

where μ_r and Σ_r represent the mean and covariance of the real images' features, and μ_g and Σ_g are the mean and covariance of the generated images' features. This formulation is equivalent to the *Wasserstein-2 distance* between two multivariate Gaussian distributions. FID captures both the *quality* (through the squared difference in means) and *diversity* (through the trace term involving covariance matrices) of the generated images, making it a popular metric. In the absence of labels, the diversity measure intrinsic to FID serves as a surrogate for assessing the generative model's generalisation ability to produce a varied sample distribution without mode collapse. FID has been shown to correlate well with human perceptual judgments, underscoring its value in generative model evaluation (Heusel et al., 2017; Chong and Forsyth, 2020; Kynkäänniemi et al., 2019).

However, FID has several limitations. It is sensitive to the choice of the feature extractor network, which can lead to inconsistent results across different architectures and datasets (Heusel et al., 2017). Additionally, FID depends on batch size,

requiring a large number of samples to produce stable estimates. This poses challenges in data-limited applications or when generating large quantities of high-quality samples is computationally expensive (Chong and Forsyth, 2020; Bińkowski et al., 2018). Moreover, while FID effectively detects variations in image quality, small differences can disproportionately affect the score, which may not always reflect the true performance of the generative model (Borji, 2019). Finally, preprocessing steps such as resizing and normalisation can greatly impact the FID score, potentially resulting in misleading evaluations if not carefully standardised (Borji, 2019).

The Inception Score (IS) is another used metric for evaluating the quality of generative models. It leverages the output of a pre-trained Inception network (Szegedy et al., 2016) to assess image quality by measuring both the confidence of predicted class labels and the diversity of generated samples. Formally, IS is defined as:

$$\text{IS} = \exp\left(\mathbb{E}_{\mathbf{x}}\left[\text{KL}(p(\mathbf{y}|\mathbf{x})\|p(\mathbf{y}))\right]\right),$$

where $p(\mathbf{y}|\mathbf{x})$ is the predicted class distribution given image $\mathbf{x}$, and $p(\mathbf{y})$ is the marginal class distribution. IS favours generated images that are confidently classified into distinct classes, while penalising those with ambiguous predictions spread across multiple classes, reflecting low-quality or noisy samples. Maximising IS requires the conditional distribution $p(\mathbf{y}|\mathbf{x})$ to be highly peaked for individual samples, while the marginal distribution $p(\mathbf{y})$ remains uniformly distributed across classes, ensuring both image quality and class diversity.

Despite its widespread use, IS has several limitations. It is sensitive to the choice of the pre-trained network and the dataset used for training, which can introduce variability in results (Barratt and Sharma, 2018). Moreover, IS does not penalise low intra-class diversity, allowing models to achieve high scores by generating high-quality but visually similar images within a limited number of classes (Borji, 2019). IS also assumes that generated images should span the full range of classes present in the training data, which may not align with tasks involving the generation of out-of-distribution samples.

The Kernel Inception Distance (KID) is a metric closely related to the Fréchet

Inception Distance (FID) but introduces key differences in its formulation and properties. KID measures the squared Maximum Mean Discrepancy (MMD) between the Inception representations of real and generated images, measured by a polynomial kernel that distributional similarity (Bińkowski et al., 2018). KID is defined as:

$$\text{KID} = \frac{1}{m(m-1)} \sum_{i \neq j} k(x_i, x_j) + \frac{1}{n(n-1)} \sum_{i \neq j} k(y_i, y_j) - \frac{2}{mn} \sum_{i,j} k(x_i, y_j),$$

where $k(x, y)$ is the polynomial kernel function, and x_i and y_j represent the features of real and generated images, respectively.

A key advantage of KID is that it provides an unbiased estimate of the similarity between generated and real images, regardless of sample size. Unlike FID, which relies on estimating large-sample covariances and can introduce bias, KID remains accurate and reliable even when the number of available images is limited. This property makes KID particularly useful in scenarios with restricted datasets or computational constraints (Bińkowski et al., 2018). However, similar to FID, KID is sensitive to the choice of feature extractor, kernel type, and kernel parameters. This sensitivity requires careful tuning to ensure meaningful evaluations across different applications (Chong and Forsyth, 2020).

Finally, user studies offer a subjective yet highly valuable method for evaluating the quality of generated images (Mantiuk et al., 2012). In these studies, human participants assess the realism or quality of images produced by different models, providing direct insight into how well the generated outputs align with human perception (Mantiuk et al., 2012; Rombach et al., 2022; Saharia et al., 2022). While highly informative, user studies are time-consuming and resource-intensive, and need careful experimental design to minimise biases. Moreover, the results are often subject to variability influenced by participant demographics and experience, posing challenges to reproducibility and generalisability.

While no single metric can fully encapsulate the quality of images generated by deep learning models, a combination of complementary approaches—such as objective metrics (e.g., element-wise metrics, FID, IS, KID) and subjective evaluations

like user studies—offers a more comprehensive assessment. The selection of metrics should be driven by the specific requirements of the task, with a clear understanding of each method's strengths and limitations.

2.2 Ultrasound Tomography

Ultrasound tomography (UT) has emerged as a well-established technique in the field of medical imaging, primarily due to its non-invasive, portable, real-time, cost-effective, and safety attributes (Duric et al., 2005; Watson, 2022). UT utilises high-frequency sound waves, typically between 2 and 15 MHz (Szabo, 2013), to characterise internal structures by transmitting acoustic waves into the body and capturing the echoes produced by their interaction with different tissues. These echoes form a distinctive *footprint* of the body's internal composition, governed by the principles of wave propagation and tissue acoustics (Calderon Agudo et al., 2018; Szabo, 2013).

The algorithms employed to reconstruct images from the recorded data are known as computerised tomography algorithms. Naturally, these algorithms are founded on the principles of wave physics, and entail solving the inverse problem that maps ultrasound footprints back to the physical properties of the tissues that generated them (Guasch et al., 2020; Calderon Agudo et al., 2018; Wiskin et al., 2007; Sandhu et al., 2015). Ray-based ultrasound tomography algorithms are commonly employed as they offer a computationally efficient means of reconstructing ultrasound images. Such methods model wave propagation by approximating finite-frequency sound as infinite-frequency rays travelling along straight or slightly curved paths (straight-ray and bent-ray approximations), achieving simplification and computational cost suppression by neglecting scattering and diffraction effects that are intricate to model (Williamson, 1991).

While efficient, the simplifications taken in ray-based methods imposes fundamental limitations on the resolution and quality of the reconstructed images. This is specially true in highly heterogeneous media where ray approximations cannot account for phase shifts and additional travel times introduced by heavy scattering and diffraction phenomena (Williamson, 1991). As a result, ray-based tomography

methods are limited in spatial resolution, and are insufficient for resolving the fine anatomical details of complex tissue structures. Full-waveform inversion (FWI) arrises as a natural solution to the limitations of ray-based tomography. Instead of approximating wavefields as infinite-frequency rays, FWI models the full physics of wave propagation by solving the wave equation in its entirety, thus capturing the complete range of wave phenomena (Tarantola, 1984; Virieux and Operto, 2009). In doing so, FWI is able to increase the spatial resolution of conventional ray-based tomography, for the same experiment, by one order of magnitude per modelled dimension (Strong, 2023).

FWI, was firstly introduced in the field of seismology, accredited to Tarantola (1984), and has since become a standard practice in both industry and academia for characterising the subsurface of the Earth. More recently, FWI was introduced to the field of medical imaging, where it has proven itself superior over traditional UT techniques in terms of spatial resolution and image quality (Figure 2.12), marking it as an active area of research with promising clinical implications (Pratt et al., 2007; Guasch et al., 2020; Strong, 2023; Robins, 2021; Calderon Agudo et al., 2018; Bates et al., 2022; Marty et al., 2022; Bernard et al., 2017).

However, FWI comes with its own set of challenges. The technique relies on complex non-linear optimisation processes, requiring extensive computational resources for voluminous datasets and long processing times (Warner et al., 2013), which hinder its application in real-time clinical imaging (Calderon Agudo, 2018). In addition, FWI suffers from all the pitfalls of local inversion algorithms. It is highly sensitive to the initial model parameters and typically requires accurate prior information on the modelled medium and high signal-to-noise ratio (SNR) to avoid convergence to suboptimal local minima, a set of conditions that is often challenging to meet in practice (Robins et al., 2021; Warner et al., 2013; Warner and Guasch, 2016; Cudeiro-Blanco et al., 2022).

The following sections will delve into the principles of FWI, its applications in medical imaging, and recent efforts to integrate FWI with deep learning techniques. While in this thesis I do not directly perform FWI reconstructions, FWI serves as a conceptual benchmark for the outcomes aimed for in this work and therefore war-

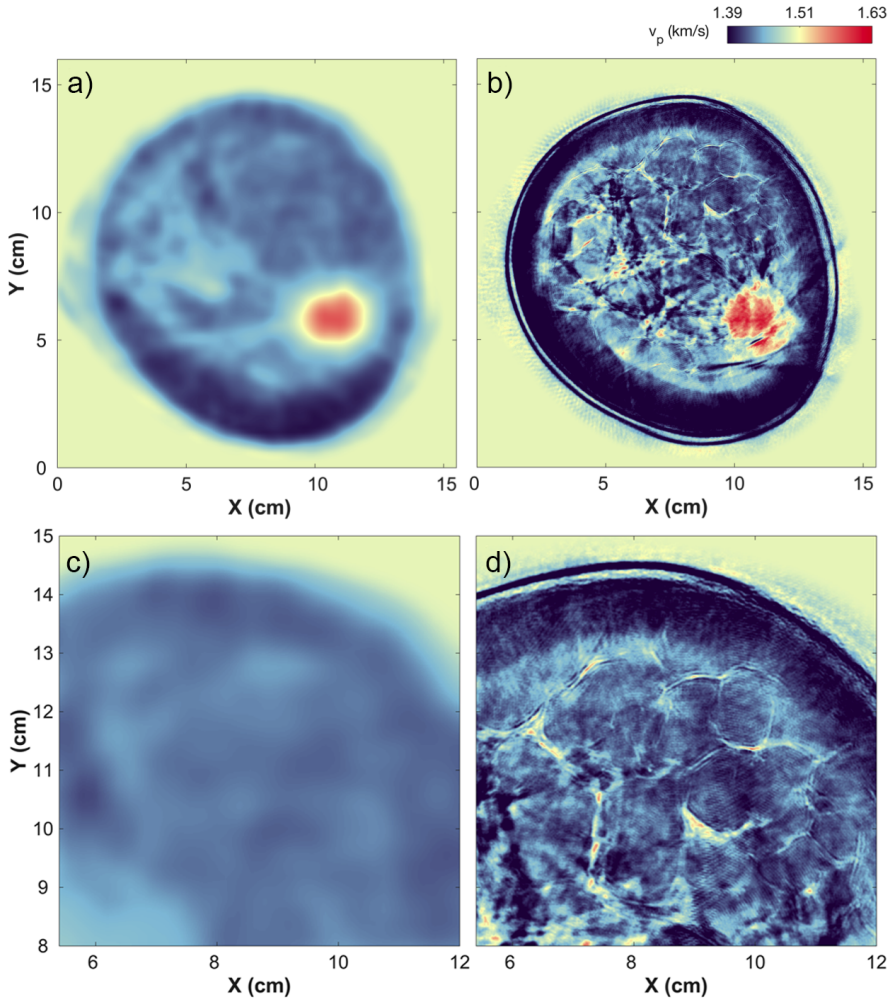


Figure 2.12: Comparison of the spatial resolution of a breast ultrasound image reconstructed using ray-based tomography (a, c) and FWI (b, d). (c) and (d) are zoomed in versions of (a) and (b), respectively. The FWI recovered image shows a higher level of detail and resolution, capturing fine anatomical structures that are not visible in the ray-based image. Modified after Calderon Agudo et al. (2018).

rants considerate level of discussion. Interestingly, FWI can be formulated as a form of neural network, a perspective that naturally motivates its integration with deep learning approaches. As will be explored later, numerous studies have sought not only to enhance FWI through deep learning but also to reframe FWI as a deep learning problem, with some even developing networks that intelligently process vast amounts of waveform data to construct physical models of the medium. In a somewhat liberal usage by the deep learning community, "FWI" is now broadly referred to any algorithm that does just that: processes acoustic waveform data in its entirety and performs a series of steps, whatever they may be, to reconstruct a spatial model

of the medium that underlies that data. It is in this expanded sense that FWI is explored in this thesis.

2.2.1 Full-waveform Inversion

> **Note to the reader:** in FWI literature, the term *model* is used to refer to the spatially-distributed physical properties of the medium being imaged, such as the speed of sound, density, and attenuation coefficients. This is distinct from the term *model* as used in deep learning, where it refers to the neural network architecture and its learned parameters, while the physical model is referred to as *samples* or *images*. Predominantly throughout this thesis, the terms *model* and *samples* will be used in the context of deep learning, except for this specific section where *model* will be used in the context of FWI to maintain consistency with the literature.

Local Inversion

The waveforms of an UT experiment can be modelled through forward modelling operator, generalised by the following equation:

$$\mathcal{G}(\mathbf{m}) = \mathbf{d}, \tag{2.114}$$

where $\mathcal{G}$ is the forward operator simulating wave propagation that maps the model $\mathbf{m}$, parameterised by the physical properties of the medium, to the modelled data $\mathbf{d}$ (Warner et al., 2013). In turn, the inverse problem of UT is to recover the optimal model $\mathbf{m}^*$ from the observed data $\mathbf{d}_{\text{obs}}$ by solving the inverse problem:

$$\mathbf{m}^* = \mathcal{G}^{-1}(\mathbf{d}_{\text{obs}}). \tag{2.115}$$

The operator $\mathcal{G}$ contains the discretised wave equation that describes the physics of the experiment for a given source-wavelet and acquisition geometry (Warner et al., 2013). Due to the non-linear nature of the wave equation, the inverse problem is ill-posed, meaning that $\mathcal{G}^{-1}$ cannot be directly defined (Warner et al., 2013). Non-linear

inverse problems of this nature are well-studied in the field of applied mathematics, where they are typically solved using local or global iterative optimisation algorithms (Seo and Woo, 2012).

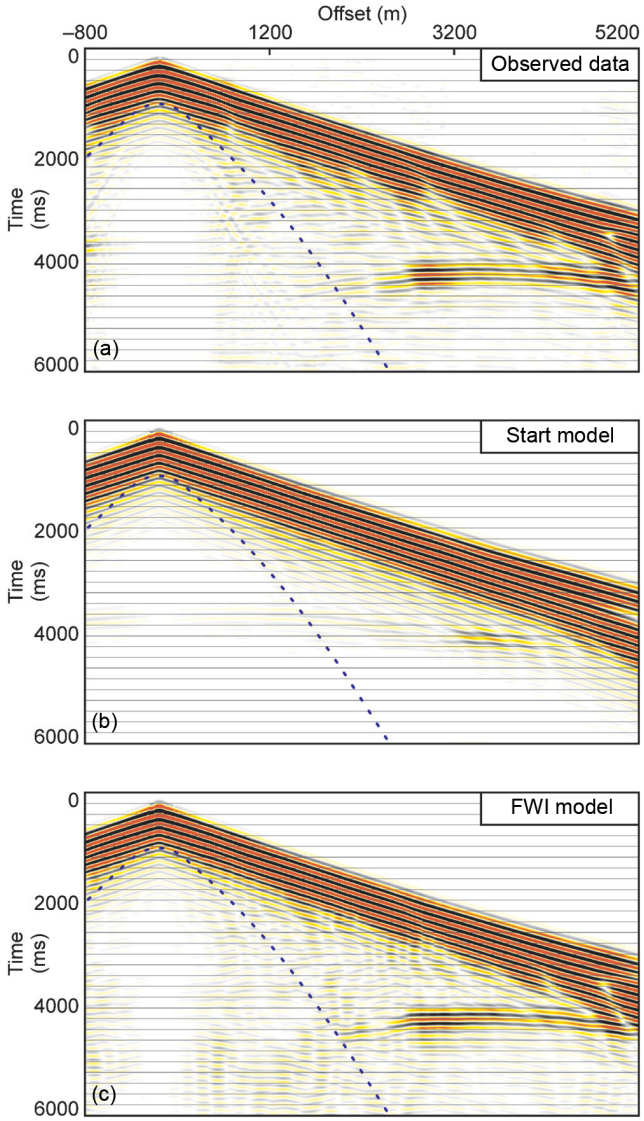


Figure 2.13: A series of shot gathers illustrating the FWI process. A starting model lacking the complex heterogeneities of the ground truth model is iteratively updated until the simulated data closely matches the observed data. The images depict one shot gather of the (a) observed data, (b) modelled data of the starting model, and (c) modelled data of the final model recovered by FWI. The shot gathers are represented as an amplitude map of the waveforms recorded over time (y-axis) at each receiver location (x-axis). Modified after Warner et al. (2013)

FWI is fundamentally a local optimisation algorithm (Figure 2.5). The high dimensionality typical of real-world FWI applications — often reaching billions of model parameters — limits the practicality of global optimisation methods, which

require exploring an extensive search space to identify an optimal solution. Like most local optimisation approaches, FWI seeks to iteratively adjust an initial model using locally linearised approximations until the simulated data aligns closely with the observed data (Warner et al., 2013). For small adjustment steps, the linear updates are sufficiently good to approximate the non-linear nature of the inverse problem. The core idea behind FWI is that, if the simulated waveforms of a modelled physical medium match the observed signals, then the model is a reasonable representation of the true physical medium that produced these signals (Figure 2.13). In turn, this match is driven by the minimisation of a misfit objective function, which quantifies the difference between the observed and simulated waveforms (Virieux and Operto, 2009; Warner et al., 2013). Broadly, the classic implementation of the FWI algorithm takes the following steps:

1. **Initialisation:** Select an initial model $\mathbf{m}_0$, ideally close to the true model, parameterised by physical properties of the medium.

2. **Forward Modelling:** Solve the wave equation over the model to generate synthetic data.

3. **Misfit & Gradient Computation:** Evalutate the misfit $\phi(\mathbf{m})$ for observed and synthetic data and compute its gradient with respect to the model $\nabla_{\mathbf{m}}\phi(\mathbf{m})$

4. **Parameter Update:** Adjust $\mathbf{m}$ using $\nabla_{\mathbf{m}}\phi(\mathbf{m})$ in a descent method, e.g., gradient descent or Gauss-Newton methods.

5. **Convergence Check:** Repeat until convergence criteria (e.g., objective function threshold, number of iterations, etc.) are met.

6. **Model Output:** Return the optimal model $\mathbf{m}^*$ as the final output.

Modelling the Wave Equation

The wave equation describes the spatial and temporal propagation of mechanical waves within a physical medium, capturing the harmonic transfer of energy across both dimensions. Mathematically, it is expressed as a coupled second-order linear partial differential equation (PDE), which relates the spatial and temporal sec-

ond derivatives of wave-dependent properties of the medium. In seismic imaging, the wave equation models the propagation of mechanical waves through the Earth, whereas in medical ultrasound imaging, it describes the transmission of these waves through biological tissues. This equation can be derived by integrating conservation principles from continuum mechanics with the constitutive stress-strain relationship of linearly elastic materials (Hooke's law) (Aki and Richards, 2002). For an unbounded, non-attenuating, anisotropic, and heterogeneous elastic medium, the wave equation is formulated as follows (Aki and Richards, 2002):

$$\rho(x_i)\frac{\partial^2 u_i}{\partial t^2} = \frac{\partial \tau_{ij}}{\partial x_j} + f_i, \tag{2.116}$$

where τ_{ij} is the stress tensor is defined by the constitutive relationship:

$$\tau_{ij} := C_{ijkl}\frac{\partial u_k}{\partial x_l}, \tag{2.117}$$

with u_i denoting the infinitesimal displacement vector at time t and position x_i, $\rho(x_i)$ as the medium's density, and f_i the body force vector in the i^{th} direction. Here, *anisotropy* signifies that the material properties vary with direction. This directional dependence is encoded in the fourth-order elasticity tensor C_{ijkl}, consisting of 21 independent parameters describing the medium's deformation response to stress across different axial planes. *Inhomogeneity* indicates spatial variability, where properties like density $\rho(x_i)$ and the elastic constants C_{ijkl} fluctuate across space, reflecting the complexities seen in geological formations and biological tissues. *Elasticity*, as governed by Hooke's law, denotes the medium's capacity to deform under stress and to return to its original configuration once the stress is relieved, and is a property found in most solid materials. The formulation of Equation 2.116 neglects phenomena of *attenuation*, the energy dissipation due to internal friction; *viscoelasticity*, which captures both viscous and elastic behaviours; and *thermoelastic* and *dispersive* effects, which are more relevant at high frequencies or smaller scales. Although these factors are often omitted in FWI due to computational constraints, they can play a crucial role in specific mediums and experiments where such properties are more pronounced. Modelling of these properties is more relevant in

geophysical studies (Calderón Agudo et al., 2018; Warner et al., 2013; Groos et al., 2014).

The wave equation presents as a complex, multi-dimensional, non-linear partial differential equation. Analytical solutions become infeasible in complex, heterogeneous, and anisotropic media, where the high parametrisation and varying material properties render closed-form solutions unattainable. Solving it requires numerical solvers. Common numerical methods include finite-difference, finite-element, and spectral methods, which discretise the model into a grid of cells and solve the wave equation at each point in simpler forms with numerical approximations (De Basabe and Sen, 2015; Kukreja et al., 2016). Discussing the specifics of these methods is beyond the scope of this revision, but it is important to note that numerical solvers play a critical role in FWI and must not be overlooked in its implementations, as they can often be a significant bottleneck to the convergence and computational cost of the algorithm.

In practical implementations of FWI, where high-resolution imaging is often required, numerically modelling all parameters in Equations 2.116 and 2.117 is computationally prohibitive. Consequently, simplified versions of the wave equation are commonly used, often by assuming simpler material properties at the cost of reduced modelling accuracy. A common simplification to the wave equation involves solving it isotropically, assuming uniform properties of the medium in all directions. Under this assumption, the stress tensor is defined by only 2 independent parameters, Lamé's parameters λ and μ (Aki and Richards, 2002):

$$\tau_{ij} = \lambda \frac{\partial u_k}{\partial x_k} \delta_{ij} + 2\mu \frac{\partial u_i}{\partial x_j}, \tag{2.118}$$

where δ_{ij} denotes the Kronecker delta. Substituting this form into Equation 2.116 yields the isotropic wave equation:

$$\rho(x_i) \frac{\partial^2 u_i}{\partial t^2} = \frac{\partial}{\partial x_j} \left(\lambda(x_i) \delta_{ij} \frac{\partial u_k}{\partial x_k} + 2\mu(x_i) \frac{\partial u_i}{\partial x_j} \right) + f_i, \tag{2.119}$$

where $\rho(x_i) \frac{\partial^2 u_i}{\partial t^2}$ represents the density-weighted acceleration of displacement u_i, with $\rho(x_i)$ denoting the spatially varying density, and $\lambda(x_i)$ and $\mu(x_i)$ the Lamé param-

eters, which also vary spatially in inhomogeneous media. The term $\lambda(x_i)\delta_{ij}\frac{\partial u_k}{\partial x_k}$ corresponds to isotropic stress, where λ serves as the first Lamé parameter, affecting only the normal stresses. The term $2\mu(x_i)\frac{\partial u_i}{\partial x_j}$ captures the shear stresses, with μ representing the shear modulus (second Lamé parameter), responsible for the medium's response to shear deformation (Aki and Richards, 2002).

As a further simplification, in homogeneous media where material properties are constant in space, the wave equation is reduced to:

$$\rho\frac{\partial^2 u_i}{\partial t^2} = \lambda\frac{\partial}{\partial x_i}\left(\frac{\partial u_k}{\partial x_k}\right) + 2\mu\frac{\partial^2 u_i}{\partial x_j \partial x_j} + f_i, \tag{2.120}$$

or, in vector notation:

$$\rho\frac{\partial^2 \mathbf{u}}{\partial t^2} = \lambda\nabla(\nabla\cdot\mathbf{u}) + 2\mu\nabla^2\mathbf{u} + \mathbf{f}. \tag{2.121}$$

Using the identity $\nabla^2\mathbf{u} = \nabla(\nabla\cdot\mathbf{u}) - \nabla\times(\nabla\times\mathbf{u})$, Equation 2.121 transforms into

$$\rho\frac{\partial^2 \mathbf{u}}{\partial t^2} = \lambda\nabla(\nabla\cdot\mathbf{u}) + 2\mu\left(\nabla(\nabla\cdot\mathbf{u}) - \nabla\times(\nabla\times\mathbf{u})\right) + \mathbf{f}, \tag{2.122}$$

which can be further simplified to:

$$\rho\frac{\partial^2 \mathbf{u}}{\partial t^2} = (\lambda+2\mu)\nabla(\nabla\cdot\mathbf{u}) - \mu\nabla\times(\nabla\times\mathbf{u}) + \mathbf{f}. \tag{2.123}$$

Taking the divergence of Equation 2.123 isolates the compressive components, as $\nabla\cdot(\nabla\times\mathbf{u}) = 0$ removes any curling terms (Strong, 2023), yielding:

$$\frac{1}{v_p^2}\frac{\partial^2}{\partial t^2}(\nabla\cdot\mathbf{u}) = \nabla^2(\nabla\cdot\mathbf{u}) + \nabla\cdot\mathbf{f}, \tag{2.124}$$

where

$$v_p = \sqrt{\frac{\lambda+2\mu}{\rho}}, \tag{2.125}$$

defines the speed of propagation of compressional waves v_p through the medium, parameterised by the Lamé parameters and density. This is known as the isotropic acoustic wave equation, which governs the propagation of purely compressional waves in homogeneous isotropic media. The quantity v_p, known too as speed of sound (SoS)

and P-wave speed, is commonly used to characterise physical models in FWI imaging.

Similarly, the shear components can be isolated by taking the curl of Equation 2.123. Applying the vector calculus properties $\nabla \times (\nabla(\nabla \cdot \mathbf{u})) = \nabla \times (\nabla k) = 0$ and $\nabla \times (\nabla \times (\nabla \times \mathbf{u})) = -\nabla^2(\nabla \times \mathbf{u})$ yields:

$$\frac{1}{v_s^2}\frac{\partial^2}{\partial t^2}(\nabla \times \mathbf{u}) = \nabla^2(\nabla \times \mathbf{u}) + \nabla \times \mathbf{f}, \tag{2.126}$$

where

$$v_s = \sqrt{\frac{\mu}{\rho}}, \tag{2.127}$$

defines the shear wave (S-wave) speed v_s, parameterised by the shear modulus and density, and governing the propagation of shear waves in homogeneous isotropic media. Comparing Equations 2.125 and 2.127 shows that shear waves travel more slowly than compressional waves.

The accuracy of forward modelling in FWI relies heavily on the chosen wave equation. In essence, any real-occurring phenomena of wave propagation that goes unmodelled is treated as noise by FWI. So the more complete the wave equation, the more precise the inversion. Although including anisotropy can significantly improve accuracy in FWI (Warner et al., 2013), for most ultrasound imaging applications, the human body can often be approximated as an isotropic medium without substantial loss of fidelity (Wood, 1971; Strong, 2023). However, the acoustic simplification may not fully capture compressional-to-shear conversions at bone-tissue interfaces. Nonetheless, this approximation is frequently necessary due to the computational processing and memory demands of elastic modelling. Ultimately, the choice of wave equation represents a balance between computational feasibility and convergence requirements.

Adjoint State Method

As a local inversion algorithm, FWI iteratively updates an initial model $\mathbf{m} \in \mathbb{R}^{d_1 \times d_2 \times d_3 = M}$ with small, localised perturbations $\delta\mathbf{m}$, under linear assumptions, to obtain a model closer to the true solution:

$$\mathbf{m} \leftarrow \mathbf{m} + \delta\mathbf{m}. \tag{2.128}$$

The perturbation $\delta\mathbf{m}$ is chosen to minimise an objective function, typically the mean squared error between the observed and modelled data:

$$\phi(\mathbf{m}) = \frac{1}{2}\|\mathcal{G}(\mathbf{m}) - \mathbf{d}_{\text{obs}}\|^2 = \frac{1}{2}\sum_{n_s}\sum_{n_r}\sum_{n_t}|d_{s,r,t} - d_{\text{obs}_{s,r,t}}|^2, \tag{2.129}$$

where $d_{s,r,t}$ and $d_{\text{obs}_{s,r,t}}$ denote modelled and observed data of source s, at receiver r, and time t, respectively.

Following the derivation from Virieux and Operto (2009), we can apply a Taylor series expansion on Equation 2.129 around the vicinity $\mathbf{m} = \mathbf{m}_0 + \delta\mathbf{m}$ of a starting model $\mathbf{m}_0$:

$$\phi(\mathbf{m}) = \phi\left(\mathbf{m}_0 + \delta\mathbf{m}\right) = \phi\left(\mathbf{m}_0\right) + \delta\mathbf{m}^T\frac{\partial\phi}{\partial\mathbf{m}}\bigg|_{\mathbf{m}_0} + \frac{1}{2}\delta\mathbf{m}^T\frac{\partial^2\phi}{\partial\mathbf{m}^2}\bigg|_{\mathbf{m}_0}\delta\mathbf{m} + \mathcal{O}\left(\delta\mathbf{m}^3\right), \tag{2.130}$$

and differentiate it with respect to $\delta\mathbf{m}$ to obtain:

$$\begin{aligned}\frac{\partial\phi}{\partial\mathbf{m}}\bigg|_{\mathbf{m}} &= \frac{\partial\phi}{\partial\mathbf{m}}\bigg|_{\mathbf{m}_0} + \left(\frac{1}{2}\delta\mathbf{m}^T\frac{\partial^2\phi}{\partial\mathbf{m}^2}\bigg|_{\mathbf{m}_0}\right)^T + \left(\frac{1}{2}\frac{\partial^2\phi}{\partial\mathbf{m}^2}\bigg|_{\mathbf{m}_0}\delta\mathbf{m}\right) + \frac{\partial}{\partial\delta\mathbf{m}}\mathcal{O}\left(\delta\mathbf{m}^3\right)\\ &= \frac{\partial\phi}{\partial\mathbf{m}}\bigg|_{\mathbf{m}_0} + \frac{\partial^2\phi}{\partial\mathbf{m}^2}\bigg|_{\mathbf{m}_0}\delta\mathbf{m} + \mathcal{O}\left(\delta\mathbf{m}^2\right).\end{aligned} \tag{2.131}$$

Setting the gradient to zero, we obtain the model perturbation that minimises the objective function:

$$\delta\mathbf{m} = -\left(\frac{\partial^2\phi}{\partial\mathbf{m}^2}\bigg|_{\mathbf{m}_0}\right)^{-1}\frac{\partial\phi}{\partial\mathbf{m}}\bigg|_{\mathbf{m}_0} - \mathcal{O}\left(\delta\mathbf{m}^2\right), \tag{2.132}$$

and by neglecting the higher-order terms, we derive the linearised first-order approximation, known as the Born approximation, of the model perturbation, leading to the conventional gradient descent method:

$$\delta \mathbf{m} \approx -\mathbf{H}^{-1} \nabla_{\mathbf{m}} \phi, \tag{2.133}$$

where $\nabla_{\mathbf{m}}\phi = \nabla_{\mathbf{m}}\phi(\mathbf{m})$ is the gradient of the misfit with respect to the current model parameters, defined by:

$$\nabla_{\mathbf{m}}\phi(\mathbf{m}) \equiv \frac{\partial \phi(\mathbf{m})}{\partial \mathbf{m}} \equiv \begin{bmatrix} \frac{\partial \phi}{\partial m_1} \\ \frac{\partial \phi}{\partial m_2} \\ \vdots \\ \frac{\partial \phi}{\partial m_M} \end{bmatrix}, \tag{2.134}$$

and $\mathbf{H} = \mathbf{H}(\mathbf{m})$ is the Hessian matrix containing second-order derivatives, representing the curvature of the error surface around the current model:

$$\mathbf{H}(\mathbf{m}) \equiv \frac{\partial^2 \phi(\mathbf{m})}{\partial \mathbf{m}^2} \equiv \begin{bmatrix} \frac{\partial^2 \phi}{\partial m_1^2} & \frac{\partial^2 \phi}{\partial m_1 \partial m_2} & \cdots & \frac{\partial^2 \phi}{\partial m_1 \partial m_M} \\ \frac{\partial^2 \phi}{\partial m_2 \partial m_1} & \frac{\partial^2 \phi}{\partial m_2^2} & \cdots & \frac{\partial^2 \phi}{\partial m_2 \partial m_M} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \phi}{\partial m_M \partial m_1} & \frac{\partial^2 \phi}{\partial m_M \partial m_2} & \cdots & \frac{\partial^2 \phi}{\partial m_M^2} \end{bmatrix}. \tag{2.135}$$

For large-scale models ($M \approx 10^6 - 10^9$) the Hessian matrix is often computationally intractable to compute, invert, and store, so it is typically approximated. In its simplest approximation, the Hessian is replaced by a constant-valued scalar, which results in the conventional gradient descent method:

$$\delta \mathbf{m} \approx -\alpha \nabla_{\mathbf{m}} \phi, \tag{2.136}$$

where α is the constant-valued step size, or learning rate, replacing the Hessian, which controls the size of the model update. Research on optimal learning rates and Hessian approximations for FWI is extensive (Pratt et al., 1998), but beyond the scope of this review.

The gradient $\nabla_{\mathbf{m}}\phi$, as shown in Equation 2.133, can theoretically be computed

by explicitly perturbing each model parameter individually and observing the resulting change in the objective function. This would require $M + 1$ forward runs to model the wavefields and construct the sensitivity matrix $\frac{\partial \mathbf{d}}{\partial \mathbf{m}}$. While straightforward in principle, this method is computationally prohibitive for realistic models due to the significant computational overhead. This high computational demand motivates the use of adjoint state methods to calculate the misfit gradient. By reformulating the gradient calculation in terms of adjoint wavefields rather than explicit sensitivity matrices, the adjoint state method provides an efficient alternative that requires only two wave equation solutions per iteration — one forward and one adjoint (Plessix, 2006). This reduction transforms an otherwise impractical computation into one that is achievable for large-scale FWI problems.

The adjoint formulation begins with the linear relationship between the wavefield $\mathbf{u}$ and a source term for propagation, represented in Equation 2.123 by the external force $\mathbf{f}$ but denoted here as $\mathbf{s}$ for clarity:

$$\mathbf{A}\mathbf{u} = \mathbf{s}, \tag{2.137}$$

where the source term $\mathbf{s}$ is the outcome of the linear operator $\mathbf{A} = \mathbf{A}(\mathbf{m})$ acting on the wavefield $\mathbf{u} = \mathbf{u}(\mathbf{m})$. In turn, the operator $\mathbf{A}$ is a matrix containing the discretisation of the wave equation for a given model and acquisition parameters. Differentiating Equation 2.137 with respect to the model yields:

$$\frac{\partial \mathbf{A}}{\partial \mathbf{m}}\mathbf{u} + \mathbf{A}\frac{\partial \mathbf{u}}{\partial \mathbf{m}} = 0, \tag{2.138}$$

where the source term vanishes as it is independent of the model. Equation 2.138 can then be rearranged to isolate the derivative of the wavefield with respect to the model:

$$\frac{\partial \mathbf{u}}{\partial \mathbf{m}} = -\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial \mathbf{m}}\mathbf{u}. \tag{2.139}$$

The data $\mathbf{d}$ can be extracted from $\mathbf{u}$ at the receiver positions via a diagonal matrix $\mathbf{R}$:

$$\mathbf{d} = \mathbf{R}\mathbf{u}, \tag{2.140}$$

and we note that derivative of the data with respect to the model, also known as the Jacobian or sensitivity matrix (Shin et al., 2001), can be expressed as a function of the operator $\mathbf{A}$ and the wavefield $\mathbf{u}$ using Equation 2.139:

$$\mathbf{J} = \frac{\partial \mathbf{d}}{\partial \mathbf{m}} = \mathbf{R}\frac{\partial \mathbf{u}}{\partial \mathbf{m}} = -\mathbf{R}\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial \mathbf{m}}\mathbf{u}. \tag{2.141}$$

Applying the chain rule, the gradient of the misfit function in Equation 2.129 with respect to the model can be reformulated as the product of the data residual and the Jacobian, noting that while $\mathbf{d}$ is a product of the model $\mathbf{m}$, the observed data $\mathbf{d}_{\text{obs}}$ is independent of it:

$$\nabla_{\mathbf{m}}\phi(\mathbf{m}) = \frac{1}{2}\frac{\partial}{\partial \mathbf{m}}\|\mathbf{d} - \mathbf{d}_{\text{obs}}\|^2 = \left(\frac{\partial \mathbf{d}}{\partial \mathbf{m}}\right)^T (\mathbf{d} - \mathbf{d}_{\text{obs}}) = \mathbf{J}^T\delta\mathbf{d}, \tag{2.142}$$

where $\delta\mathbf{d} = \mathbf{d} - \mathbf{d}_{\text{obs}}$ is the data residual. Substituting the Jacobian from Equation 2.141 into Equation 2.142 gives:

$$\nabla_{\mathbf{m}}\phi(\mathbf{m}) = -\mathbf{u}^T\left(\frac{\partial \mathbf{A}}{\partial \mathbf{m}}\right)^T \mathbf{A}^{-T}\mathbf{R}^T\delta\mathbf{d}. \tag{2.143}$$

The final term of the gradient in Equation 2.143 can be interpreted as the back-propagation of the data residual through the model. When the matrix $\mathbf{R}$ represents the extraction of wavefields at receiver positions, $\mathbf{R}^T$ corresponds to re-injecting the wavefield at the receiver positions back into the model. Similarly, if $\mathbf{A}$ is spatially symmetric (i.e., self-adjoint), and it is in practice designed to be, and propagates the wavefield forward in time, the adjoint operator $\mathbf{A}^T$ is analogous to propagating the wavefield through the model backward in time, a concept not too dissimilar from back-propagation in deep-learning. This final term thus generates a residual wavefield $\delta\mathbf{u}$ by back-propagating the residual data as a source in reverse time, formalised as:

$$\mathbf{A}^T\delta\mathbf{u} = \mathbf{R}^T\delta\mathbf{d}, \tag{2.144}$$

which mirrors Equation 2.137. Thus, the gradient of the objective function with respect to $\mathbf{m}$ is given by the adjoint state method as:

$$\nabla_{\mathbf{m}}\phi(\mathbf{m}) = -\mathbf{u}^T \left(\frac{\partial \mathbf{A}}{\partial \mathbf{m}}\right)^T \delta\mathbf{u}. \tag{2.145}$$

Equation 2.145 tells us that computing the gradient of the misfit function with respect to the model requires first solving the wave equation forward in time to obtain the wavefield $\mathbf{u}$, then solving the adjoint wave equation backward in time to acquire the adjoint wavefield $\delta\mathbf{u}$. This adjoint wavefield is then multiplied by the gradient of the operator $\mathbf{A}$ with respect to the model, which is derived analytically. This process represents the zero lag of the temporal cross correlation of the forward wavefield for a particular source with the back-propagated wavefield generated by the data residuals at each receiver for that source, calculated at every point in the model, and weighted and modified by an analytical expression (Warner et al., 2013). For each source, this approach requires only two wavefield computations — one forward and one adjoint — rather than $M + 1$ computations, making FWI computationally feasible. The FWI algorithm is summarised in Algorithm 6.

FWI In Practice

Offering an unprecedented level of detail and resolution in ultrasound characterisation, FWI has transformed geophysical imaging and has gained popularity in ultrasound medical imaging (Guasch et al., 2020; Calderon Agudo, 2018; Strong, 2023; Calderon Agudo et al., 2018). By leveraging the entire wavefield, FWI produces high-resolution reconstructions that surpass traditional methods like ray-based travel-time tomography and migration (Warner et al., 2013; Virieux and Operto, 2009). Despite its achievements, FWI remains computationally intensive and practically challenging, particularly when applied in environments where the physics of wave propagation is complex. At the core of these challenges lies its inherently ill-posed nature, stemming from the highly non-linear character of the inverse problem, where no unique solution exists and many model solutions can fit the data equally well.

The non-linearity of FWI is exacerbated by the limited frequency range of

Algorithm 6 Full-Waveform Inversion (FWI) Algorithm

Require: Initial model parameters $\mathbf{m}_0$, observed data $\mathbf{d}_{\text{obs}}$, convergence threshold ϵ, maximum iterations $N_{\max}$

Ensure: Optimal model parameters $\mathbf{m}^*$

Set $k \leftarrow 0$

Initialise model $\mathbf{m} \leftarrow \mathbf{m}_0$

while $\|\nabla_{\mathbf{m}}\phi(\mathbf{m})\| > \epsilon$ **and** $k < N_{\max}$ **do**

1. Forward Modelling: Solve the forward wave equation to obtain the modelled data

$$\mathcal{G}(\mathbf{m}) = \mathbf{d}$$

2. Compute Data Residual: Calculate the data residual $\delta\mathbf{d} = \mathbf{d} - \mathbf{d}_{\text{obs}}$

3. Adjoint Wavefield: Compute the residual wavefield $\delta\mathbf{u}$ by solving the adjoint problem:

$$\delta\mathbf{u} = \mathbf{A}^{-T}\mathbf{R}^T\delta\mathbf{d}$$

4. Compute Gradient: Compute the gradient of the objective function $\phi(\mathbf{m})$ with respect to $\mathbf{m}$:

$$\nabla_{\mathbf{m}}\phi(\mathbf{m}) = -\mathbf{u}^T\left(\frac{\partial\mathbf{A}}{\partial\mathbf{m}}\right)^T\delta\mathbf{u}$$

5. Update Model: Update the model parameters $\mathbf{m}$ using a descent method, e.g., gradient descent:

$$\mathbf{m} \leftarrow \mathbf{m} - \alpha\nabla_{\mathbf{m}}\phi(\mathbf{m})$$

where α is the step size.

Increment $k \leftarrow k + 1$

end while

return $\mathbf{m}^* \leftarrow \mathbf{m}$

ultrasound data and inadequate starting models (Sirgue, 2006). At frequencies lower than 7 Hz, the optimisation landscape near the global minimum is sufficiently flat to allow for successful waveform inversion starting from an initial model derived from ray-based tomography (Sirgue, 2003). In addition, low-frequencies are less affected by attenuation effects, boosting SNRs. Without sufficient low-frequency data, FWI struggles to resolve the large-scale structures of the model, which are associated with these frequencies (Sirgue, 2003; Shah et al., 2012). The inversion becomes prone to cycle-skipping, a condition in which the inversion aligns incorrect phases between synthetic and observed data, ultimately leading to convergence towards erroneous final models trapped in local minima (Virieux and Operto, 2009) (Figure 2.14). Equally important, a starting model with insufficient or incorrect kinematics can

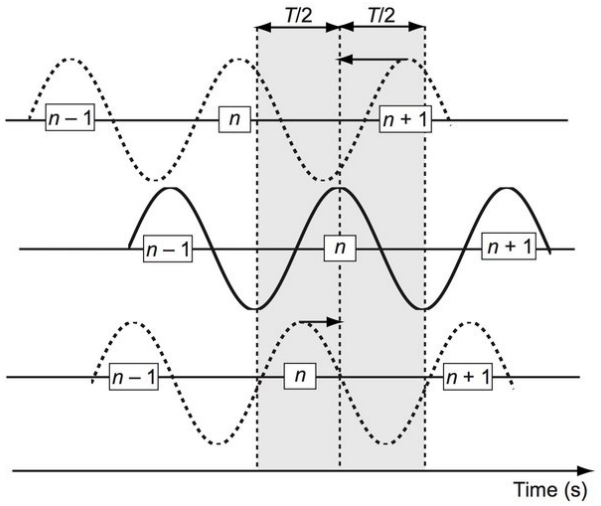


Figure 2.14: Cycle-skipping in FWI. The inversion process is susceptible to cycle-skipping when the phase of the synthetic data is misaligned with the observed data by more than half a wavelength. The solid black line indicates a monochromatic waveform with a period T as a function of time. The upper dashed line shows the modelled monochromatic waveforms with a time delay exceeding $T/2$. In this scenario, FWI will adjust the model so that the $(n+1)^{\text{th}}$ cycle of the modelled waveforms aligns with the n^{th} cycle of the observed waveform, misrepresenting the kinematics of the system. In the lower example, FWI modifies the model such that the modelled and recorded n^{th} cycles remain in-phase, as the time delay is less than $T/2$. Figure extracted from (Virieux and Operto, 2009).

also result in misaligned waveforms and lead to cycle-skipping (Warner et al., 2013; Shah et al., 2012). Successful FWI, therefore, hinges both on the quality of the initial model and the low-frequency content of the data, where one can compensate the other (Shah et al., 2012).

Obtaining the lowest frequencies essential for robust FWI, however, poses its own difficulties, as the bandwidth provided by acoustic sources is often limited (Sirgue, 2003; Robins et al., 2021; Cudeiro-Blanco et al., 2022). When such frequencies are achieved, they can still be obscured by environmental noise, which complicates matters further (Sirgue, 2003). In this case, it becomes even more crucial that the starting model is a truthful representation of the medium in order to mitigate such risks (Raknes and Arntsen, 2017; Virieux and Operto, 2009).

The body of research on strategies to mitigate cycle-skipping is extensive, including regularisation techniques, frequency-dependent schemes, multi-scale approaches, and even other refined variants of FWI algorithm. Details of such ap-

proaches are intentionally left out here, as they lie beyond the scope of this discussion. Notably, however, Adaptive Waveform Inversion (AWI), a variant of FWI, has emerged as a compelling solution for addressing cycle-skipping issues, particularly in cases with suboptimal starting models (Warner and Guasch, 2016). The foundational principles of AWI form the basis for the work presented in Chapter 3, where they will be discussed in more detail.

Transitioning from two-dimensional (2D) to three-dimensional (3D) FWI is critical for capturing the full complexity of wave propagation. In 2D, wavefield modelling relies on restrictive in-plane assumptions, leading to incomplete representations of wave dynamics and potentially inaccurate model reconstructions, especially in heterogeneous regions (Warner et al., 2013; Raknes and Arntsen, 2017; Warner et al., 2008; Calderon Agudo, 2018). The transition to 3D, however, introduces considerable computational and memory burdens. Not only the modelling dimensions are cubically scaled from $n \times n$ to $n \times n \times n$, 3D FWI demands computational scales on the order of n^6 due to the increased model volume, source requirements, and wave propagation across three dimensions (Warner et al., 2008). For large 3D models, the volume of temporal wavefield data alone can reach tens of terabytes, imposing challenging storage and processing requirements (Raknes and Arntsen, 2017; Warner et al., 2013). Access to high-performance computing (HPC) or GPU resources with tailored parallelisation strategies is therefore indispensable for managing these computational loads effectively (Raknes and Arntsen, 2017; Warner et al., 2013; Cueto et al., 2022).

Intensifying this challenge, the choice of wave equation remains a constant tug-of-war between computational feasibility and model convergence in FWI. Acoustic isotropic models, though more computationally efficient, fail to incorporate shear waves and mode conversions, introducing potential biases and kinematics mismatches in high-contrast elastic environments. Conversely, elastic models, which capture more complex wave interactions, require more computational resources and memory due to the additional degrees of freedom for shearing parameters (Raknes and Arntsen, 2017; Guasch et al., 2012). Despite the many developments to alleviate its computational burden, 3D elastic FWI remains one of the most resource-intensive applications of

the method.

In geophysics, FWI operates within characteristic P-wave velocities from 1450 to 4500 m/s and frequencies between 2 and 30 Hz for a 5 to 50m grid resolution (Calderon Agudo, 2018). In contrast, medical ultrasound requires resolutions at a much smaller scale, with grid spacings as fine as 0.5 mm (Calderon Agudo et al., 2018; Guasch et al., 2020). In practice, the need for fine resolution requires modelling at significantly higher frequencies, typically in the range of 0.2 to 2 MHz, to capture the finer spatial details essential for imaging clinically important anatomical features (Calderon Agudo, 2018; Strong, 2023), particularly in soft tissues, which exhibit much smaller acoustic contrasts compared to geological structures. As a consequence, the computational load of FWI in medical imaging is substantially higher, resulting in longer optimisation times for accurate inversions and often prohibiting its feasibility in real-time applications.

Brain imaging with FWI presents its own unique challenges. Conventional ray-based tomography, which primarily models reflected waves as straight or bent paths, is ineffective for brain imaging because of the high impedance contrast between the skull and its medium, and the low impedance contrast between the different types of soft tissues. This means that soft tissues are only registered in the data as transmission signals, which are often obscured by the skull's strong reflections. In addition, the high frequencies necessary for high-resolution imaging exacerbate scattering and anelastic losses within both the skull and brain tissue, limiting the signal quality of intracranial soft tissues (Guasch et al., 2020). In this context, conventional ultrasound methods cannot, and have not, been able to image intra-cranial structures of the adult human brain (Calderon Agudo, 2018). FWI, however, by leveraging the full-waveform signal, has the capacity to incorporate both reflected and transmitted waves, allowing for a more comprehensive capture of wave propagation through the skull. While transmitted signals remain weak and easily masked by the skull's strong reflections, the inclusion of these signals offers a critical advantage, presenting a chance, despite a challenging one, to obtain an ultrasound image of the brain (Guasch et al., 2020; Calderon Agudo, 2018).

Achieving this goal is not straightforward, as many of the same factors that

hinder conventional ultrasound also complicate FWI. Research by Calderon Agudo (2018) demonstrates that successful neuroimaging with FWI is theoretically possible only under strict conditions: both transmitted and reflected ultrasound data are recorded, i.e. long-offset acquisitions, 3D effects are incorporated, accounting for the full complexity of wave propagation in space, appropriate sources are used to obtain high-quality low-frequency signals, typically below 1 MHz, and accurate prior information on the skull's properties, i.e a good starting model, can be obtained. The work of Guasch et al. (2020) further corroborates the feasibility of FWI for brain imaging with synthetic examples, showing that under such conditions, MRI-like images of the brain can be obtained via ultrasound modelling. Given these requirements, the clinical feasibility of FWI for brain imaging hinges on two critical challenges: the computational burden of modelling high-frequency waveforms in 3D and the need for a reliable starting model for the skull. While advances in computational hardware and PDE modelling strategies could eventually mitigate the time and resource demands over the years, the dependency on a robust prior remains an obstacle for practical applications.

2.2.2 Full-waveform Inversion with Neural Networks

A close examination of algorithms 1 and 6, show that FWI is not far removed from the concept of a neural network. They are, after all, data-driven local optimisation algorithms aimed at resolving an exceedingly non-linear parameter space, often involving billions of degrees of freedom. In fact, Richardson (2018) reframed FWI within a deep learning framework by structuring it as a recurrent neural network (RNN), a type of autoregressive model designed for sequential processing. In an RNN, each unit corresponds to a sequential processing step, where states are passed from one layer to the next, allowing information from previous time steps to influence current outputs. Analogously in FWI, the wave equation is solved iteratively over time steps, with each step depending on prior states of the wavefield. By treating each layer of the RNN as a finite difference time step, the neural network forward pass effectively becomes an FWI forward solver, with its trainable parameters corresponding to those of the physical parameters of the FWI model. In this sense, the terminology

of *models* in FWI and *models* in deep learning are, in fact, closely aligned. This approach allows FWI to benefit from deep learning tools like automatic differentiation for gradient calculations and advanced optimisation algorithms, such as Adam (Kingma and Ba, 2014), to both accelerate and stabilise convergence. Similar work has been carried by Sun et al. (2020).

Numerous studies have explored deep-learning solutions to FWI challenges, addressing specific tasks including data denoising (Richardson and Feller, 2019), regularisation and prior generation for the inversion workflow (Wang et al., 2023; Zhu et al., 2022; Yao and Wang, 2022; Yao et al., 2023b; Siahkoohi et al., 2022; Li et al., 2021; Vantassel et al., 2022; Mosser et al., 2020; Lewis and Vigh, 2017; Strong, 2023; He and Wang, 2021; Wu and McMechan, 2019), acceleration of forward modelling (Moseley et al., 2020; Giannakis et al., 2019; Rasht-Behesht et al., 2022), acoustic-to-elastic mode conversions for data and FWI gradients (Li et al., 2019; Yao et al., 2020, 2023a), low-frequency data extrapolation (Sun and Demanet, 2020, 2021; Hu et al., 2021; Robins et al., 2021), model uncertainty quantification (Siahkoohi et al., 2020; Yin et al., 2024a), data augmentation (Yang et al., 2022), and more. The majority of these advancements have been developed within the domain of geophysics, where these deep neural networks address the main challenges in the inversion process summarised as: i) difficulties with algorithm conversions in real-world datasets and unreliable solutions due to FWI's inherent ill-posedness; ii) the need for better starting models, prior-based regularisers, and frequency-extrapolation techniques to mitigate cycle-skipping effects; and iii) the exhaustive computational demands of accurate wave modelling. Fortunately or not, these issues are also present in medical imaging, allowing solutions developed in geophysics to be directly applied to ultrasound imaging.

Of particular interest, a few studies have attempted to replace the FWI algorithm partially or entirely with deep learning models, a concept that closely aligns with the ambition of this thesis. There is a natural inclination to do so, given the conceptual parallels between FWI and neural networks, and the fast rise of processing capabilities of deep learning models. The following discussion provides an overview of some of the most influential studies in this topic, acknowledging that this review

is representative rather than exhaustive. It reflects the dynamic, experimental phase of deep learning in FWI, aiming to highlight key accomplishments and opportunities for development.

Araya-Polo et al. (2018) proposed a deep learning approach for velocity model building to bypass FWI utilising a deep neural network trained on synthetic 2D data to perform data-to-model inference. This approach leverages semblance panels that capture coherent energy across seismic traces to reduce dimensionality, while emphasising key velocity information. Tested on synthetic data, it demonstrated high structural similarity, with SSIM values averaging 0.8717 for K-Means segmented models and 0.8939 for continuous layered models. The DNN struggled with salt structures, where SSIM dropped to 0.8101, indicating limitations in handling high-contrast boundaries. Despite efficient inference, the authors highlight that generalisation to real seismic environments remains challenging due to reliance on synthetic data and simplified acoustic assumptions, which may not fully represent real geological variability.

Yang and Ma (2019) introduced a fully convolutional network tailored to multi-shot seismic data, differing from Araya-Polo et al. (2018) by training directly on raw seismic waveforms rather than semblance-derived features. Using a U-Net-based architecture with multiple input channels, their approach improved robustness to noise and the absence of low frequencies, achieving PSNR and SSIM scores on synthetic data that approached traditional FWI outputs. In comparison, while each FWI model required 37 minutes to compute, the FCN achieved inference within seconds after training, suggesting a promising application for real-time seismic inversion.The authors also highlight the network's reliance on synthetic data limits its adaptability to diverse geological structures.

Araya-Polo et al. (2020) proposed a 3D CNN to accelerate and automate seismic tomography for velocity model building. Initially trained on semblance-based features, like in their previous work (Araya-Polo et al., 2018), and later extended to raw data, this CNN achieved SSIM scores of 0.919 for layer-cake models and 0.892 for models inclusive of salt. The model encountered difficulties with sharp boundaries, particularly near salt structures, where its R^2 score was 0.741. The rapid inference

time offers significant workflow advantages, though the CNN's geological adaptability is constrained by its need for retraining across different regions. Additionally, its limited high-frequency detail resolution renders it a complementary, rather than standalone, solution to conventional FWI.

Wu and Lin (2019) developed InversionNet, an encoder-decoder CNN integrated with a conditional random field module for FWI, which enforces structural continuity within the generated velocity models. Bypassing traditional iterative inversion, InversionNet enables direct, end-to-end learning from seismic data to velocity model outputs. Tested on synthetic datasets, the model significantly reduces errors compared to traditional FWI techniques, particularly excelling in geometrically complex formations. While the conditional random field aids spatial continuity in complex regions, the authors also emphasise that InversionNet's reliance on synthetic data poses limitations for adaptation to field data with real-world noise and geological variability.

Mosser et al. (2018) instead approached FWI using a cycle-consistent GAN, re-framing seismic inversion as a domain transfer problem. Their GAN achieved rapid structural fidelity on synthetic geological models, efficiently managing complex structures like folding, faulting, and dyke intrusions. While promising, the GAN's training is computationally intensive, raising scalability concerns. Tested on field data, the model encountered artifacts and struggled with low-contrast reflectors, highlighting limitations in handling real-world noise and structural variability.

Zhang et al. (2020) introduced AD-DLFWI, a hybrid FWI method that enhances velocity models through adjoint-driven gradient refinement using an FCN. AD-DLFWI demonstrated superior fine-scale detail retention over conventional FWI because it uses the neural network to estimate and apply improved gradient updates in a physically meaningful way. This method, unlike InversionNet, retains physical interpretability by refining FWI's gradients, achieving SSIM values of 0.92 and PSNRs of 25 dB on synthetic data.

Ren et al. (2024) proposed PEN-UBT, a physics-embedded neural network for ultrasound brain imaging, aiming to reduce reliance on traditional FWI. PEN-UBT integrates a convolutional network for forward wavefield propagation with an inver-

sion subnetwork to reconstruct tissue structure, both trained concurrently. Achieving SSIM values above 0.95 on synthetic brain models, PEN-UBT outperforms traditional FWI (SSIM around 0.81), highlighting the advantages of embedding physics into the network. However, limitations of synthetic data and dependence on a suitable initial model persist, and the authors suggest transfer learning may be necessary for clinical adaptability.

Orozco et al. (2023) introduced an ultrasound imaging method using amortised normalising flows, bypassing traditional inversion through efficient, probabilistic sampling. Using a physics-informed summary statistic as input in the form of the first gradient of an FWI run, the model compresses high-dimensional data by a factor of 70. The probabilistic framework is also designed to support Bayesian uncertainty quantification, informing on the model's reliability. Achieving PSNR of 38.67 and SSIM of 0.9646 on synthetic data, it outperformed both FWI and U-Net benchmarks. Unlike AD-DLFWI or PEN-UBT, this method emphasises data efficiency along with robust uncertainty estimates, though the authors propose that real-world adaptation may require diverse anatomical training data.

Yang and Ma (2023) introduced FWIGAN, a GAN-based FWI method incorporating the acoustic wave equation directly within its framework. This structure allows for velocity model optimisation without traditional gradient updates, instead using adversarial learning for efficient convergence. FWIGAN achieved an SSIM of 0.6802 and SNR of 17.29 on the Marmousi model, outperforming both traditional FWI and deep image prior methods. While FWIGAN is theoretically robust against noise and poor initial models, its adaptability to field data may still be limited by sensitivity to data variability.

Saad et al. (2024) proposed SiameseFWI, a FWI method using a Siamese network to align observed and synthetic seismic data in a shared feature space for loss quantification, enhancing inversion stability. Tested on synthetic and field data, it achieved a 40% improvement over traditional FWI in complex geological data (SSIM = 0.65, SNR = 24 dB). However, the model's memory demands limit scalability, and it remains sensitive to data quality, particularly in real-world scenarios where noise and variability challenge the robustness observed in synthetic settings.

Across these studies, common challenges persist around generalisability, data reliance, and computational demands. Training deep learning models remains resource-intensive, often requiring extensive data volumes and high-performance computation, and consequently restricting scalability in real-world applications. Not only that, despite innovations like feature-based regularisation and physics-informed structures, deep-learning-based FWI methods still exhibit dependency on starting models, which can limit their efficacy when prior information is unavailable or unreliable.

Generalisability remains another key challenge, as synthetic training environments typically lack the noise and complexity found in real-world data. In contrast, medical imaging applications benefit from more standardised data, as anatomical structures are comparatively uniform, potentially easing the path to model generalisation in deep-learning-based ultrasound brain imaging. This consistency, however, remains underexplored and may represent an avenue for future research in intelligent UT methods.

Chapter 3

Convolutional Data Comparison with Wiener Filters

Data comparison is a fundamental building block of optimisation problems aimed at learning data distributions, whether through similarity metrics, loss functions, distance measures, or other similar objectives. Mean squared error (MSE) is a commonly employed data comparison strategy due to its simplicity, computational efficiency, and advantageous mathematical properties, such as smoothness and differentiability (Wang and Bovik, 2009). However, MSE-based reconstructions often lack quantitative accuracy and perceptual fidelity required for high-resolution synthesis tasks. This is because MSE is an element-wise operation that assumes all data points are independent and equally important, neglecting contextual correlations (Pappas et al., 2000; Wang and Bovik, 2009, 2006; Wu and Rao, 2017; Girod, 1993). In machine learning, this issue is exacerbated by neural networks' biases towards low-frequency content, bound by spectral biases and the F-principle, leading to high-frequency degradation and limited structural recovery (Rahaman et al., 2019; Xu et al., 2019; Wang and Bovik, 2009).

The exploration of better data-driven criteria has been relatively neglected in the machine learning community, which tends to favour increasingly complex network architectures to address low-fidelity issues (Kim et al., 2016; Child, 2020). Alternative metrics have been proposed but often suffer from training instabilities or require

costly computational operations. The Structural Similarity Index Measure (SSIM) evaluates similarity based on luminance, contrast, and structural information but can result in unstable training outcomes (Wang et al., 2004; Zhao et al., 2016; Pambrun and Noumeir, 2015); perceptual losses, such as LPIPS, compare data representations within the feature maps of pre-trained convolutional neural networks, and have become well-established due to their ability to learn hierarchical representations that are sensitive to correlations within data. However, they can be computationally expensive and potentially biased by the network architecture and the distribution of the training data (Johnson et al., 2016; Zhang et al., 2018a,b); adversarial losses rely on a trainable discriminator network as a loss function, but not only they introduce additional trainable weights, they are also prone to training instabilities due to their adversarial nature. (Goodfellow et al., 2014); Focal Frequency Loss (FFL) proposes a Fourier-domain distance method to mitigate high-frequency degradation, though further research is needed to validate its efficacy (Jiang et al., 2021); the Wasserstein distance from optimal transport theory has also been leveraged as a loss function, successfully improving multi-label classificaiton (Frogner et al., 2015) and non-parametric generative modelling (Liutkus et al., 2019) problems. Yet, its formulation remains computationally intensive and limited to non-negative, normalised signals (Peyré et al., 2019; Thorpe et al., 2017). In-depth discussion of these metrics can be found in Section 2.1.2.

In this research chapter, I introduce a novel data comparison strategy using Wiener filters, which leverages convolutional analysis to compare data samples without assuming local element-wise relationships. Specifically, I propose a new loss function based on Wiener filter analysis, named Wiener Loss, designed to optimise problems requiring comparison of data samples of the same domain. This work was presented at the NeurIPS 2023 Medical Imaging workshop (Cruz et al., 2023b), released in full on arXiv (Cruz et al., 2023a), and is briefly introduced in the thesis work of my co-author Dr. George Strong (Strong, 2023). This chapter follows closely to those publications in terms of writing style, structure, and visual elements, while incorporating additional experiments and discussions that are presented exclusively here.

As will be shown in the following sections, the proposed Wiener Loss enhances the fidelity of recovered samples by promoting full-spectrum recovery through contextual awareness. The method is formulated within a least-squares framework, resulting in a smooth solution space, and is therefore, cost-effective and competitive to other existing metrics.

The efficacy of the Wiener Loss is demonstrated here in two deep-learning applications: data compression and medical imputation. In the medical imputation example, I focus specifically on brain scans, where the contributions of the Wiener Loss are particularly beneficial due to the complex, irregular, and unique structures of the brain, such as cranial and skin formations, cortical folds, and ventricles. In these experiments, networks trained with the Wiener Loss can better capture data distributions, boost perceptual quality, and generate realistic data samples from tight latent spaces at reduced costs. Although this new formulation is implemented in deep-learning tasks, it is worth noting that this methodology can be used for any task where data comparison is required and preservation of data correlation is desired.

The direct inspiration for this work comes from Adaptive Waveform Inversion (AWI) (Warner and Guasch, 2016), which uses a loss function based on convolutional data-matching Wiener filters for seismic imaging. AWI is a refined variant of Full Waveform Inversion (FWI) (see Section 2.2), developed to address convergence challenges in FWI, particularly when fitting the non-linearities of seismic data in the presence of large phase differences between modelled and observed signals. In this context, AWI is formulated within a convolutional phase-matching framework, which I leverage here to promote perceptuality. Mathematically and conceptually, AWI is the technique most analogous to the proposed Wiener Loss.

3.1 Wiener Loss Formulation

Wiener-filter theory (Wiener, 1942) is central to the technique presented in this work. A Wiener filter can be defined as the convolutional filter that provides the best least-squares match between two signals. Mathematically, the Wiener filter that matches

a signal $\mathbf{y}$ to a signal $\mathbf{x}$, expressed as column vectors, is obtained by minimising the following expression with respect to $\mathbf{v}$:

$$\min_{\mathbf{v}} \frac{1}{2}||\mathbf{Y}\mathbf{v} - \mathbf{x}||^2, \tag{3.1}$$

where $\mathbf{v}$ is the Wiener filter, and $\mathbf{Y}$ is a matrix that represents convolution with signal $\mathbf{y}$. If signals $\mathbf{x}$ and $\mathbf{y}$ are one-dimensional, then $\mathbf{Y}$ will take the form of a Toeplitz matrix, with progressively lagged values of the signal $\mathbf{y}$ placed in each column (Hansen, 2002). If signals $\mathbf{x}$ and $\mathbf{y}$ are matrices representing two-dimensional images, then $\mathbf{Y}$ is given by a doubly block Toeplitz matrix (Hansen, 2002). The solution to this linear least-squares problem is given by the normal equation, obtained by setting the derivative of Equation 3.1 with respect to $\mathbf{v}$ to zero:

$$\mathbf{v}(\mathbf{x}, \mathbf{y}) = (\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{x}, \tag{3.2}$$

where $\mathbf{Y}^T$ is the transpose of the Toeplitz matrix $\mathbf{Y}$, that represents the cross-correlation with signal $\mathbf{y}$ (Hansen, 2002). In full, this equation can be interpreted as the cross-correlation of $\mathbf{y}$ with $\mathbf{x}$, deconvolved by the autocorrelation of $\mathbf{y}$.

In this work, full-lag Wiener filters that are of at least the same size as signals $\mathbf{x}$ and $\mathbf{y}$ are implemented. The receptive field of any element within signal $\mathbf{x}$ is therefore equivalent to the entirety of signal $\mathbf{y}$ and vice versa. This means that Wiener filter $\mathbf{v}(\mathbf{x}, \mathbf{y})$ globally matches signal $\mathbf{y}$ to signal $\mathbf{x}$ through convolution, without assuming any local relationships between any of the individual elements or structures contained within signals $\mathbf{x}$ and $\mathbf{y}$. This sensitivity to the global content within the samples is a core advantage of using Wiener filters for data comparison over conventional methods sensitive only to local element-wise differences. It is worth noting that truncated Wiener filters could of course be used in applications where it may be advantageous to limit the receptive field in the data matching process, though that topic is left for exploration in further research.

If signals $\mathbf{x}$ and $\mathbf{y}$ are identical, the resulting Wiener filter that best matches them is given by the Dirac delta function, i.e. the identity of the convolution operation. Equivalently, the closer the Wiener filter coefficients are to a Dirac delta

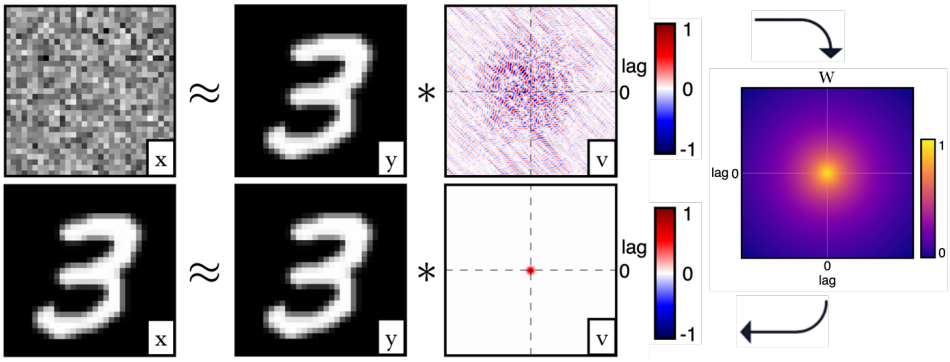


Figure 3.1: Top: visual illustration of two-dimensional Wiener filters that match MNIST sample $\mathbf{y}$ to random noise $\mathbf{x}$. Bottom: matching $\mathbf{y}$ to $\mathbf{x}$ where $\mathbf{x} = \mathbf{y}$, demonstrating that the Wiener filter when $\mathbf{x} = \mathbf{y}$ is given by the Dirac delta function. During Wiener Loss optimisation, the whitening matrix $\mathbf{W}$ iteratively penalises the residuals between the Wiener filter and zero-lag spike, driving it towards a Direct delta function, and $\mathbf{x}$ and $\mathbf{y}$ towards identity.

function, the more similar the sample pairs are (Figure 3.1). The formulation of the Wiener Loss leverages this notion of similarity by explicitly driving the matching Wiener filter between $\mathbf{x}$ and $\mathbf{y}$ towards the convolutional identity. This process, in turn, implicitly steers signals $\mathbf{x}$ and $\mathbf{y}$ also towards the identity. This is the central idea behind the Wiener Loss.

One formulation that can naturally achieve this is the minimisation of the Rayleigh quotient of the Wiener filter between $\mathbf{x}$ and $\mathbf{y}$ with respect with a penalisation matrix $\mathbf{T}$. The quotient $\mathcal{R}$ is defined as:

$$\mathcal{R}(\mathbf{T}, \mathbf{v}(\mathbf{x}_\theta, \mathbf{y})) = \frac{||\mathbf{T}\mathbf{v}(\mathbf{x}_\theta, \mathbf{y})||^2}{||\mathbf{v}(\mathbf{x}_\theta, \mathbf{y})||^2}. \tag{3.3}$$

where, $\mathbf{y}$ is a vector containing the observed or target data, $\mathbf{x}_\theta$ is the predicted data from a model parameterised by θ, $||\cdot||^2$ is the squared Euclidean norm, and $\mathbf{T}$ is an arbitrary *monotonically decreasing* function centred at zero-lag that penalises non-zero-lag energy in the filter, driving it towards a zero-lag spike. The penalisation is enforced through the Hadamard product between $\mathbf{T}$ and $\mathbf{v}(\mathbf{x}, \mathbf{y})$. The normalisation term in the denominator of Equation 3.3 is crucial to prevent collapses towards trivial solutions. Without it, the numerator is minimised by simply suppressing all of the values in the filter. The minimisation of $\mathcal{R}(\mathbf{T}, \mathbf{v}(\mathbf{x}_\theta, \mathbf{y}))$ is the objective that defines AWI.

One important drawback of Equation 3.3 is that it renders itself insensitive to amplitude variations within the data. The Rayleigh quotient is known to be minimised when $\mathbf{v}$ is the eigenvector $\mathbf{v}_{min}$ of $\mathbf{T}$ associated with the smallest eigenvalue (Johnson and Horn, 1985). Because any constant scaling of an eigenvector is also a solution of its eigenspace (defined as the set of all the eigenvectors corresponding to the same eigenvalue), there exist multiple scaled solutions of $\mathbf{v}$, and therefore of $\mathbf{x}$, that satisfy the minimisation condition. Minimising Equation 3.3, therefore, only requires the filter energy to be focused at zero-lag, being indifferent to its amplitude. The formulation also has no constraints over the amplitudes of individual elements of the filter and, at worst, minimising Equation 3.3 can result in a negative Dirac delta function approximation with mixed positive and negative values, leading to erroneous polarities in the recovered data.

The amplitude indifference of Equation 3.3 is of little concern in seismic imaging because signal normalisation is a fundamental part of its heuristic pipelines. The primary focus is on the phase matching of signals. However, in machine learning, both amplitude and phase are crucial for most applications (Jiang et al., 2021; Goodfellow et al., 2014; Johnson et al., 2016; Zhang et al., 2018a,b). To address this limitation, the proposed Wiener Loss in this work does not directly penalise the filter, but instead penalises the residuals between the filter and a Dirac delta, defined as:

$$\mathcal{L}_{\mathrm{w}}(\mathbf{x}_\theta, \mathbf{y}) = \frac{1}{2}||\mathbf{W}\left\{\mathbf{v}(\mathbf{x}_\theta, \mathbf{y}) - \boldsymbol{\delta}\right\}||^2, \tag{3.4}$$

where $\boldsymbol{\delta}$ is the multi-dimensional Dirac delta function, $\mathbf{y}$ is a vector containing the observed or target data, $\mathbf{x}_\theta$ is the predicted data from a model parameterised by θ, and the penalisation matrix, here represented as $\mathbf{W}$ instead of $\mathbf{T}$, is an arbitrary *monotonically increasing* function centred at zero-lag that penalises *energy residuals* of the filter, with emphasis at zero-lag. The penalisation is, therefore, higher for residuals at zero-lag, preventing any negative polarities and non-unitary values (Figure 3.1). Finally, due to its least-squares nature, the Wiener Loss retains desirable optimisation properties inherent to MSE of smoothness, convexity and differentiability.

3.1.1 Fourier-Domain Implementation

An efficient method for solving Toeplitz systems and calculating one-dimensional Wiener filters from Equation 3.2 is the Levinson-Durbin recursion (LDR) algorithm (Levinson, 1946; Durbin, 1960). Although extensions of this algorithm for block Toeplitz systems can be used to calculate higher dimensional-Wiener filters (Musicus, 1988), such implementations become computationally intractable with increased sample sizes. For greater computational efficiency, Equation 3.2 is implemented in the Fourier-domain as follows:

$$\mathbf{v}(\mathbf{x},\mathbf{y}) = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\{\mathbf{x}\}\otimes\mathcal{F}\{\mathbf{y}\}+\lambda}{\mathcal{F}\{\mathbf{x}\}\otimes\mathcal{F}\{\mathbf{x}\}+\lambda}\right\} \tag{3.5}$$

where $\mathcal{F}$ represents the Fast Fourier transform (FFT), $\otimes$ represent the Hadamard product (element-wise multiplication), the vinculum represents Hadamard division (element-wise division), and λ is a pre-whitening scalar used to stabilise the deconvolution.

In this way, the convolution operations inherent to the Wiener filter are converted into element-wise operations, eliminating the need for expensive matrix multiplications and inversions. Note that λ is a stabilising parameter added to both numerator and denominator in Equation 3.5 to ensure that the Wiener filter $\mathbf{v}(\mathbf{x},\mathbf{y})$ when $\mathbf{x}=\mathbf{y}$ is given by a Dirac delta function with an amplitude of 1.

3.1.2 Statistical Interpretation

The probability of a variable $\mathbf{z}\sim p(\mathbf{z})$ belonging to a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ is given by the well known density function:

$$p(\mathbf{z}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{k/2}|\boldsymbol{\Sigma}|^{1/2}}\exp\left(-\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{z}-\boldsymbol{\mu})\right), \tag{3.6}$$

where k is the dimensionality of the variable $\mathbf{z}$, and the exponential term is the square of the Mahalanobis distance between the point $\mathbf{z}$ and the distribution $\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$ (McLachlan, 1999). In the formulation of the Mahalanobis distance, $\boldsymbol{\Sigma}^{-1}$ is decomposed to $\boldsymbol{\Sigma}^{-1}=\mathbf{W}^T\mathbf{W}$ by the spectral theorem, where $\mathbf{W}$ is the *whitening matrix*

that scales and decorrelates data, effectively converting the covariance matrix into an identity matrix. In other words, the whitening matrix performs a decorrelation transformation on the point $\mathbf{z}$ such that the distance between the point and the defined distribution is equivalent to the distance between the point after transformation and a multivariate Gaussian whose covariance matrix is the identity matrix. Although the existence of $\mathbf{W}$ is guaranteed, its existence is not unique (Kessy et al., 2018).

Gaussian modelling over $p(\mathbf{z})$ is achieved by parameterising $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, such that the logarithm of $p(\mathbf{z} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is maximised, a process known as Maximum Likelihood Estimation (MLE). This is formulated as a minimisation problem of the negative log-likelihood of Equation 3.6, where the constants are omitted and the optimisation is written as a proportion:

$$\min_{\boldsymbol{\mu},\boldsymbol{\Sigma}} -\log p(\mathbf{z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \propto \frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{z}-\boldsymbol{\mu}). \tag{3.7}$$

The AWI objective in Equation 3.3 can be expanded to take the form of the Mahalanobis distance:

$$\mathcal{R}(\mathbf{T}, \mathbf{v}(\mathbf{x}_\theta, \mathbf{y})) = \frac{1}{2}\left(\frac{\mathbf{v}}{\sqrt{\mathbf{v}^T\mathbf{v}}}\right)^T \mathbf{T}^T\mathbf{T}\left(\frac{\mathbf{v}}{\sqrt{\mathbf{v}^T\mathbf{v}}}\right). \tag{3.8}$$

Combined with Equation 3.7, the minimisation of Equation 3.8 over the network parameters then takes the form:

$$\min_{\boldsymbol{\theta}} -\log p(\mathbf{v}|\boldsymbol{\mu} = \vec{0}, \boldsymbol{\Sigma}) \propto \frac{1}{2}\left(\frac{\mathbf{v}}{\sqrt{\mathbf{v}^T\mathbf{v}}}\right)^T \mathbf{T}^T\mathbf{T}\left(\frac{\mathbf{v}}{\sqrt{\mathbf{v}^T\mathbf{v}}}\right), \tag{3.9}$$

which is interpreted as maximising the probability of a normalised Wiener filter $\mathbf{v} = \mathbf{v}(\mathbf{x}_\theta, \mathbf{y})$ belonging to a multivariate Gaussian with covariance matrix $\boldsymbol{\Sigma} = (\mathbf{T}^T\mathbf{T})^{-1}$ and mean zero.

Conversely, the proposed Wiener Loss formulation in Equation 3.4 can be expanded to take the following form of the Mahalanobis distance:

$$\mathcal{L}_{\mathrm{w}}(\mathbf{x}_\theta, \mathbf{y}) = \frac{1}{2}\Big(\mathbf{v}-\boldsymbol{\delta}\Big)^T\boldsymbol{\Sigma}^{-1}\Big(\mathbf{v}-\boldsymbol{\delta}\Big), \tag{3.10}$$

and minimisation of Equation 3.10 over the network parameters takes the form:

$$\min_{\boldsymbol{\theta}} -\log p(\mathbf{v}|\boldsymbol{\delta}, \boldsymbol{\Sigma}) \propto \frac{1}{2}(\mathbf{v}-\boldsymbol{\delta})^T\boldsymbol{\Sigma}^{-1}(\mathbf{v}-\boldsymbol{\delta}), \tag{3.11}$$

which can be interpreted as maximising the probability of a Wiener filter $\mathbf{v} = \mathbf{v}(\mathbf{x}_\theta, \mathbf{y})$ belonging to a multivariate Gaussian with covariance matrix $\boldsymbol{\Sigma} = (\mathbf{W}^T\mathbf{W})^{-1}$ and mean defined as the Direct delta. Equation 3.11 can be additionally interpreted as maximising the likelihood of the residuals between a Wiener filter and a Direct delta belonging to a multivariate Gaussian with covariance matrix $(\mathbf{W}^T\mathbf{W})^{-1}$ and mean zero. Both interpretations are equally valid.

The primary distinction between the two methods, therefore, lies in the assumption of normality: the AWI objective assumes normality over the Wiener filters, while the proposed Wiener Loss assumes normality over the residuals between the filters and the Dirac delta. Many statistical models, such as linear regression, commonly assume normality over residuals, making this approach a well-understood and stable practice in the field (Neter et al., 1996; Goodfellow et al., 2016; Bishop and Nasrabadi, 2006; Ikotun et al., 2023; Sayed, 2011; Breiman, 2001). Although delta-like filters might satisfy this assumption, it is generally more straightforward to fit the residuals than the Wiener filters because unlike the filters, the residuals can be safely assumed to be decorrelated. This assumption simplifies the design of penalty functions, and even using the identity matrix as covariance can be effective.

3.1.3 Optimisation Landscape

A notable issue with the AWI objective is that the optimisation landscape is solely defined by the choice of $\mathbf{T}$. Narrow forms of $\mathbf{T}$ are required to suppress filter energy across the entire range from far to near-zero offsets, leading to two major problems. Firstly, the narrow landscape results in steep gradients near the minimum, causing larger gradients near convergence (Figure 3.2), which contradicts the design principles of modern deep-learning optimisers. Secondly, far from the minimum, the gradients are almost flat, slowing down convergence and making it harder to achieve, heavily relying on initialisation. A visual representation of the optimisation

2939 landscape for each loss is shown in Figure 3.3.

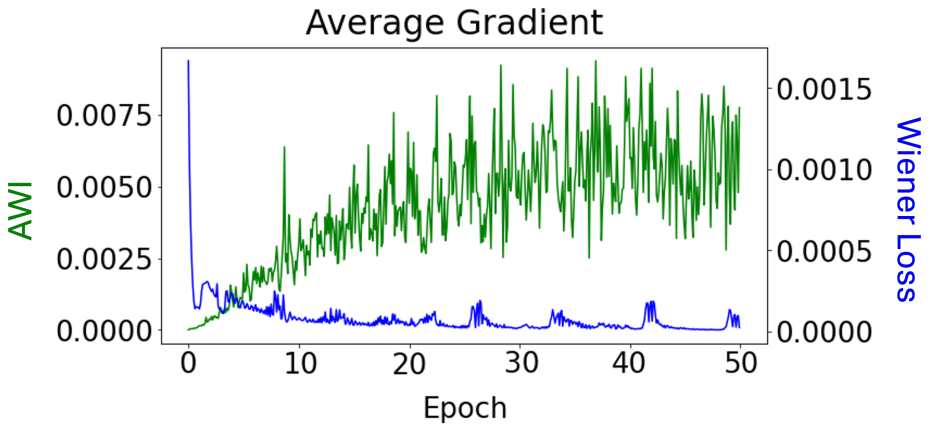


Figure 3.2: Average gradient of 2D-AWI and 2D-Wiener Loss objectives for a Variational Autoencoder (VAE) model successfully trained with both methods. The model gradients under the AWI objective increase at near the minimum, while model gradients under the Wiener Loss decrease, making the latter more befitting to most standard deep learning optimisers.

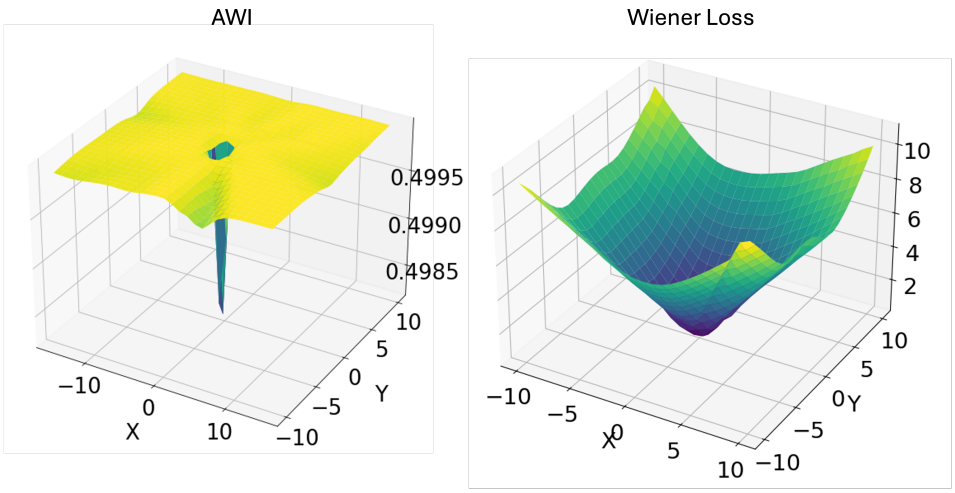


Figure 3.3: Comparison of and estimation of the loss landscapes of a VAE model successfully trained to reconstruct MNIST digits (LeCun et al., 1998) with 2D-AWI (left) and 2D-Wiener Loss (right). For convergence, the 2D-model required a Delta-like penalty function ($\mathbf{T} \approx \boldsymbol{\delta}$), whereas the Wiener Loss achieved superior results with $\mathbf{W} = \boldsymbol{I}$. The steep gradients required in AWI create challenging conditions for optimisation, while the loss landscape of the model trained with 2D-Wiener Loss resembles the smooth surface of the MSE. The visualisation, achieved with the method described in Li et al. (2018), represents a two-dimensional parametrisation of a the optimised multi-million parameter model.

In contrast, the Wiener Loss, which minimises over the residuals, presents a much smoother minimum and offers greater flexibility with the choice of penalties. This approach effectively mitigates the two major problems associated with the AWI's objective landscape. By focusing on residuals, the training benefits from more stable and appropriate gradient behaviour, leading to more efficient and reliable convergence during optimisation.

3.2 Results and Discussion

In this section, I present the framework of the Wiener Loss and conduct three reconstruction experiments, benchmarking its performance against MSE, FFL, LPIPS, and adversarial losses. I also provide a critical analysis of the Wiener Loss in these contexts to identify its limitations and propose directions for future research that could address and overcome these shortcomings.

3.2.1 Framework and Implementation

The Wiener Loss is available at /https://github.com/dpelacani/WienerLoss, a public GitHub repository, and is implemented as a single Python class derived from PyTorch's (Paszke et al., 2019) `nn.Module` class. This implementation was chosen to maximise compatibility with PyTorch's other loss classes, ensuring that minimal code modification is needed to integrate the Wiener Loss into existing training workflows. The class includes basic default parameters for straightforward utilisation but remains highly customizable to suit specific tasks:

```
class WienerLoss(nn.Module):
    """
    Args:
        method, optional (str)
            "fft" for Fast Fourier Transform or "direct" for the
            Levinson-Durbin recursion algorithm. Defaults to "fft"
        filter_dim, optional (int)
            the dimensionality of the filter. This parameter should be
```

```
        upper-bounded by the dimensionality of the data. If data is
        3-dimensional and filter_dim is set to 2, one filter is
        ↪ computed
        per channel dimension assuming format [B, NC, H , W]. Current
        implementation only supports filter dimensions for 1D, 2D and
        ↪ 3D.
        Defaults to 2
    filter_scale, optional (int)
        the scale of the filters compared to the size of the data.
        Defaults to 2
    reduction, optional (str)
        specifies the reduction to apply to the output, "mean" or
        ↪ "sum".
        Defaults to mean
    mode, optional (str)
        "forward" or "reverse" computation of the filter. For details
        ↪ of
        the difference, refer to the original AWI paper. Default
        ↪ "reverse"
    penalty_function, optional (func)
        the penalty function to apply to the filter. If None, the
        ↪ penalty
        function is the identity. Must take as argment a single
        ↪ multi-dimensional
        torch mesh. Default None
    store_filters, optional (str)
        whether to store the filters in memory, useful for debugging.
        Option to store the filers before or after normalisation with
        "norm" and "unorm". Default False.
    lmbda, optional (float)
        the stabilisation value to compute the filter. Default 1e-4.
    clamp_min, optional (float)
        filters are clipped to this minimum value after computation. If
        None, operation is disabled. Default None
"""
```

```
def __init__(self. **kwargs):
    super(WienerLoss, self).__init__()
    ...
    return


# Instantiation with optional kwargs
>>> wiener_loss = WienerLoss(**kwargs)

# Usage
>>> x = torch.randn((10, 3, 64, 64))
>>> y = torch.randn((10, 3, 64, 64))
>>> wiener_loss(x, y, gamma=1e-2., eta=0., lmbda=1e-4)
tensor(1.4861, grad_fn=<DivBackward0>)

>>> wiener_loss(x, x, gamma=0., eta=0., lmbda=0.)
tensor(0., grad_fn=<DivBackward0>)
```

The `method` argument defines the operational approach for the Wiener filter computation, allowing the user to choose between explicitly constructing the Toeplitz matrix to perform Levinson-Durbin recursion or computing the filters in the frequency domain using FFT (see Section 3.1). The Wiener filters are computed in the frequency domain with the following class method:

```
...
def wienerfft(self, x, y, filter_shape, lmbda=1e-9):
    """
    Calculates the optimal least squares convolutional Wiener filter
    ↪  that transforms signal x into signal y using FFT
    """
    assert x.shape == y.shape, "signals x and y must be the same shape"

    # Cross-correlation of x with y
    Fccorr = torch.fft.fftn(x, dim=self.dims)\
```

```
        * torch.conj(torch.fft.fftn(y, dim=self.dims))

    # Auto-correlation of x
    Facorr = torch.fft.fftn(x, dim=self.dims)\
        * torch.conj(torch.fft.fftn(x, dim=self.dims))

    # Deconvolution of Fccorr by Facorr with pre-whitening
    Fdconv = (Fccorr + lmbda) / (Facorr + lmbda)

    # Inverse Fourier transform
    rolled = torch.fft.irfftn(Fdconv, filter_shape[-self.filter_dim:],
    ↪  dim=self.dims)

    # Unrolling
    rolling = tuple([int(-x.shape[i]/2) - 1
                     for i in range(1, len(x.shape),
                     ↪  1)])[-len(self.dims):]
    return torch.roll(rolled, rolling, dims=self.dims)
```

The class initialisation also provides an option to pass a custom penalty function (or whitening matrix from 3.1), which defaults to the identity matrix for 2D and 3D inputs and the Direc delta function for 1D inputs. However, it is important to note that in the 2D and 3D cases, the form of the penalty function $\mathbf{W}$ is constructed differently from the whitening matrix defined through $\boldsymbol{\Sigma} = (\mathbf{W}^T\mathbf{W})^{-1}$ in Equation 3.11. The matrix $\mathbf{W}$ in this context is designed to perform a the Hadamard product on the filters that is equivalent to the matrix multiplication operation defined in Equation 3.11, more closely related to the diagram of Figure 3.1. For instance, when the whitening matrix is chosen to be $\mathbf{W} = \boldsymbol{I}$, the penalty function is a tensor of same size and dimension as the filters, populated entirely with ones. The matrix multiplication of the filters with the whitening matrix is equivalent to the Hadamard product of the filters with its penalty function counterpart. This substitution is made because the Hadamard product is significantly more computationally efficient than matrix multiplication. In general, the penalty function should be represented

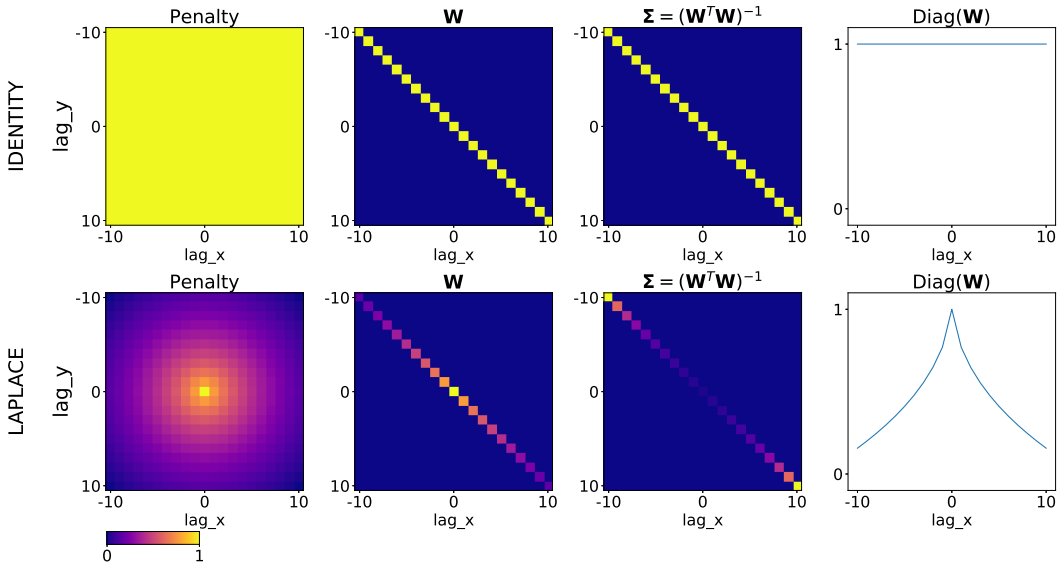


Figure 3.4: Identity (top) and Laplace (bottom) penalty functions along with their corresponding whitening matrices ($\mathbf{W}$) is illustrated. Both approaches assume decorrelation in the residuals, but show different self-correlations. The Identity function has unit variance and suppresses residuals uniformly, whereas the Laplace function imposes higher variance at far offsets, emphasising near-zero offset residual suppression.

as the spatially-symmetric form of the diagonal elements of the whitening matrix, see Figure 3.4.

In the `forward` method of the class, this takes the simple form:

```
...
def forward(self, recon, target, lmbda=None, gamma=0., eta=0.)
    # White noise to recon and target for stabilisation
    recon = recon + gamma * torch.rand_like(recon)
    target = target + gamma * torch.rand_like(target)
    ...
    # Evaluate penalty function
    penalty = self.make_penalty(
        penalty_function=self.penalty_function,
        eta=eta, **kwargs
    )
    ...
    # Option for dynamic stabilisation
    lmbda = self.lmbda if lmbda is None else lmbda
```

```
# Compute wiener filter
v = self.wienerfft(target, recon, filter_shape, lmbda)

# Normalise filter and store if prompted
self.filters = None
if self.store_filters == "unorm":
    self.filters = v[:]
elif self.store_filters == "norm":
    vnorm = torch.norm(v, p=2, dim=self.dims)
    self.filters = v[:] / vnorm

...
# Evaluate Loss for each sample
f = 0.5 * torch.norm(penalty * (v - delta), p=2, dim=self.dims)

# Reduce batch losses
f = f.sum()
if self.reduction == "mean":
    f = f / recon.size(0)
return f
```

An option to store the computed filters within the class is also provided. This is particularly useful for assessing the quality of training by inspecting the resulting Wiener filters. Access to the filters can be easily achieved through by accessing the `filters` attribute of the class via `wiener_loss.filters` when the `store` argument is passed.

The `filter_scale` parameter defines the receptive field of the filter relative to the size of the inputs and targets. For example, if the input size is (28, 28) and the filter scale is set to 2, the resulting Wiener filter will be of size (55, 55), noting that the filter size is always rounded to the closest odd integer for easier manipulation.

The `lmbda` parameter is the λ pre-whitening stabilisation value added to the FFT deconvolution, as per Equation 3.5. This value can be specified as a static value during class instantiation or dynamically modified during class calls.

The `gamma` and `eta` parameters control the level of white noise added to the input and target, and to the penalty function, respectively. Both parameters are dynamic and can be specified directly when calling the class, rather than being statically initialised attributes.

The `mode` parameter controls the mode of filter computation. Due to the commutative property of convolutions, the filter that transforms the input $\mathbf{x}$ into the target $\mathbf{y}$ is the same filter that transforms the target $\mathbf{y}$ into the input $\mathbf{x}$. Following the nomenclature of Warner and Guasch (2016), the former is described as the `forward` mode of operation and the latter as the `reverse` mode of operation. For the performed experiments, only the reverse mode is presented, with earlier experiments showing only marginal differences between the two.

Finally, the presented implementation is also capable of handling multiple dimensionalities through the argument `filter_dim`. The dimensionality of the filter can be considered another hyperparameter, although the class can only handle a filter dimensionality equal to or lower than that of the data (i.e., 3D data can be flattened for operation with 1D filters, but 1D filters cannot be reshaped for operation in 3D). 2D implementations operate with one filter per channel dimension, while 3D filters involve one filter per sample encompassing in it the channel dimension. It was empirically observed that for 2D data, 2D filters are more appropriate than 1D filters, as they can capture pixel correlation in both dimensions. For the described experiments, 3D filters did not significantly improve the results for multi-channel 2D data, although I stress that 3D implementations of the Wiener Loss for 3D data remain untested.

3.2.2 Reconstruction of MNIST

In this section, I conduct a preliminary study of the Wiener Loss by comparing the VAE reconstruction of the well-known MNIST (LeCun et al., 1998) dataset, which contains 60,000 single-channel 28x28 images of handwritten digits. At this stage, I compare only with MSE since later experiments will offer a more comprehensive comparison between the Wiener Loss and other competitive methods.

The immediate advantage of the Wiener Loss over MSE, as discussed in the

chapter introduction, is its ability to better recover the full range of frequencies in the data through a phase-matching approach and, by extension, its finer structures. This is demonstrated in Figure 3.5, which shows the comparison of one instance of each digit (0-9) recovered by network trained with MSE (top row) and the same network trained with the Wiener Loss (middle row). Figure 3.6 further supports this claim by comparing the frequency spectrum of each digit under both training regimes. In this comparison, the MSE (top row) shows limited recovery in the high-frequency regions, indicating its limited ability to capture fine details. The Wiener Loss reconstruction (middle row) appears more accurate, with less distortion and better preservation of high-frequency details, demonstrating its effectiveness in recovering finer structures compared the to ground truth (bottom row).

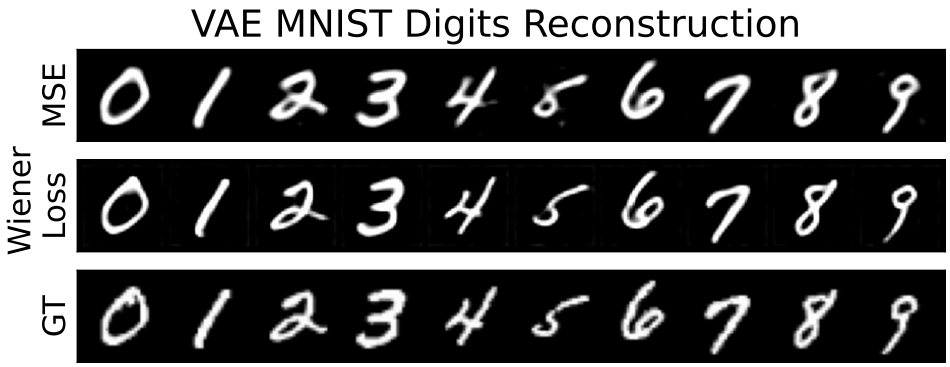


Figure 3.5: Reconstruction of MNIST digits using a variational autoencoder (VAE) from the validation batch. The top row represents the reconstruction using Mean Squared Error (MSE) loss, the middle row shows the reconstruction using Wiener Loss, and the bottom row displays the ground truth (GT) images. The implementation of the Wiener Loss demonstrates superior performance in recovering accurate and high-frequency details, as evidenced by the sharper edges of the digits compared to the MSE counterpart.

The VAE model used in this study is a U-Net (Ronneberger et al., 2015) model, consisting of a feature layer with 16 channels and 2 hidden layers of channel dimensions (32, 32), each containing 2 residual units and a group normalisation layer (Wu and He, 2018) of group size of 16 to improve generalisation. Each non-feature layer performs a downsampling operation, reducing the size of the image by a factor of 2. Consequently, the latent space of the model has a size of $\frac{28}{2^2} = 7$ and is arbitrarily chosen to be single-channel like the dataset images. Overall, this model contains

280,728 trainable parameters and was trained with Mish (Misra, 2019) as activation function and Adam (Kingma and Ba, 2014) as optimiser parameterised by $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The Kullback-Leibler Divergence (KL) divergence was applied with a low weight value of 1×10^{-4} to function as a simple latent regulariser. The data was split into training and validation 80/20 with equal class balance. For both losses, the model was trained with a batch size of 32, for 5 epochs, until convergence.

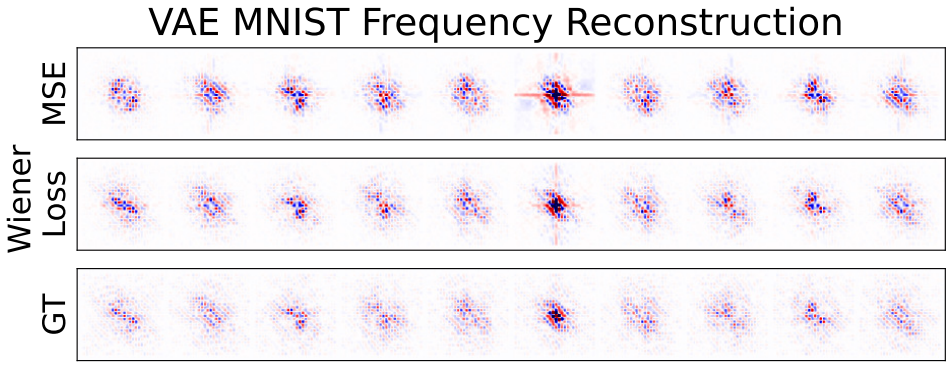


Figure 3.6: Frequency domain representations of the reconstructed MNIST digits in Figure 3.5. The top row displays the real Fast Fourier Transform (FFT) of the digits reconstructed using Mean Squared Error (MSE) loss, the middle row shows the real FFT using Wiener Loss, and the bottom row presents the real FFT of the ground truth (GT) images. The Wiener Loss captures a wider range of frequencies than the MSE loss, and is a more accurate reflection of the GT spectrum.

In the Wiener Loss training, the penalty function is a matrix of same size of the filter, populated entirely with ones, such that the whitening matrix is the identity and assumes uniform unit variance over the residuals (Figure 3.4). The Wiener filters were chosen to be twice the size of the image and were computed in the `reverse` mode with the FFT method. The results presented used a static pre-whitening value of $\lambda = 7$ and a static image noise level of $\gamma = 0.25$.

In Figure 3.7, I further illustrate these findings by comparing samples at different stages of training for both methods. At the start, the MSE-based model quickly produces recognisable but blurry digit reconstructions, stabilising early due to its focus on minimising pixel-wise differences. However, these reconstructions lack fine details and high-frequency components, even after convergence. The Wiener Loss model exhibits a different trajectory, focusing primarily on phase alignment. Early in the training, the reconstructions show significant amplitude mismatches because the

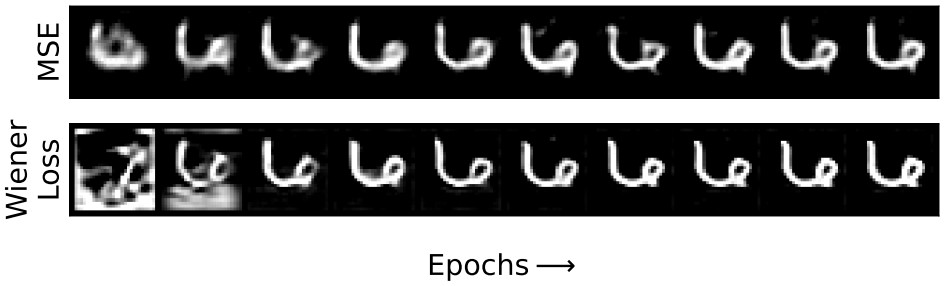


Figure 3.7: Progressive reconstructions of MNIST digits using MSE (top) and Wiener Loss (bottom) models, highlighting the network's initial biases towards low frequencies. Columns represent progressive epochs from left to right. The MSE-based model quickly produces recognisable but blurry digit reconstructions, lacking finer details and high-frequency components even at convergence. In contrast, the Wiener Loss model initially shows significant amplitude mismatches due to prioritising phase alignment but rapidly fits low-to-mid frequencies, gradually improving towards higher frequencies that the MSE fails to achieve.

model prioritises phase over individual pixel matching. This initial focus on phase alignment results in the Wiener Loss model quickly fitting low-to-mid frequencies and gradually catching up with the higher frequencies that MSE fails to achieve.

As training progresses, the Wiener Loss model fine-tunes high-frequency components and corrects amplitude mismatches, resulting in sharper and more detailed reconstructions. This phenomenon is also observed in Figure 3.8, where a batch of the Wiener filters is plotted across progressing epochs. As training evolves, the filters progressively become more spike-like at zero lag recovering the higher frequencies. Initially, the model focuses on suppressing non-zero lag energy to align the phases of the data at different frequencies, but as the training stabilises, it begins to address the amplitude of the spike and further suppresses these non-zero lag values. Overall, given sufficient training time, the Wiener Loss model is likely to account for amplitudes effectively after achieving good phase matches.

I note that the initial bias towards fitting low-frequency components before high-frequency components strongly resembles the F-Principle observed in neural networks (Xu et al., 2019). In Figure 3.7, this low-frequency prioritisation is also evident during MSE training, although there appears to be a frequency upper bound that MSE cannot surpass, unlike the Wiener Loss. It is plausible that the biases seen with the Wiener Loss are inherited from the network architecture rather than

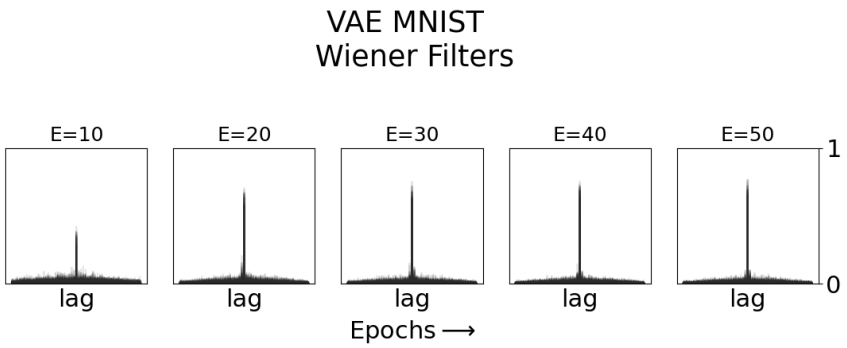


Figure 3.8: Evolution of the Wiener filters for a batch of 64 reconstructed MNIST digits (validation set) across different epochs. The filters have been flattened for better visualisation. Initially, the model focuses on suppressing non-zero lag energy to align the phases of the data at different frequencies. As the training stabilises, it begins to address the amplitude of the spike and further suppresses these non-zero lag energy values, demonstrating a refined convergence towards accurate frequency alignment and amplitude correction over time.

being intrinsic to the loss function itself. This remains an intriguing topic for further research.

Empirical observations showed that the convergence of the Wiener Loss is highly dependent on the choice of stabilisation values. Stabilisation values are particularly important when performing inversions of ill-conditioned matrices and in FFT deconvolution, as they prevent numerical instabilities (Golub and Van Loan, 1996). In the `WienerLoss` class implementation, stabilisation values are controlled through two main methods: the pre-whitening constant λ, which is added to both terms of the deconvolution as per Equation 3.5, and a constant noise level variable γ, which adds levels of Gaussian noise directly to the input and target. In the Fourier domain, both stabilisation methods aim to boost unrepresented frequencies in FFT deconvolution to avoid numerical instability in filter computation.

The implementation of γ is dynamic and asymmetrical, as the noise tensor added to the input differs from the one added to the target, and also changes with every `forward` call. This, unlike λ, results in varying stabilisation effects throughout the training process. Through experimentation, I found that balancing λ and γ is essential to stabilise and speed up the convergence of the Wiener Loss.

In Figure 3.9, I explored the behaviour of these two parameters through a simple

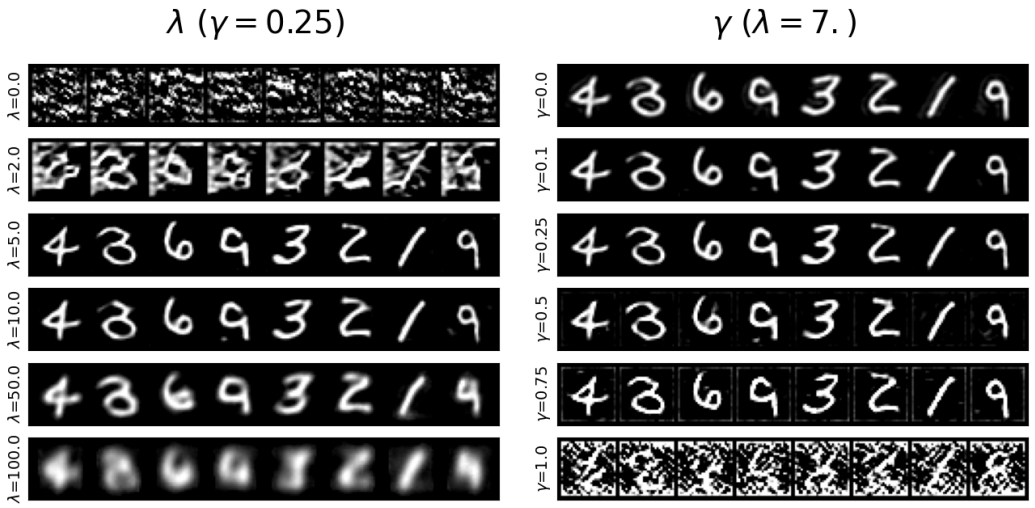


Figure 3.9: Effect of varying λ and γ on MNIST digit reconstructions using the Wiener Loss, demonstrated with samples from the validation dataset. The left panel illustrates the effect of varying λ while keeping γ fixed at 0.25, showing that low λ values hinder convergence due to numerical instabilities, while high λ values lead to blurry reconstructions. The right panel demonstrates the effect of varying γ with λ fixed at 7, indicating that low γ values slow convergence and reduce high-frequency data fitting, whereas high γ values disrupt convergence entirely.

experiment. Five models were trained with a fixed γ and varying λ values (left), and another five models were trained with a fixed λ and varying γ values, with all other parameters kept constant. Low values of λ can prevent model convergence due to numerical instabilities in the Hadamard division of Equation 3.5 or in high condition numbers in Equation 3.2. Low values of γ can result in slow convergences, leading to poor high-frequency data fitting. In contrast, excessively high λ values can result in blurry modelling of the data due to spectrum obfuscation. However, because the constant is applied equally to both terms, model convergence is still possible as the resulting Wiener filters can achieve spike at zero lag. Conversely, high values of γ, which regularise each term differently, prevent the model from reaching any sensible convergence.

In practical terms, λ is a substantially more challenging hyperparameter to tune compared to γ. Since γ is applied directly to the input and target variables, its values are always relative to the numerical range of the data. For datasets scaled between 0 and 1, γ values ranging from 0.1 to 0.3 were empirically observed to stabilise and accelerate training without compromising the results. On the other hand, because λ

is applied in the frequency domain, its adequate range highly depends on the relative magnitudes contained in the data spectrum, which varies across datasets. For the MNIST experiments presented, the best value for λ was 7, with higher values leading to blurry results. In contrast, some other datasets explored required an λ value of approximately 250.

Finally, for these experiments, I observed no additional benefit from adding noise to the penalty function or using a filter larger than twice the image size. However, I found that clamping the filters at a minimum value of 0, thus enforcing consistently positive values, helped speeding up convergence.

3.2.3 Normalisation and Relative Pre-Whitening

Since composing the previous section, I have incorporated several implementations to improve the pre-whitening stability of the Wiener Loss. The first modification involved normalising the magnitude of the cross-correlation and the auto-correlations in Equation 3.5 by their respective root mean square values (RMS). This normalisation prevents computations with excessively high values that could propagate numerical instabilities to the network. The second modification involved using the pre-whitening value λ as a percentage of the magnitude of RMS of the cross-correlation, applied equally to the nominator and denominator terms in Equation 3.5. This approach helps standardising tests with λ across datasets, reducing variability. However, since λ is primarily designed to address *relative* amplitude imbalances within the cross-correlation and auto-correlation, adequate values of this parameter remain dataset dependent. The results presented in Sections 3.2.4 and 3.2.5 make use of this normalisation and relative pre-whitening strategies described.

3.2.4 Reconstruction of Natural Images

In this section, the Wiener Loss is applied to a reconstruction problem for the CelebA (Liu et al., 2015) dataset, which poses a comparatively greater challenge to MNIST. The CelebA dataset comprises colour images of celebrity faces with a higher resolution (typically 178x218 pixels) and larger volume ($\sim$ 200,000 images). These images have more complex and diverse features, including variations in facial expressions,

lighting conditions, hairstyles, image corrections, and varied backgrounds that are harder for the network to capture. Additionally, the dataset includes faces of various celebrities with variability in pose, expression, accessories (e.g. glasses and hats), and other attributes, which makes it more challenging for models to generalise.

The reconstruction problem is set up similarly to Section 3.2.2, although here the comparison of the Wiener Loss is presented alongside four other methods: MSE, FFL, LPIPS, and an adversarial loss. The latter is also known as an adversarial autoencoder (AAE) (Makhzani et al., 2016). The mathematical formulations of these methods have been detailed in Section 2.1.2.

The network architecture is consistent across the methods to ensure fair experimentation. It comprises a VAE with a U-Net (Ronneberger et al., 2015) backbone, consisting of a feature layer with 32 channels and 2 downsampling and upsampling layers of channel dimensions (64, 64), each containing 3 residual units and a group normalisation layer (Wu and He, 2018) with a group size of 32 to improve generalisation. Each downsampling layer reduces the image size by a factor of 2, resulting in a latent space size of $\frac{128}{2^2} = 32$, and arbitrarily chosen to be single-channel. Overall, this model contains just over 1 million trainable parameters and was trained with the Adam (Kingma and Ba, 2014) optimiser, parameterised by $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and Mish (Misra, 2019) as activation function. Additionally, the KL divergence was applied to each loss with a low weight value of 1×10^{-4} to function as a simple latent regulariser. This architecture is made robust such that each loss function can effectively address the task at hand, limited only by its own formulation. The data was split into 80% training and 20% validation sets, with each image resized to a resolution of 128x128. For AAE training, the discriminator network consisted of 16 feature channels and 2 linear layers. The LPIPS loss is computed via a VGG-16 (Simonyan and Zisserman, 2014) network trained on a ImageNet (Deng et al., 2009) classification task. For all experiments, the model was trained with a batch size of 64 for 5 epochs until convergence. Evaluations over the validation set for the model trained with each loss are shown in Figure 3.10.

All the aforementioned losses, except for the Wiener Loss, are best implemented jointly with the MSE loss, and the presented results reflect this combination, FFL



Figure 3.10: Reconstruction of CelebA (Liu et al., 2015) images using a variational autoencoder (VAE) trained with different loss functions. The first row shows the ground truth, or target, image. Each following row represents the model trained with different losses (LPIPS, Wiener Loss, AAE, FFL, and MSE), and each column shows a different sample from the validation set. The Wiener Loss model outperforms all other models in terms of resolution, except for LPIPS. However, the Wiener Loss model demonstrates superior fidelity in preserving the original image details compared to all other models.

and LPIPS using a 1:1 MSE-to-loss ratio, and AAE 10:1. In certain contexts, incorporating a minor weight of the MSE loss to the Wiener Loss can facilitate amplitude corrections; however, I observed that this addition has little to no impact on model convergence and was not necessary for this specific task. This highlights the robustness of the Wiener Loss as a comprehensive and self-encompassing training reconstruction problems.

The Wiener Loss experiment yielded the best results when the gamma value was set to $\gamma = 0.1$ and λ was applied as a relative value expressing 0.5% of the cross-correlation RMS. The Wiener filters were selected to be two-dimensional and

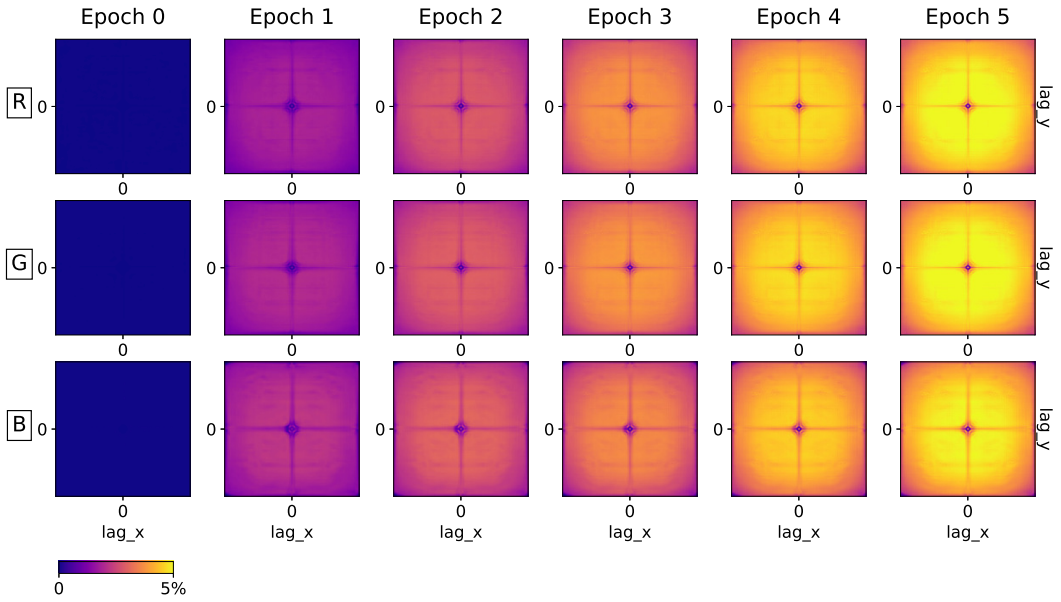


Figure 3.11: Evolution of the 2D trainable penalty matrix for the Wiener Loss experiment in reconstructing CelebA (Liu et al., 2015) images, illustrated in Figure 3.10. The rows represent different colour channels (R: top, G: middle, B: bottom), while the columns correspond to epochs of training. Colours indicate the percentage drop in values of the penalty matrix from its initial state. As training progresses, penalisation intensifies around the zero-lag axes and the vicinity of the zero-lag point, reflecting the model's efforts on recovering high-frequency components.

twice the size of the image, and were computed in the `reverse` mode in the Fourier domain. An additional improvement to the Wiener Loss implementation, which enhanced convergence in this reconstruction task, was enabling the penalty matrix of Figure 3.4 to be a trainable matrix of the same size as the filters, jointly optimised with the network. This dynamic implementation allowed the filter penalisation to be learned directly from the data and to evolve and adapt as the model converged. The evolution of the penalty matrix over epochs for each RGB channel is depicted in Figure 3.11 and is presented as a percentage drop from its initialised value.

Optimal performance with the trainable penalty parameters is achieved when they are initialised at a constant high value, arbitrarily set to 100, and when their learning rate is set inversely proportional to the total number of optimisation steps. This approach prevents the penalty parameters from collapsing to zero value, optimising the loss with meaningless filters. Figure 3.11 illustrates that the optimised penalty maintains a good degree of symmetry and remains relatively consistent across channels, in alignment with the statistical assumption of decorrelation discussed in

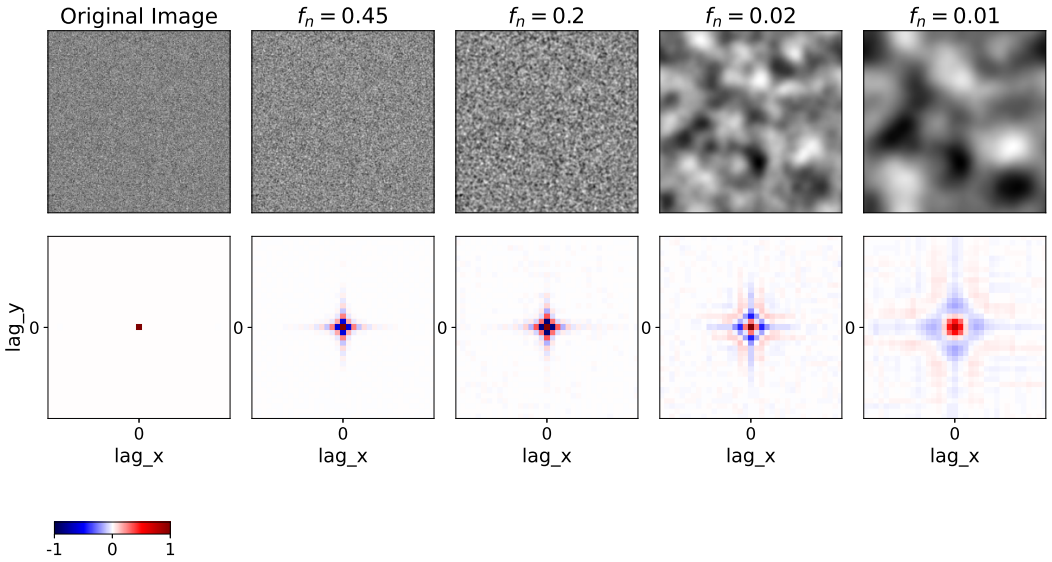


Figure 3.12: Comparison of Wiener filters (bottom) between random Gaussian noise and low-passed versions of itself, at different frequencies (top). The level of high and middle-frequency degradation increases to the right. The low frequency images are generated using a Butterworth low-pass filter of order 2, and the frequencies are normalised in the range [-0.5, 0.5]. The top row displays the filtered images at different frequency caps, while the bottom row shows the normalised 2D Wiener filters between the original and low-passed images, zoomed at zero lag for better visualisation. The blurrier the image, the more concentrated the filter energy becomes at the zero-lag axes and at the radial vicinity of the zero-lag point.

Section 3.1.2. An unexplored and more elegant design choice would be to treat the penalty matrix as an energy function whose sum is constrained to a constant value of 1. This would naturally prevent zero-value collapse, but would also force the filter to learn relative suppression.

In addition, Figure 3.11 demonstrates that the network benefits from maintaining higher penalty values for the zero-lag axes, and particularly at the vicinity of the zero-lag point. This behaviour is intuitive, as the Wiener filter that transforms a signal into its low-frequency counterpart retains most of its energy at these specific regions. This is observed, for instance, in a simple demonstration presented in Figure 3.12, which shows the resulting Wiener filter between a random Gaussian noise signal and low-passed versions of itself, at different frequencies. The ring and cross-like structures at the vicinity of the zero-lag point are gradually more pronounced as the high-frequencies degrade, matching the structures observed in Figure 3.11. The network, therefore, attempts to recover high frequencies by focusing the energy

suppression in these regions, particularly as it reaches convergence. Note, however, that a drop of 5% still results in substantial penalisation in the remaining areas of the filter.

In the CelebA reconstruction task, the Wiener Loss demonstrates superior resolution of accurate details compared to both MSE and FFL, with FFL showing minimal improvements over its MSE baseline. Although LPIPS excels in frequency recovery, its fidelity is lower than that of the Wiener Loss, exhibiting more hallucinations in high-frequency regions and poorer stylistic matching. For example, this can be observed in the eyes and mouth features in all reconstructions of Figure 3.10, hair strands and light details in reconstructions of column 7, and the impressive reconstruction of the hat with the "NY" text in column 6, achieved by the Wiener Loss.

This is further corroborated by the metrics presented in Table 3.1, where the Wiener Loss exhibits better performance in all metrics except for LPIPS and the Fréchet Inception Distance (FID) (Heusel et al., 2017). The latter, being based on the mean and covariance of features extracted from a pre-trained network, is adversely affected by blurriness where variance is reduced at higher frequencies, resulting in greater mismatches and poorer scores.

For AAE, the model produces sharper edges but also fails to match the fidelity of the Wiener Loss, although improved adversarial balance could potentially enhance the results. This, however, highlights a significant limitation of adversarial training, as achieving optimal performance requires delicate selection of network architectures and other hyperparameters.

Overall, the Wiener Loss offers a self-contained and comprehensive method for recovering high-fidelity, high-resolution structures from complex data. The results presented in this section demonstrate its effectiveness in handling natural images, establishing it as a strong competitor among the highly regarded methods used in the deep learning community.

	MSE ↓	Wiener Loss ↓	LPIPS ↓	MS-SSIM ↑	PSNR ↑	FID ↓
LPIPS	2.87×10^{-3}	4.32×10^{-2}	$\mathbf{1.28}\times10^{-2}$	9.65×10^{-1}	2.55×10^{1}	$\mathbf{2.92}\times10^{-1}$
WienerLoss	$\mathbf{9.98}\times10^{-4}$	$\mathbf{9.45}\times10^{-3}$	6.22×10^{-2}	$\mathbf{9.95}\times10^{-1}$	$\mathbf{3.00}\times10^{1}$	6.31×10^{-1}
AAE	2.20×10^{-3}	4.16×10^{-2}	2.57×10^{-1}	9.61×10^{-1}	2.66×10^{1}	1.73×10^{0}
FFL	1.90×10^{-3}	3.77×10^{-2}	2.36×10^{-1}	9.67×10^{-1}	2.72×10^{1}	1.62×10^{0}
MSE	1.98×10^{-3}	3.93×10^{-2}	2.45×10^{-1}	9.64×10^{-1}	2.70×10^{1}	1.66×10^{0}

Table 3.1: Performance metrics for the reconstruction problem of CelebA (Liu et al., 2015) with a Variational Autoencoder (VAE) using different loss functions (rows). Metrics include Mean Squared Error (MSE), Wiener Loss, Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018a), Multi-Scale Structural Similarity (MS-SSIM), Peak Signal-to-Noise Ratio (PSNR), and Fréchet Inception Distance (FID) (Heusel et al., 2017). The LPIPS metric employed a VGG-16 (Simonyan and Zisserman, 2014) network pre-trained on ImageNet (Deng et al., 2009), whereas the FID metric utilised features from an InceptionV3 (Szegedy et al., 2016) network, also pre-trained on ImageNet (Deng et al., 2009), following standard deep-learning practices. These metrics were generated using the 'torchmetrics' (Detlefsen et al., 2022) package. The Wiener Loss demonstrates superior fidelity, as indicated by the lowest MSE values, the highest MS-SSIM and PSNR, and competitive FID and LPIPS scores, outperforming other losses in terms of overall image quality and structural preservation.

3.2.5 Medical Scan Imputation

Imputation (Dalca et al., 2019) is a well-established technique across the medical imaging field that addresses sparse or limited resolution acquisitions through statistical estimation. Engineering datasets are often compromised due to poor design, unexpected failures or restrictions in acquisition times (Liu and Sacchi, 2004; Wang et al., 2019; Zhao et al., 2019; Li et al., 2015; Brudfors et al., 2019; Dong et al., 2015). Interpolation is a popular strategy to imitate or simulate missing data, but it is prone to failure when the amount of missing information is large (Aganj et al., 2012). Imputation (Dalca et al., 2019) is an alternative strategy to estimate missing values using statistics from across the dataset. In this section, I focus on the imputation of Magnetic Resonance Imaging (MRI) samples of the brain, which due to large resources and acquisition time constraints are often sparse, have varying resolutions, or limited fields of view (Dalca et al., 2019, 2017; Brudfors et al., 2019; Rousseau et al., 2010).

For this experiment, I designed a simple neural network to perform an imputation on MRI samples, and, similar to Section 3.2.4, the performance of the Wiener Loss is compared against MSE, FFL, LPIPS and adversarial losses. An additional

method uses bicubic interpolation to address the task and serves as a baseline for the deep-learning driven models. Like the previous section, the network used for these experiments is a U-Net (Ronneberger et al., 2015) VAE, kept consistent across all methods to ensure fair experimentation. The VAE architecture is identical to the one used in Section 3.2.4, except for an increase in latent size to three channels to capture more details from the larger image sizes, and an adversarial network for AAE comprising 8 feature layers instead of 16. The training parameters differed only in that the KL weight was set to 1×10^{-6}, and the networks were trained with a batch size of 32 for 100 epochs until convergence.

The dataset used to train the networks is the 1,000 T1-weighted MRI volumetric scans of healthy adult brains from the HCP Young Adult database (Van Essen et al., 2012). These scans were sliced in their mid-sagittal plane to obtain single-channel 256x256 samples for 2D tests. Further details of the dataset are described in Chapter 4, where a more in-depth analysis is provided in a higher relevance context. For this experiment, outliers were removed by capping values above the upper quartile range, and the samples were scaled between 0 and 1 using an image-wise min-max normalisation method.

Acquisition sparsity was simulated by corrupting the full-resolution images with a sampling mask, which served as input to the network. The chosen mask omitted 2 every 5 acquisition lines, and was rotated by 10 degrees to simulate non-principal acquisition axes and prevent any vertical or horizontal biases in the network. The network was trained by minimising a data comparison metric, defined by the various loss functions, between the model outputs and the full-resolution, uncorruped images. Output samples from the validation set for each method are compared in Figure 3.13.

The Wiener Loss experiment achieved the optimal results $\gamma = 0.1$ and $\lambda = 1 \times 10^{-3}$, representing 0.1% of the cross-correlation RMS. The Wiener filters were selected to be two-dimensional and twice the size of the image, and were computed in the `reverse` mode in the Fourier domain. Similar to Section 3.2.4, the penalty was a matrix of the same size of the filters, with entries initialised at a constant value of 100 and jointly optimised with the network. The final optimised penalty for this experiment is shown in Figure 3.14, and is a higher-resolution and more feature

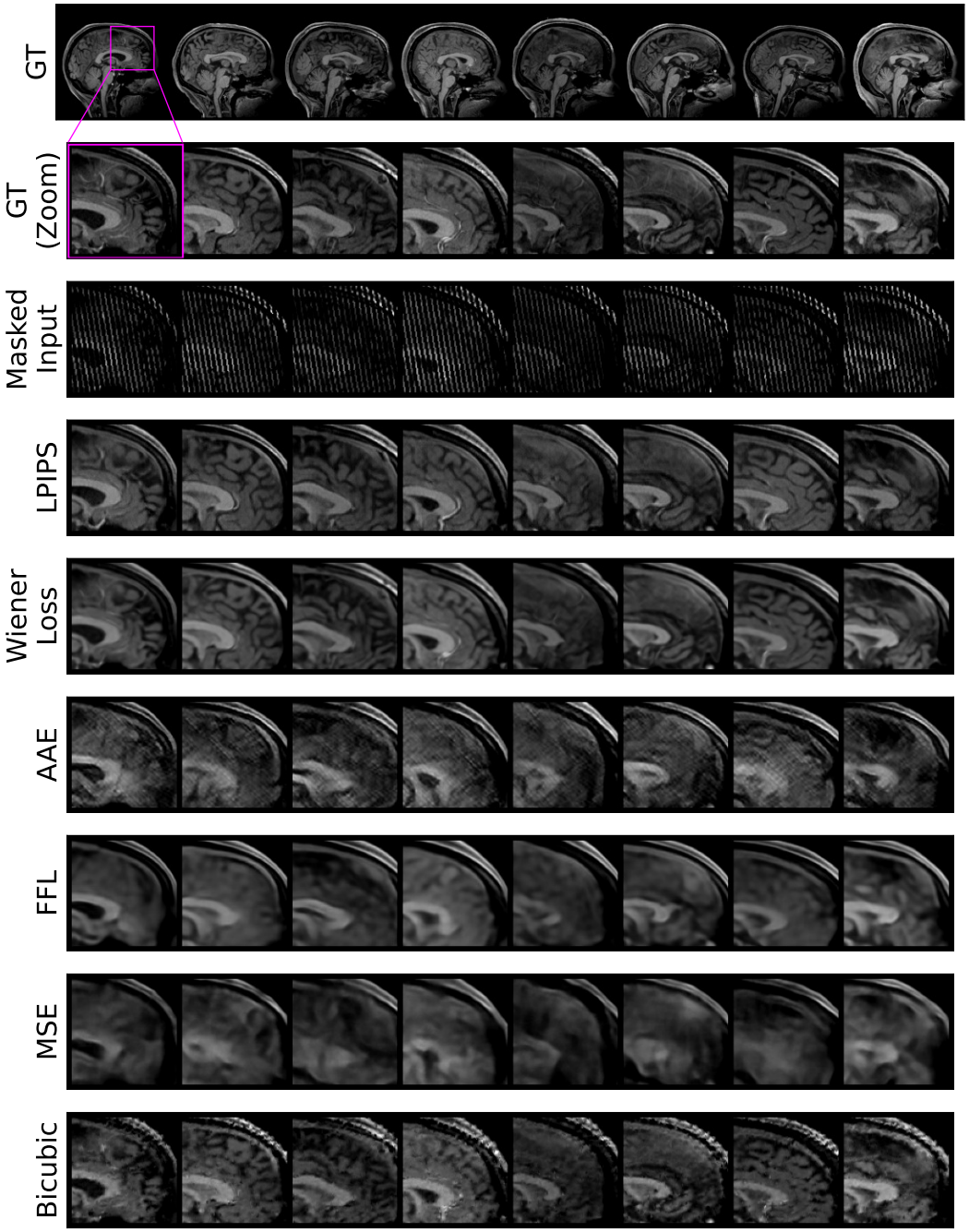


Figure 3.13: Outputs of a variational autoencoder (VAE) trained to impute heavily corrupted T1-weighted MRI images, with the corruption shown in the third row. The first row presents the ground truth (GT) images from the validation set, and the second row provides a zoomed version of the GT for better visualisation. Subsequent rows display the VAE outputs using different loss functions: LPIPS, Wiener Loss, Adversarial Autoencoder (AAE), Focal Frequency Loss (FFL), Mean Squared Error (MSE), and a simple bicubic interpolation for benchmarking. The MSE and FFL models show minimal differences and fail to recover fine details, resulting in blurry outputs. While the LPIPS model excels in capturing high-frequency details, the Wiener Loss outperforms it in preserving the fidelity of finer anatomical structures such as the ventricles and fissures of the brain, which appear brighter in T1-weighted images.

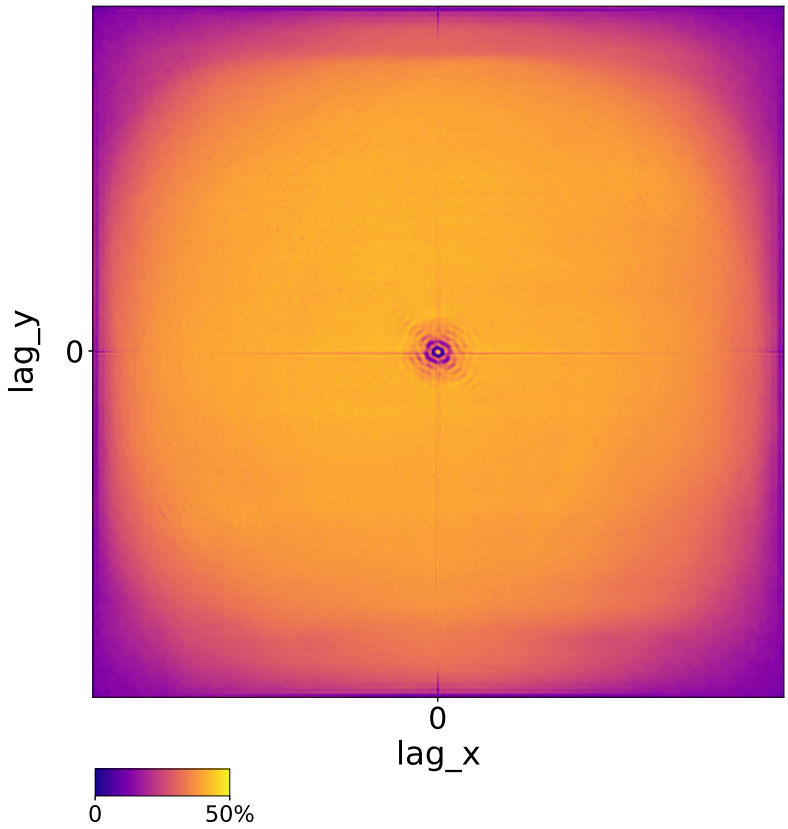


Figure 3.14: Optimised 2D penalty matrix of the final training epoch for the Wiener Loss experiment in the imputation of MRI scan images, as illustrated in Figure 3.13. Colours indicate the percentage drop in values of the penalty matrix from its initial state. The evolution of the penalty closely resembles the one presented in Figure 3.10, for the face reconstruction experiment, although for this example the percentage drop of $\sim 50\%$ is much larger. The cross-like structure at the zero-lag axes and the ring-like structures surrounding the zero-lag point correspond to the near-zero energy in the filters of the low-passed images shown in Figure 3.12. This highlights the penalty's substantial effort to recover high-frequency components.

rich version of the optimised penalties from Figure 3.11. The cross-like structure at the zero-lag axes and the ring-like structures surrounding the zero-lag point match the near-zero energy in the filters of the low-passed images in Figure 3.12. Along with the larger 50% drop of penalty values in the remaining areas, this highlights the Wiener Loss' great effort to recover high-frequency components.

The results presented in Figure 3.13 demonstrate the efficacy of the Wiener Loss in achieving the heavily-corrupted imputation task, qualitatively corroborated in Table 3.2. In the medical context, the fidelity of recovery is of utmost importance, surpassing even the need for detailed image resolution. The recovered images under

the Wiener Loss implementation are of superior quality compared to its MSE, FFL and AAE counterparts. The recoveries from the MSE and FFL models fail to drive the network towards detailed reconstructions, resulting in blurry outputs that perform worse than the bicubic baseline. The AAE model achieves better resolutions than the MSE and FFL models, but the fidelity of these recovered scans is poor and the images contain obvious artefacts. This behaviour could be an effect of a small dataset, a scenario in which adversarial losses are known to underperform (Li et al., 2022). The low quality of these three models could potentially be mitigated by improving the network architecture and fine-tuning training parameters. However, in this study, they serve as benchmarks to present the robustness of the Wiener Loss.

The model under LPIPS training remains the most significant competitor to the Wiener Loss, outperforming it in perceptual quality. Nonetheless, the Wiener Loss provides superior fidelity and suppressed hallucination artefacts, as evidenced by its ability to distinctly preserve finer anatomical structures of the brain, seen in the improved recovery of ventricles and fissures forms. The high level of fidelity achieved by the Wiener Loss could have a significant impact on the accurate recovery of pathologies in future experiments, making it a more reliable choice for clinical applications.

3.3 Further Applications

In this section, three additional avenues for exploring the Wiener metric in deep learning applications are discussed. These avenues have been partially explored or are in the early stages of investigation by colleagues, with my collaboration. It has been included in this research chapter for the sake of completeness.

3.3.1 Translation Invariance

The first avenue of exploration involves exploiting the properties of Wiener filters concerning translated signals. The Wiener filter that arises when $\mathbf{x}$ is a translated copy of $\mathbf{y}$ is simply a Dirac delta function, translated in the same manner. This concept is visually demonstrated in Figure 3.15, where the resulting 2D Wiener filter

	MSE ↓	Wiener Loss ↓	LPIPS ↓	MS-SSIM ↑	PSNR ↑	FID ↓
LPIPS	2.04×10^{-3}	2.40×10^{-2}	$\mathbf{1.48}\times10^{-1}$	9.49×10^{-1}	2.65×10^{1}	$\mathbf{5.38}\times10^{-1}$
WienerLoss	$\mathbf{1.46}\times10^{-3}$	$\mathbf{1.40}\times10^{-2}$	2.29×10^{-1}	$\mathbf{9.74}\times10^{-1}$	$\mathbf{2.79}\times10^{1}$	3.08×10^{0}
AAE	3.62×10^{-3}	3.88×10^{-2}	3.85×10^{-1}	8.81×10^{-1}	2.39×10^{1}	5.59×10^{-1}
FFL	2.29×10^{-3}	3.47×10^{-2}	3.76×10^{-1}	8.92×10^{-1}	2.59×10^{1}	6.65×10^{0}
MSE	3.56×10^{-3}	4.56×10^{-2}	4.36×10^{-1}	8.33×10^{-1}	2.40×10^{1}	6.17×10^{0}
Bicubic	3.00×10^{-3}	3.15×10^{-2}	3.01×10^{-1}	8.48×10^{-1}	2.45×10^{1}	4.29×10^{-1}

Table 3.2: Comparative metrics for the imputation problem of heavily-corrupted MRI scans using a variational autoencoder (VAE) with different loss functions (rows). The LPIPS (Zhang et al., 2018a) metric utilised a VGG-16 (Simonyan and Zisserman, 2014) network pre-trained on ImageNet (Deng et al., 2009), while the FID (Heusel et al., 2017) metric employed features from an InceptionV3 (Szegedy et al., 2016) network, also pre-trained on ImageNet (Deng et al., 2009). These metrics were generated using the *torchmetrics* package (Detlefsen et al., 2022). The Wiener Loss exhibits superior performance in terms of MSE and MS-SSIM, and shows competitive results in the LPIPS metric, outperforming traditional losses such as MSE, FFL and AAE, and the bicubic interpolation. The model trained under the LPIPS metric shows better performance due to its sharper resolution. But overall, these metrics demonstrate the efficacy of the Wiener Loss in recovering high-fidelity details when imputing MRI scans.

that matches a sample from the MNIST dataset to a translated copy of itself is the equally translated delta function.

We can thus formulate a problem where this translation invariability can be exploited. To illustrate this, I present a simple experiment using a K-Nearest Neighbours (KNN) algorithm (Cover and Hart, 1967) for a classification problem. The task is to classify digits from the affNIST dataset (Tieleman, 2013), which consists of padded MNIST images that have undergone affine transformations, using only the original (but equivalently padded) MNIST data as the training set. Samples from the two sets are presented in Figure 3.16.

A baseline model is first evaluated using the Manhattan distance metric and a $k = 3$ to classify the transformed digits. This model failed to classify samples from the affNIST test set, achieving an accuracy of **14%**. The poor performance is attributed to the translation of the digits which fails to achieve reasonable matching scores under any pixel-wise metric.

A Wiener-based metric can be formulated to be translation invariant:

$$\mathcal{L}_{\text{TI}}(\mathbf{x}, \mathbf{y}) = -\max\left(\frac{\mathbf{v}(\mathbf{x}, \mathbf{y}) - \mu_{\mathbf{v}}}{\sigma_{\mathbf{v}}}\right) \tag{3.12}$$

where $\mu_{\mathbf{v}}$ is the mean value of filter $\mathbf{v}(\mathbf{x}, \mathbf{y})$, and $\sigma_{\mathbf{v}}$ is its standard deviation. If the energy of $\mathbf{v}(\mathbf{x}, \mathbf{y})$ is focused in a narrow spike, then the process of standardisation amplifies the magnitude of the spike to ensure unit standard deviation. On the other hand, if the energy of $\mathbf{v}(\mathbf{x}, \mathbf{y})$ is unfocused and distributed across all lags, then unit standard deviation is achieved without much interference on the filter. $\mathcal{L}_{\text{TI}}$ is therefore only sensitive to whether the energy in the filter is focused, but not where. As supported by Figure 3.15, $\mathcal{L}_{\text{TI}}$ is thus invariant to translation. Using $\mathcal{L}_{\text{TI}}$ as a distance metric in KNN (Cover and Hart, 1967), with a $k = 10$ on the same task achieves a dramatically improved test accuracy of **61%**. The confusion matrix in Figure 3.17 shows the contrast in performance between models.

This simple example demonstrates the effectiveness of a Wiener-based metric in resolving tasks where translational invariance is important, and paves the way for further exploration on the topic.

3.3.2 Wiener Diffusion

The second avenue of research has been conducted by my co-author Dr. George Strong and is thoroughly explored and discussed in their thesis (Strong, 2023). In that work, they combined the AWI objective with the Wiener loss to develop a non-parametric generative model, named Wiener Diffusion, which can effectively sample from complex latent spaces and does not require model training. The Wiener Diffusion is introduced as a new generative model that employs Wiener filters to

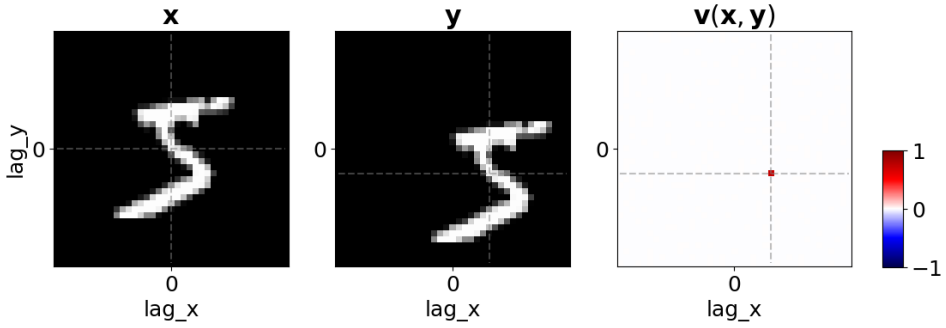


Figure 3.15: Schematic illustrating that the Wiener filter between a signal and a translated copy of itself results in a Dirac delta function, translated in the same manner. This property is exploited to formulate a translation invariant version of the Wiener Loss.

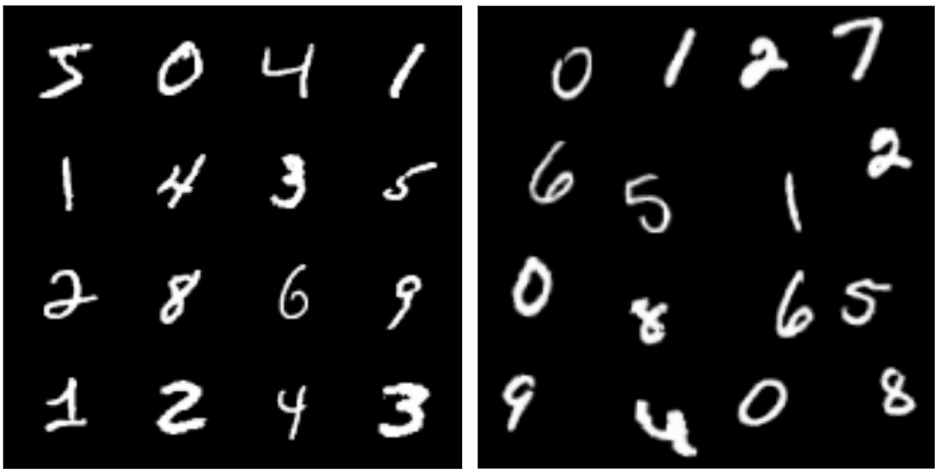


Figure 3.16: Image classification task to use the MNIST training data (left) to predict the affNIST testing data (right). A KNN classifier with the Manhattan distance metric achieves and accuracy of **14%**, whereas a KNN classifier with the proposed translation invariant metric $\mathcal{L}_{\text{TI}}$ achieves an accuracy of **61%**

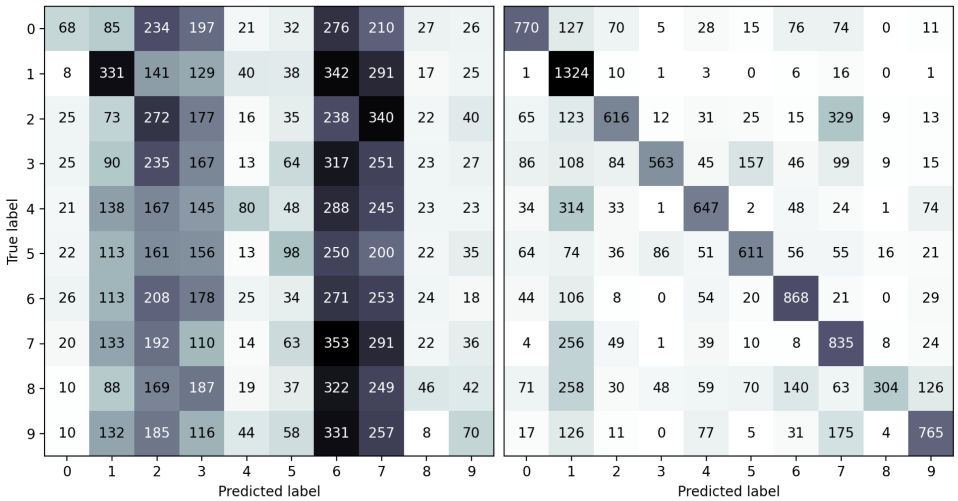


Figure 3.17: Confusion matrices for comprehensive multi-class performance evaluation of the baseline KNN classifier (left), vs the translation invariant KNN (right). Performance is substantially improved using the proposed translation invariant metric $\mathcal{L}_{\text{TI}}$. The Wiener filters in Equation 3.12 were computed in the Fourier domain in the `reverse` mode, with a 2-dimensional filter twice the size of the input image, and absolute pre-whitening $\lambda = 100$. Extracted from Cruz et al. (2023a)

define a non-parametric energy-based model, $E(\mathbf{x})$, representing the non-normalised negative log-likelihood of the data distribution. The probability density of $E(\mathbf{x})$ in energy-based models is expressed by the Boltzmann distribution:

$$p(\mathbf{x}) = \frac{\exp(-E(\mathbf{x}))}{\int \exp(-E(\mathbf{x}))\ d\mathbf{x}}. \tag{3.13}$$

Direct computation of $p(\mathbf{x})$ is computationally intractable due to the normalisation term in the denominator (MacKay, 2003). However, efficient sampling can be performed using Langevin Markov Chain Monte Carlo (MCMC) methods, which utilise the gradient of the log-energy density (Song et al., 2021b; Song and Ermon, 2019; Grenander and Miller, 1994; Rossky et al., 1978). The Langevin MCMC diffusion process for generating samples over T steps is defined as:

$$\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \frac{\alpha_t}{2}\nabla E(\mathbf{x}_t) + \mathbf{z}_t, \text{ for } t = 0, 1, \cdots, T-1, \tag{3.14}$$

where, $\mathbf{x}_t$ is the sample at step t, α_t represents the step size schedule, and $\mathbf{z}_t$ denotes random noise sampled from a Gaussian distribution $\mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \beta_t)$ defined by the variance schedule β_t. The energy function for Wiener diffusion is given by:

$$E(\mathbf{x}) = \sum_{i=1}^{N} \frac{1}{2}\mathcal{R}(\mathbf{T}, \mathbf{v}(\mathbf{x}, \mathbf{y}_i)) + \frac{\gamma}{2}||\boldsymbol{\delta} \otimes (\mathbf{v}(\mathbf{x}, \mathbf{y}_i) - \boldsymbol{\delta})||^2, \tag{3.15}$$

where, the summation is over the dataset samples, γ is a scalar controlling the strength of the second term, and $\otimes$ denotes the Hadamard product. The first term minimises the Rayleigh quotient of a penalty matrix $\mathbf{T}$ and Wiener filters $\mathbf{v}(\mathbf{x}, \mathbf{y}_i)$, as defined in Equation 3.3, pushing the data-matching Wiener filters towards zero-lag spikes. The second term, a corrective component, ensures sensitivity to amplitude information, maintaining the convolutional identity by encouraging the zeroth-lag of the filters to have an amplitude of 1. The noise term in the Langevin diffusion process helps prevent overfitting by avoiding the collapse of generated samples onto the training data.

The penalty matrix $\mathbf{T}$ shapes the energy landscape and must be carefully designed to facilitate diverse and accurate sample generation. Similar to AWI, $\mathbf{T}$ is

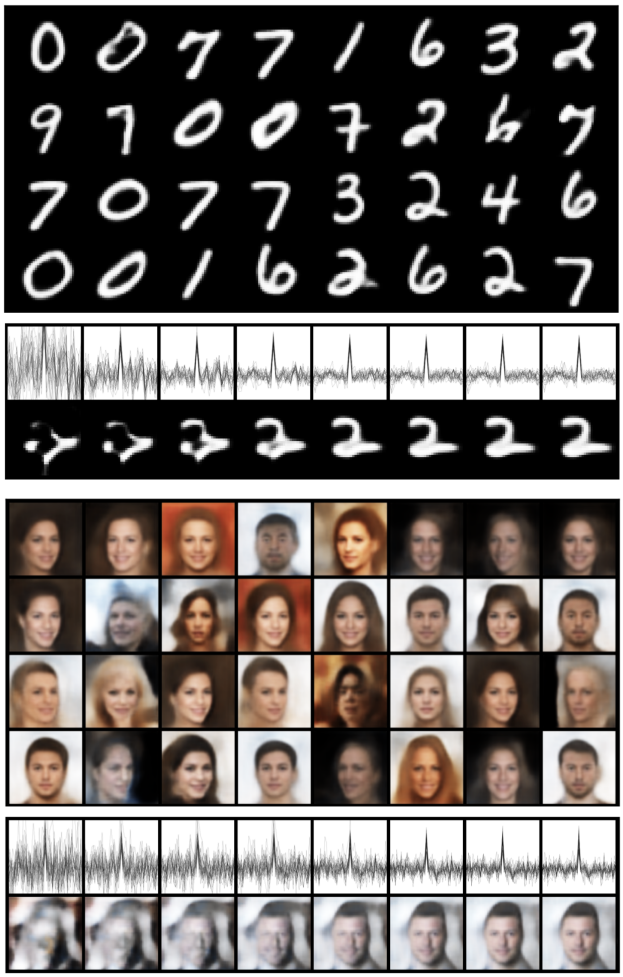


Figure 3.18: Latent samples generated using the Wiener diffusion, visualised using the decoder, and an illustration of the diffusion process alongside the 30 closest Wiener filters to the sample $\mathbf{x}_t$ (as defined by 3.15) for MNIST (LeCun et al., 1998) (top) and CelebA (Liu et al., 2015) (bottom).

taken as a diagonal matrix with a monotonically decreasing penalty function centred at zero-lag. Additionally, the gradient of the penalty function must be monotonically increasing towards the zeroth-lag to ensure the generated samples remain close to the data distribution in a convolutional sense, while concurrently ensuring the sampling does not collapse towards the average of the dataset. Examples of the Wiener Diffusion applied to MNIST and a resized version of CelebA (Liu et al., 2015) is shown in Figure 3.18. For a more comprehensive exploration of the topic, the reader is encouraged to consult his thesis (Strong, 2023).

3.3.3 Wiener Attention

The third and final proposed research avenue is a reformulation of the concept of self-attention, proposed in (Vaswani et al., 2017), using the Wiener Loss. Self-attention mechanisms have revolutionised the deep learning community by enabling models to dynamically focus on different parts of an input sequence, capturing long-range dependencies in the data more effectively. This innovation has been a cornerstone in the development of state-of-the-art architectures like the Transformer (Vaswani et al., 2017), which has largely advanced natural language processing and other sequence modelling tasks. The self-attention mechanism is defined in Equation 2.112, and is discussed in greater detail in Section 2.1.6.

A natural implementation of the Wiener-based metric within the self-attention mechanism lies in the comparison of Q and K, under a new measure of similarity with the Wiener Loss defined in 3.4. However, rather than computing the batch average for each optimisation step, here the Wiener Loss is employed to measure the individual similarity between each i-th query to every j-th key, such that:

$$\mathcal{L}_{\mathrm{w}}^{i,j}(\mathbf{q}_i, \mathbf{k}_j) = \frac{1}{2}||\mathbf{W}\left\{\mathbf{v}(\mathbf{q}_i, \mathbf{k}_j) - \boldsymbol{\delta}\right\}||^2, \tag{3.16}$$

and $\mathcal{L}_w(Q, K)$ is the matrix constructed with Equation 3.16 for every possible i and j combination. The proposed Wiener Attention method then takes the following form:

$$\text{WienerAttention}(Q, K, V) = \text{softmax}\left(\frac{\mathcal{L}_w(Q, K)}{\sqrt{d_k}}\right)V. \tag{3.17}$$

By applying the Wiener metrics to examine each query and key vector from a global convolution perspective, the proposed method aims to enhance relevance and performance in their comparison. This is presented here as a theoretical proposition, with a preliminary exploration of the method previously submitted as a master's dissertation by Andrei Danilla, whom I co-supervised in 2024 at Imperial College London. Experimentation with this method is beyond the scope of this thesis.

3.4 Future Work

Despite the promising results demonstrated by the Wiener Loss, a few areas of this proposed methodology still fall short from optimal. For instance, the bias towards low frequencies at the beginning of training can hinder the speed of convergence. The use of trainable penalty matrices seems to partially address this issue, but it remains largely untested, particularly regarding its initialisation and learning rate parameters. The consistency with which the penalty matrices evolve during training across different tasks stands out, especially in how the matrix penalises the zero-lag axes and the vicinity of the zero-lag point. Exploiting this consistent pattern as optimisation constraints might help accelerate the network's convergence towards higher frequencies. Additionally, experimenting with smaller receptive fields for the Wiener filter by decreasing its relative size, or applying the loss in image patches, similar to the approach in Dosovitskiy et al. (2020), could enhance the focus on local structures and further improve high-frequency recovery.

The most significant challenge of the Wiener Loss, however, is achieving numerical stabilisation. Numerical stabilisation in the presented experiments proved challenging and highly-data dependent. Future work should explore methods to make this stabilisation less dependent on the specific datasets. One option is to implement the Levinson-Durbin Recursion (LDR) using PyTorch's convolution modules, potentially achieving better performance at the cost of increased computational time. Another approach is to gain a deeper understanding of the statistical assumptions made by the Wiener Loss over the data, as applying adequate preprocessing on the data could mitigate the data-dependency of stabilisation values. Implementing a simple pre-whitening scheduler could also help stabilise convergence.

Finally, testing the Wiener Loss in 3D applications, with the imputation task being a prime candidate, represents another promising direction for future research. This extension could provide further insights into the efficacy and adaptability of the Wiener Loss in more complex data structures and alleviate the need to evaluate large networks to retrieve 3D features for perceptual losses.

3.5 Summary

In this chapter, I introduced the Wiener Loss, a new loss function designed, although not exclusively, for deep learning tasks where data comparison at high resolution and high fidelity is required. Formulated in a least-squares framework, it is memory-efficient, scalable, computationally efficient in the Fourier domain, and effective with smaller models and datasets. The Wiener Loss implementation is made available in a public GitHub repository, and consists of a single Python class derived from PyTorch's (Paszke et al., 2019) `nn.Module`, ensuring compatibility with other PyTorch losses and providing seamless workflow integration with minimal code. The class includes default parameters for straightforward use but is designed to be highly customisable to adapt to specific tasks.

The efficacy of the Wiener Loss is demonstrated in two deep-learning experiments: reconstruction of images of human faces and imputation of heavily-corrupted MRI medical scans, both leveraging a U-Net variational autoencoder architecture to address their corresponding tasks. In these experiments, the Wiener Loss showed remarkable consistency and robustness, outperforming traditional losses such as MSE, FFL, LPIPS and adversarial losses. While LPIPS is effective, it can be memory-intensive in large image sizes. Adversarial losses require substantial efforts to stabilise training, needing larger datasets and careful hyperparameter tuning. Despite its promising concept, FFL has proven unreliable in practice for the given tasks.

In contrast, the Wiener Loss is naturally contextually aware, and was able to improve the range of frequency recovery with high fidelity and reduced computational cost. It successfully reconstructed accurate details of key features such as eyes, hair, and mouth in the face reconstruction experiment, more reliably than its counterparts. In the MRI imputation task, it accurately recovered anatomical brain structures that were either not visible or poorly reconstructed by other loss functions. The success of the Wiener Loss in these experiments positions it as a powerful alternative to other loss functions.

I also demonstrated that the Wiener-filter-based approach can extend beyond the Wiener Loss to other specific tasks. I briefly discussed how the Wiener Loss can

be reformulated as a translation-invariant loss, a non-parametric diffusion model, and a novel method for comparing queries and keys in attention mechanisms. Although these propositions remain largely untested, they pave the way for promising expansions of the Wiener Loss methodology.

Despite the promising results demonstrated by the Wiener Loss, several areas remain open for future research. The most significant challenge is stabilising the Wiener filter computation in the frequency domain, as different datasets require tailored stabilisation parameters. Future work should explore methods to reduce this dependency, such as implementing an efficient algorithm for the Levinson-Durbin Recursion (LDR) or gaining a deeper understanding of the statistical assumptions made about the data to apply appropriate preprocessing. Additionally, the recovery of high frequencies from the Wiener Loss could be further improved by optimising the parametrisation of the trainable penalty matrices and reducing the receptive field of the filter to focus on localised structures. Finally, testing the Wiener Loss in 3D could provide further insights into its efficacy and adaptability in more complex tasks and data structures.

Chapter 4

Generative Ultrasound Brain Imaging with Unconditional LDMs

"Creating noise from data is easy; creating data from noise is generative modeling"

– Yang Song

Breakthroughs in deep learning have primarily emerged within the domains of computer vision, natural language processing, and reinforcement learning. These advancements are largely attributable to the availability of extensive datasets comprising millions of text and image samples. Landmark datasets, such as COCO (Lin et al., 2014) and ImageNet (Deng et al., 2009), have been crucial for developing influential image generation models (He et al., 2017; Redmon et al., 2016; Lin et al., 2017b; Krizhevsky et al., 2012; Simonyan and Zisserman, 2014; He et al., 2016), while extensive text datasets have enabled the emergence of transformative NLP frameworks like BERT (Devlin et al., 2019a) and GPT-3 (Brown et al., 2020). Multimodal architectures, exemplified by CLIP (Radford et al., 2021), further underscore the role of large-scale datasets in advancing deep learning research.

Despite the significant success of neural networks in medical imaging, particularly in segmentation and automated diagnosis, the widespread adoption of large-scale models remains limited by the scarcity of high-quality data. Acquiring medical samples is inherently challenging, as it is both labour-intensive and costly. Addi-

tionally, requirements to safeguard patient privacy further hinder the development of robust medical datasets by restricting data sharing and utilisation.

To mitigate the scarcity of medical samples, generative models have been employed to augment existing datasets (Liu et al., 2019; Lee et al., 2022; Sundaram and Hulkund, 2021; Rashid et al., 2019; Frid-Adar et al., 2018). These models are designed to approximate the underlying distribution of the training data $p_{\text{data}}(\mathbf{x})$, either directly or indirectly, enabling the generation of new instances that reflect the statistical properties of the original dataset. When adequately trained, generative models can produce synthetic samples on demand, expanding datasets while preserving fidelity to the distribution of real data.

Generative adversarial networks (GANs) (Goodfellow et al., 2014) have become a prominent method for synthetically augmenting datasets by leveraging the adversarial interaction between two neural networks, a generator and a discriminator, to produce high-resolution samples from Gaussian noise. These networks are trained concurrently in a zero-sum game (see Section 2.1.4), where the discriminator evaluates the generator's output to distinguish between real and synthetic samples. Despite their effectiveness, GANs exhibit notable limitations. The primary challenges include: (1) difficulties in convergence (Weng, 2017; Arjovsky et al., 2017) due to the instability of the zero-sum objective; (2) vulnerability to mode collapse (Goodfellow et al., 2014; Arjovsky et al., 2017; Weng, 2017), wherein the generator produces a restricted set of outputs that deceive the discriminator but fail to capture the full diversity of $p_{\text{data}}(\mathbf{x})$; and (3) a reliance on large quantities of data to train both networks effectively, limiting their capacity to generalise from small datasets, perpetuating the issue of data scarcity.

As an alternative, diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Rombach et al., 2022; Song et al., 2021b; Song and Ermon, 2019; Song et al., 2021a) have garnered attention for their ability to address the limitations of GANs. Unlike GANs, which directly map $p_\psi(\mathbf{x} \mid \mathbf{z})$, where $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ in a single inference step, diffusion models generate data through a gradual, multi-step process that incrementally transforms Gaussian noise into samples drawn from the target distribution modelling $p_\psi(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ for $t \sim [0, T]$ and $p_\psi(\mathbf{x}_T) \approx \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$. This iterative de-

noising approach improves training stability and enhances robustness against hyperparameter sensitivity and mode collapse, even when data availability is constrained (see Section 2.1.5). The success of diffusion models in applications such as DALL·E 2, Midjourney, and Google Imagen highlights their capacity to generate high-quality samples at scale, underscoring their potential for widespread use in computer vision tasks.

In the medical imaging field, recent work by Pinaya et al. (2022) has explored the application of diffusion models to enhance a small dataset of volumetric MRI scans. The study demonstrated that diffusion models could sample statistically significant and structurally accurate samples from the distribution of the training set. These samples are conditioned on sex, age and brain volume metadata that guide the operation of the diffusion steps. The authors publicly released a dataset comprising 100,000 synthetic samples, opening doors for further deep learning research in the field.

The scarcity of high-quality data samples is particularly pronounced in ultrasound imaging due to the limitations of conventional ultrasound tomography (UT) methods. Traditional UT relies primarily on travel-time inversion, resulting in lower image resolution compared to other imaging modalities (Calderon Agudo et al., 2018). In the context of brain imaging, acoustic waveforms dominated by strong skull reflections prevent UT from producing images of intra-cranial structures altogether, further exacerbating the scarcity issue.

To address this limitation, the work presented in this chapter builds on the methodologies established by Pinaya et al. (2022), adapting their approach to ultrasound brain imaging. Specifically, this chapter details the construction and training of a Latent Diffusion Model (LDM) for a small dataset of neuroacoustic images, parameterised by the acoustic properties of the brain. In contrast to the original study, this work employs a condition-free diffusion model, generating synthetic samples directly from Gaussian noise that initiates the diffusion process. However, the proposed framework is inherently adaptable, allowing for seamless integration of metadata conditioning in future research. The results demonstrate that, for a limited dataset ($N = 1{,}000$), a well-trained condition-free LDM can produce statistically consistent

synthetic samples, effectively augmenting existing neuroacoustic datasets.

In addition to addressing scarcity of ultrasound image data, this chapter serves as a precursor to more advanced explorations in the following chapter, where complex conditionings are introduced. Given the complexity of these conditionings, it's crucial to first construct a robust unconditioned diffusion model that will ensure stability and effectiveness of the techniques that will be developed later.

4.1 Latent Diffusion Models

As introduced in 2.1.5, denoising diffusion probabilistic models (DDPMs) represent a class of likelihood-based generative models that approximate the data distribution through a T-stepped denoising process applied to an arbitrary noise distribution. Training typically unfolds in two stages: the *perceptual compression* stage and the *semantic compression* stage (Rombach et al., 2022). The perceptual compression stage captures semantic variations by discarding high-frequency and imperceptible details, while the semantic compression phase learns to generate large-scale and conceptual structures of the data (Rombach et al., 2022).

In the data space, maximum-likelihood models disproportionately allocate computational resources to reconstructing perceptual details, leading to potential instabilities and prolonged training times for convergence. Latent diffusion models (LDMs) (Rombach et al., 2022) are introduced as a more efficient variant of conventional DDPMs, designed to improve convergence rates and reduce computational load during both training and inference. This improvement is achieved by decoupling perceptual and semantic compression into a two-stage framework (Figure 4.1): the first stage employs an autoencoder architecture to project data into a perceptually compressed latent space, simultaneously reducing its dimensionality; the second stage applies a denoising model that learns the iterative mapping from Gaussian noise to the distribution of the latent space, which is subsequently passed through the decoder to reconstruct perceptual details. This separation of tasks allows the diffusion model to focus exclusively on the semantic compression of the latent space, enhancing training efficiency and reducing computational overhead.

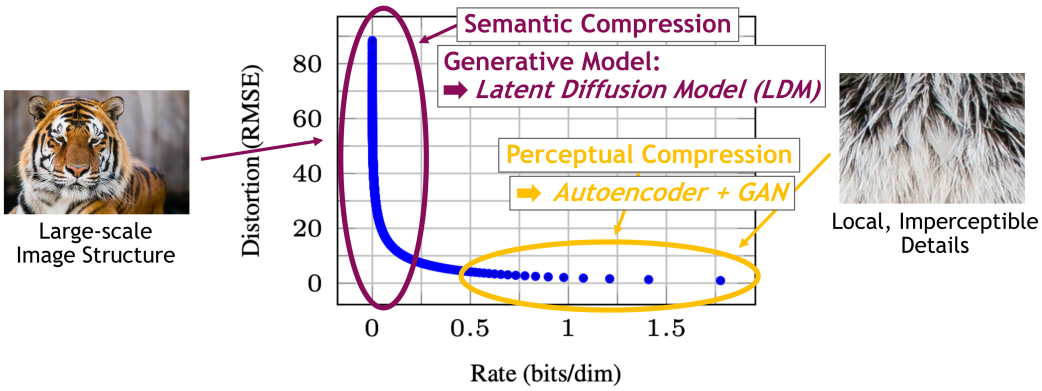


Figure 4.1: While standard diffusion models perform perceptual and semantic compressions simultaneously within the latent variables of the Markov chain, in latent diffusion models these tasks are separated: a first-stage training of an autoencoder performs perceptual compression, leaving the diffusion model to only perform the semantic compression. Image extracted from Kreis et al. (2023).

The first-stage autoencoder model consists of a regularised probabilistic encoder $\mathcal{E}_\theta(\mathbf{x})$, parameterised by trainable weights θ, which compresses a sample $\mathbf{x}$ into a latent representation $\mathbf{z}$ by learning $q_\theta(\mathbf{z} \mid \mathbf{x})$. Conversely, the probabilistic decoder $\mathcal{D}_\phi(\mathbf{z})$, parameterised by trainable weights ϕ, models the distribution $p_\phi(\mathbf{x} \mid \mathbf{z})$ to reconstruct the latent variable $\mathbf{z}$ back into the data space, yielding $\tilde{\mathbf{x}}$ (Figure 4.2). The training process for the first-stage model is outlined in Algorithm 2, with a detailed discussion of its formulation provided in Section 2.1.3.

With an effective first-stage model, the LDM operates similarly to a standard DDPM, with both the forward and reverse processes applied in the autoencoder's latent space (Figure 4.3). During the forward process, a sample $\mathbf{x}$ from the training data is compressed into its latent representation $\mathbf{z}$, which is progressively degraded to a noise sample ϵ over T steps, following Equation 2.59. As in DDPMs, a model $\boldsymbol{\epsilon}_\psi(\mathbf{z}_t, t)$, parameterised by ψ, learns the denoising proccess by estimating the noise ϵ at each timestep t from the degraded latent sample $\mathbf{z}_t$. See Algorithm 7. This is achieved by minimising the following objective:

$$\mathcal{L}_{LDM} = \mathbb{E}_{t\sim[0,1],\mathbf{z}_0\epsilon}\left[||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\psi(\mathbf{z}_t, t)||_2^2\right]. \tag{4.1}$$

Similarly to DDPMs, sampling is performed by first drawing $\mathbf{z}_T$ from a standard Gaussian distribution in the latent space. The reverse process is then applied, fol-

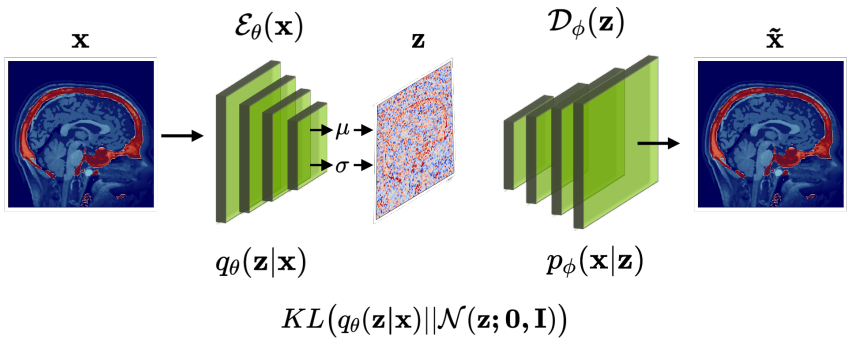


Figure 4.2: Schematics of a standard variational autoencoder, where the latent sample $\mathbf{z}$ is drawn from the latent prior modelled by the probabilistic encoder $\mathcal{E}_\theta(\mathbf{x})$ for an image $\mathbf{x}$, parameterised as a Gaussian with mean μ and standard deviation σ. The latent space is regularised through the Kullback-Leibler (KL) divergence to a standard Gaussian distribution (Equation 2.31). This regularisation, combined with the perceptual compression of the image, stabilises the learned diffusion process in the second stage of LDM training. The latent sample is subsequently mapped back to the data space by the probabilistic decoder $\mathcal{D}_\phi(\mathbf{z})$, which computes the posterior and reconstructs the perceptual details of the image.

lowing Equation 2.91, to iteratively denoise the latent sample (Algorithm 8). Once the latent sample is refined by the diffusion model, it is then passed through the decoder to reconstruct the data in the original space.

Implementations of LDMs in the literature typically employ a U-Net (Ronneberger et al., 2015) backbone for both the first and second stage models, incorporating positional embeddings of timesteps, and residual units with skip connections interleaved with attention mechanisms (Vaswani et al., 2017; Oktay et al., 2018; Ho et al., 2020; Rombach et al., 2022; Vincent, 2011) (Figure 4.4). Residual units play an important role by preserving and transmitting feature maps from the downsampling to the upsampling paths of the network, enhancing gradient flow and mitigating information loss (Ronneberger et al., 2015). The integration of attention mechanisms within the U-Net architecture (Oktay et al., 2018) allows the model to selectively focus on the most relevant features, improving performance on the task.

Overall, the key advantages of LDMs over DDPMs can be summarised as follows:

- **Computational Efficiency**: by operating in the lower-dimensional latent space of a pre-trained autoencoder, LDMs significantly reduce the computational cost associated with both training and sampling.

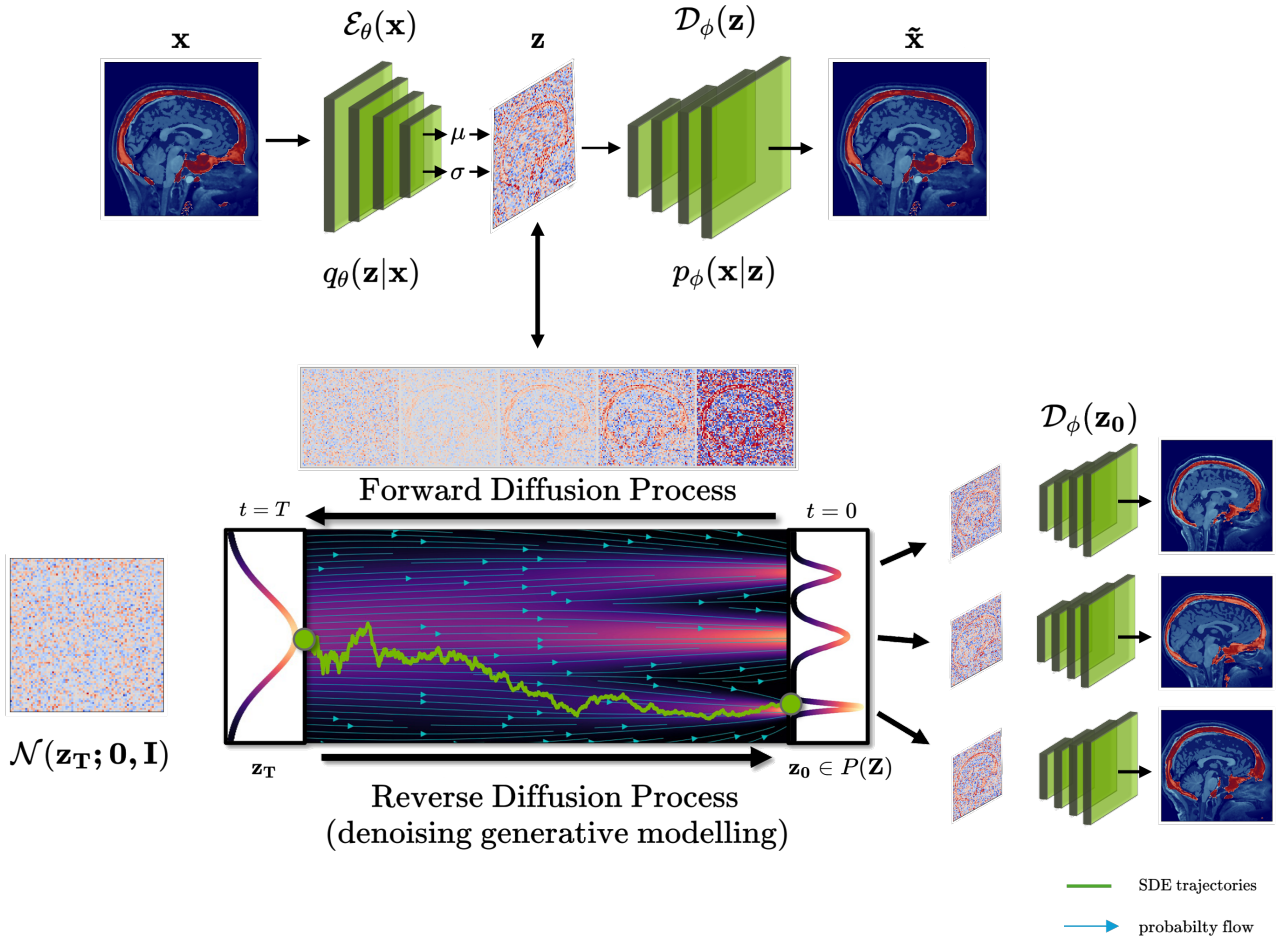


Figure 4.3: Schematics of a latent diffusion model as proposed by Rombach et al. (2022). A sample is encoded to its latent representation by the trained probabilistic encoder, serving as the first sample of the diffusion process chain. The latent sample is then progressively degraded towards Gaussian noise through the forward diffusion process (Equation 2.59), following the stochastic differential (SDE) trajectories. A regularised latent space ensures smoothness in the probability flows that define these trajectories. For image generation, only the reverse diffusion process is used by applying the reverse Markov Chain of the SDE to random gaussian noise (Equation 2.91), which is then fed to the trained decoder for perceptual upsampling towards the data space. Adapted from Kreis et al. (2023).

Algorithm 7 Training Algorithm for LDMs - second stage model

Input: Training data distribution $q(\mathbf{x})$, pretrained encoder $\mathcal{E}_\theta$, noise schedule $\bar{\alpha}_t$
Output: Trained denoising model $\boldsymbol{\epsilon}_\psi$
Initialise model parameters ψ
repeat
  for number of training iterations **do**
    Step 1: Sample data and encode to latent space
    Sample data point $\mathbf{x} \sim q(\mathbf{x})$
    Encode latent representation:

$$\mathbf{z}_0 \leftarrow \mathcal{E}_\theta(\mathbf{x}) \tag{4.2}$$

    Step 2: Corrupt latent sample and compute loss
    Sample diffusion step $t \sim \text{Uniform}(\{1, \ldots, T\})$
    Sample noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

$$\mathcal{L} = \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\psi \left(\sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t \right) \right\|_2^2 \tag{4.3}$$

    Step 3: Update model parameters

$$\psi \leftarrow \psi - \alpha \nabla_\psi \mathcal{L} \tag{4.4}$$

  end for
until convergence of the denoising model

Algorithm 8 Sampling Algorithm for LDMs

Input: Trained denoising model $\boldsymbol{\epsilon}_\psi$, pretrained decoder $\mathcal{D}_\phi$, noise schedule $\alpha_t, \bar{\alpha}_t, \sigma_t$
Output: Generated sample $\mathbf{x}$
Initialise: Sample latent noise $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ **do**
  Sample $\zeta \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else set $\zeta = 0$
  Perform denoising step:

$$\mathbf{z}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{z}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\psi(\mathbf{z}_t, t) \right) + \sigma_t \zeta \tag{4.5}$$

end for
Return: $\mathbf{x} = \mathcal{D}_\phi(\mathbf{z}_0)$ (final generated sample)

- **Stability and High Performance**: by decoupling perceptual decompression from the denoising task, the diffusion process is stabilised, as the latent spaces are low-frequency and regularised, resulting in smoother probability flows along the SDE trajectories.

- **Flexible Modality**: the autoencoder can be trained for a variety of tasks, such as inpainting and super-resolution, offering the flexibility to adapt to different problems without requiring modifications to the diffusion process.

4.2 Dataset and Pre-Processing

The dataset used to train the unconditional diffusion model is the MRI-derived acoustic neuroimaging dataset developed by Strong (2023). This dataset is constructed from the Human Connectome Project (HCP) Young Adult database (Van Essen et al., 2012), which provides high-quality structural T1-weighted and T2-weighted volumetric MRI data for 1,000 subjects. The samples are single-channel with dimensions of 256x320x320.

A notable feature of this dataset is the obfuscation of facial anatomy for each subject, achieved through a normalised anterior filtering process that preserves the integrity of the brain's hard structures and soft tissues while safeguarding patient privacy. The smoothness of this filter effectively prevents sharp edges that could introduce unwanted wave-propagation artifacts during experimental modelling. Randomly selected T1-weighted samples from this dataset are shown in Figure 4.5.

The conversion of MRI data to speed of sound (SoS) data, which parameterises the neuroacoustic images, follows a structured pipeline involving the registration of T1- and T2-weighted MRI modalities, segmentation of tissue categories, and estimation of the speed of sound for each tissue based on a combination of literature sources, as described by Strong (2023). Randomly selected samples from this dataset are shown in Figure 4.6.

For its current application in acoustic wavefield processing, the images are presented submerged in water ($\pm 1500\, m/s$). Due to uncertainties in the literature regarding SoS values for brain tissues, Strong (2023) estimate an error margin of

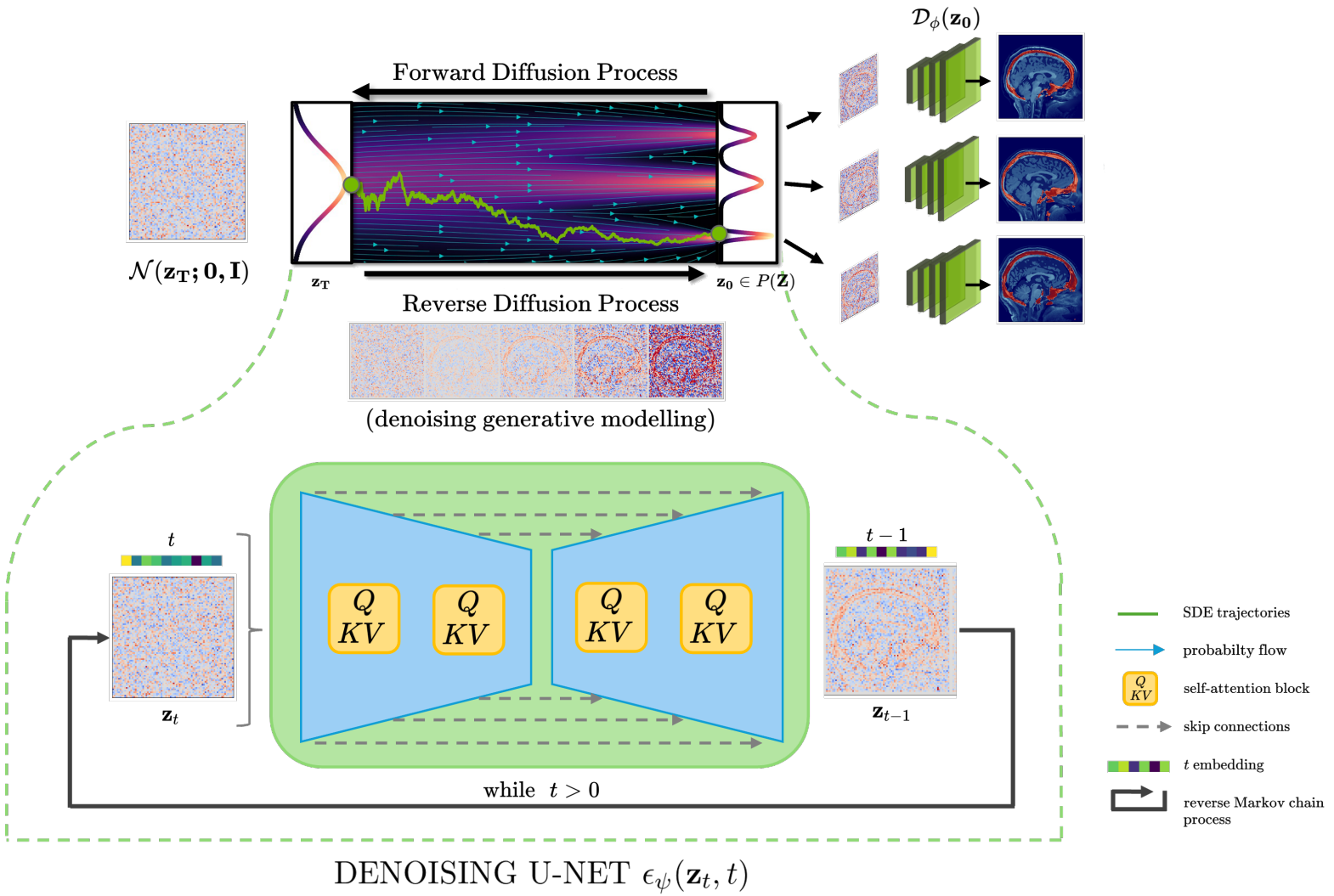


Figure 4.4: Dive into the denoising attention U-Net (Ronneberger et al., 2015) that is trained to predict the reverse diffusion process of the forward SDE. The model takes as input the sample degraded by $\boldsymbol{\epsilon}$ at time step t, defined by the noise scheduler, along with the sinusoidal positional embedding vector of the associated time step. The model then learns to predict the degradation noise $\tilde{\boldsymbol{\epsilon}} = \boldsymbol{\epsilon}_\psi(\mathbf{z}_t, t)$, at each t, by minimising the Euclidean distance $||\boldsymbol{\epsilon} - \tilde{\boldsymbol{\epsilon}}||_2^2$. Once trained, the model's predicted noise is subsequently used to estimate the previous sample of the diffusion process. Figure adapted (Kreis et al., 2023) and (Rombach et al., 2022).

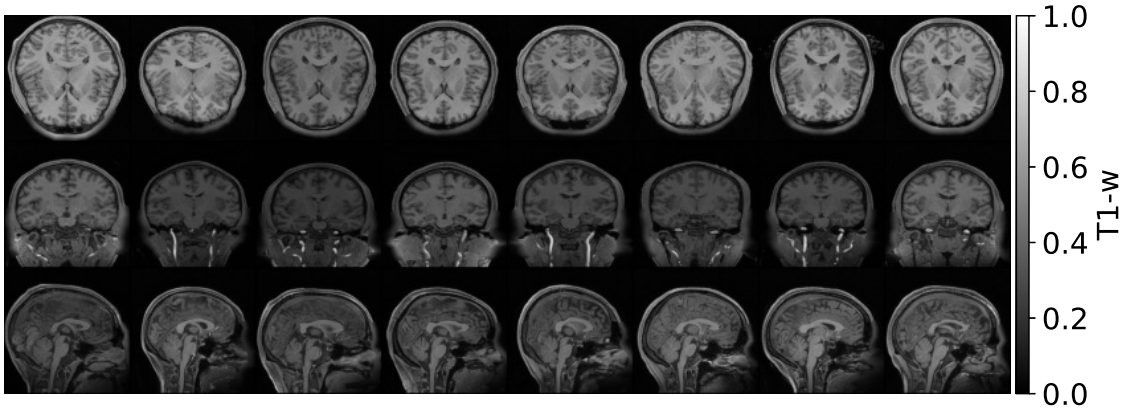


Figure 4.5: T1-weighted MRI volumetric samples from the Human Connectome Project (HCP) Young Adult database (Van Essen et al., 2012), viewed in the mid-plane of the axial (top), coronal (middle), and sagittal (bottom) axes.

$\pm 50\,m/s$. Additionally, as shown in Figure 4.6, structural inaccuracies are sometimes introduced by this pipeline, particularly in the form of incoherent or misaligned skull delineations. For a detailed explanation of the dataset derivation process, readers are referred to Strong (2023). To the best of my knowledge, this dataset represents the only structurally accurate and high-resolution acoustic dataset of the human brain, making it a valuable resource for deep learning research in neuroacoustic imaging. However, it still stands that the dataset is semi-synthetic, relying on a series of assumptions regarding the acoustic properties across different brain tissues, which may not fully capture the true values observed *in vivo*.

For this work, each sample is reduced to 2D by extracting the mid-sagittal and mid-axial planes from the volumetric data. Experiments are conducted independently on each sectional plane, which is resized to 256×256 using an antialiasing filter to preserve structural integrity. The SoS values in this dataset range from $1480\,m/s$ to $3000\,m/s$. The lowest velocities correspond to the water background and the brain's intra-cranial soft tissues, composed of grey and white matter, while the highest velocities are associated with the skull.

Data normalisation is a critical technique in machine learning that mitigates exploding and vanishing gradients while promoting smoother model convergence. By scaling features within a dataset to comparable ranges, normalisation ensures that variables with different units and magnitudes contribute equally during train-

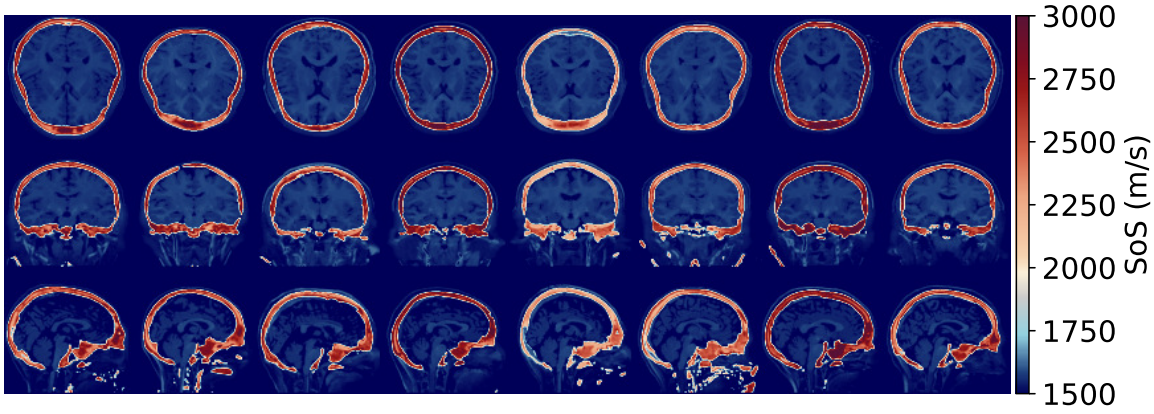


Figure 4.6: Volumetric speed of sound (SoS) samples, measured in m/s, from the dataset released by Strong (2023). Viewed in the mid-plane of the axial (top), coronal (middle), and sagittal (bottom) axes of the brain.

ing. This process prevents features with larger numerical values from overshadowing those with smaller ranges, thereby reducing biases from imbalanced learning and improving the reliability and accuracy of model predictions (Arora, 2024). However, in ultrasound medical imaging, SoS values often deviates from this ideal due to the pronounced disparity between the acoustic properties of soft tissues compared to bone. This discrepancy produces a bimodal distribution with minimal overlap (Figure 4.7), resulting in poorly equalised features.

For this acoustic dataset, careful consideration of data normalisation is essential. The objective is to enhance the contrast between intra-cranial brain regions and the skull to improve feature equalisation while ensuring reversibility for downstream tasks, such as priors or starting models for tomographic inversions. The preprocessing pipeline consists of two steps: first, each data point is divided by a constant of 3000, scaling values to approximately between 0.5 and 1. Next, a logarithmic transformation is applied, followed by the addition of 1, such that the values fall within the range $[0, 1]$. Figure. 4.8 shows the preprocessed samples from Figure 4.6, with the updated distribution in Figure 4.7. Although more advanced reversible normalisation techniques could improve sample contrast, exploring such methods falls outside the scope of this work.

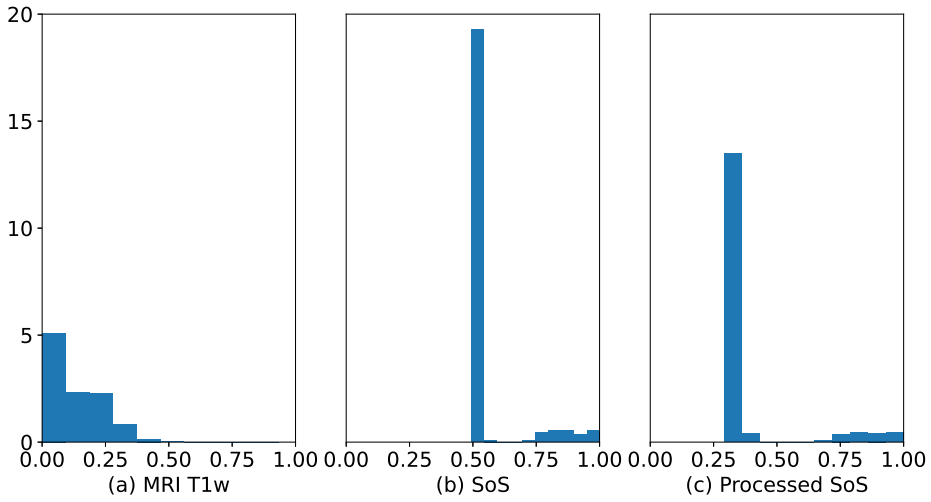


Figure 4.7: Density histograms of 128 volumetric samples of the (a) T1-weighted MRI HPC (Van Essen et al., 2012) dataset, (b) converted and min-max normalised speed of sound dataset by Strong (2023), and (c) the logarithmic processed speed of sound used in this work. The logarithmic processing of the samples is a reversible operation that gently helps equalising the SoS values.

4.3 Results and Discussion

In this section, I present the implementation and results of training an LDM to generate synthetic neuroacoustic images of the brain. Although the training on full 3D volumes was not completed at this point, all experiments were conducted on the mid-plane slices across all principal axes of the brain — axial, coronal, and sagittal — showing feasibility for future 3D expansions. The findings discussed here are applicable across all three slices.

4.3.1 Framework and Implementation

The LDM framework for this thesis is developed using PyTorch (Paszke et al., 2019) due to its flexible and efficient handling of dynamic computation graphs. Central to the implementation is also the Monai Generative Models prototyped package (Cardoso et al., 2022; Pinaya et al., 2023), optimised for generative modelling in medical imaging applications, which simplifies the deployment of latent diffusion models. The framework allows for mixed precision training (Micikevicius et al., 2017) to enhance computational speed and reduce memory usage as well as FlashAttention

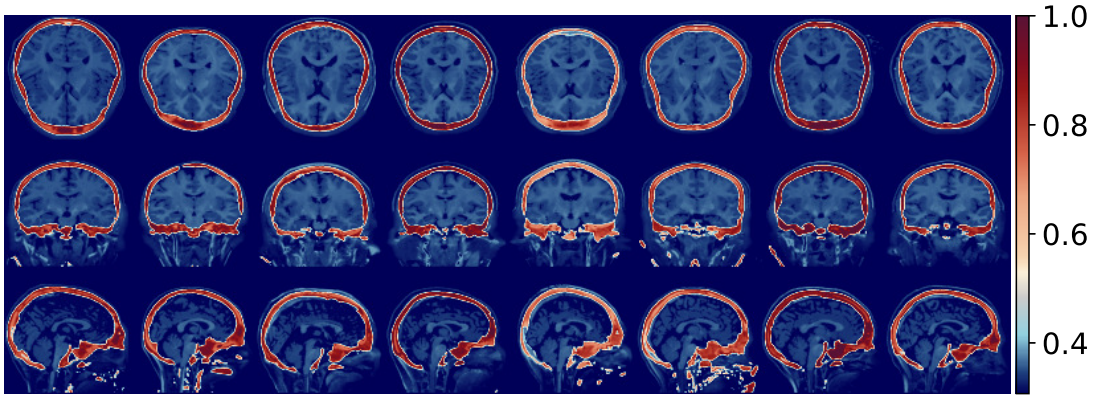


Figure 4.8: Volumetric speed of sound (SoS) samples, measured in m/s, from the dataset released by Strong (2023) after the preprocessing pipeline. Viewed in the mid-plane of the axial (top), coronal (middle), and sagittal (bottom) axes of the brain. The preprocessing gently boosts the values of the intra-cranial tissues while gently decreasing the values of the skull.

(Dao, 2023) to optimise the implementation of attention mechanisms.

Configurability is a fundamental aspect of the developed framework, achieved through a YAML (Ben-Kiki et al., 2009) configuration file that standardises and simplifies the setup for various complex experiments. In the configuration file, users can specify model hyperparameters, optimiser settings, data paths, and other training parameters. An example is shown in Appendix A.1. Features such as data parallelism, automated model and optimiser checkpointing and the ability to resume training are also implemented in order to enhance usability, particularly in shared cluster environments. This is all combined into a command line interface, that takes the following arguments:

```
InverseLDM - training
    usage: train.py  --config CONFIG --name NAME [--logdir LOGDIR]
                     [--comment COMMENT] [--jobid JOBID] [--device DEVICE]
                     [--gpu_ids GPU_IDS] [--seed SEED] [--resume_training]
                     [--overwrite] [-y] [-h][--verbose VERBOSE]

    options:
    -h, --help          show this help message and exit
    --config CONFIG     Path to the config.yml file for training
```

```
    --logdir LOGDIR    Path for saving running related data.
    --name NAME        Name of experiment and also the log folder.
    --comment COMMENT  A string for experiment comment
    --jobid JOBID      Store job ID for HPC systems
    --device DEVICE    Device to which run. Options: cuda or cpu.
                     If cuda, specify gpu_ids with flag
    --gpu_ids GPU_IDS  List of gpu ids to be used in DataParallel.
                     If None, all will be used
    --seed SEED        Random seed for reproducibility
    --resume_training  Whether to resume training. Used in conjunction
                     with checkpoint flag in yaml config. If checkpoint not
                     specified, will use latest model.
    --overwrite        Whether to overwrite experiment of same name
    -y                 Automatic 'yes' to user input questions
    --verbose VERBOSE  Verbose level: info | debug | warning | critical


InverseLDM - sampling
    usage: sample.py  --config CONFIG --name NAME [--logdir LOGDIR]
                      [--jobid JOBID] [--device DEVICE] [--gpu_ids GPU_IDS]
                      [-h][--verbose VERBOSE]

    options:
    -h, --help         show this help message and exit
    --config CONFIG    Path to the config.yml file for sampling
    --logdir LOGDIR    Path for retrieving models and saving samples.
    --name NAME        Name of experiment to load and also the log folder.
    --jobid JOBID      Store job ID for hpc systems
    --device DEVICE    Device to which run. Options: cuda or cpu.
                        If cuda, specify gpu_ids with flag
    --gpu_ids GPU_IDS  List of gpu ids to be used in DataParallel.
                        If None, all will be used
    --verbose VERBOSE  Verbose level: info | debug | warning | critical
```

The framework includes a specialised dataset class tailored for preprocessing

2D SoS and MRI images (Appendix A.2), ensuring data is optimally prepared for processing. It also allows for easy integration of user-created datasets, mirroring functionality found in the original Pix2Pix implementation (Isola et al., 2017), thereby expanding its utility across different imaging tasks. Custom loggers can be developed to suit specific post-processing needs, with integrated logging capabilities via platforms like TensorBoard (Abadi et al., 2015) and a more standard JSON (Pezoa et al., 2016) format (Appendix A.3). The entire framework, named `InverseLDM`, is openly accessible on GitHub at `https://github.com/dpelacani/InverseLDM` to promote transparency and community collaboration.

The Latent Diffusion Models (LDMs) presented in this work were trained on a single NVIDIA A100-SXM4 GPU with 40GB of dedicated memory. At a resolution of 256x256, the LDMs described later in this chapter required approximately two hours to train, utilising around 35GB of memory, reflecting the demands of the model size, attention layers, and batch configurations. However, 3D experiments could not be conducted given the available resources at the time. Unexplored solutions to this limitation include the use of model parallelism, activation and gradient checkpointing, and the adoption of more memory-efficient deep learning frameworks like Julia's Flux (Innes, 2018). Such techniques could be particularly advantageous for managing 3D attention mechanisms, where memory constraints represent a primary bottleneck.

In contrast, the inference and sampling processes of the LDM, once trained, demand substantially less computational power and memory. The high-quality samples presented in this section were generated using a single NVIDIA GeForce RTX 2080-Ti GPU with 11GB of memory, completing at near-real time.

4.3.2 Structural Recovery with Perceptual Losses

The quality of the LDM autoencoder is a critical determinant of latent diffusion model performance, directly impacting both the resolution of generated samples and the stability of model convergence. Since the diffusion process in LDMs unfolds within the semantically compressed latent space $\mathbf{z}$, the decoder effectively acts as an upper bound on the achievable quality of image synthesis. The success of LDMs in high-resolution image generation is largely attributable to the quality of the autoen-

coder.

In the foundational work by Rombach et al. (2022), the first-stage autoencoder was trained using a combination of perceptual loss (Zhang et al., 2018a), a patch-based adversarial objective (Dosovitskiy and Brox, 2016; Yu et al., 2021; Esser et al., 2021; Isola et al., 2017), and traditional pixel-space losses such as MAE and MSE. This multi-faceted training approach ensures the decoder can accurately reconstruct perceptual features from the latent space, preserving fidelity in image reconstructions.

Rombach et al. (2022) also conducted a detailed comparison across different compression levels, underscoring the trade-off between computational efficiency and synthesis quality. Their findings indicate that preserving structural integrity requires a spatial latent space (rather than a vectorised one) and moderate compression or regularisation to avoid compromising image fidelity. For a comprehensive review of this analysis, readers are encouraged to consult the original paper.

Here, I evaluate four distinct autoencoder models trained with varying loss functions and latent space dimensions, focusing on their reconstruction quality and the characteristics of their respective latent spaces. All models are based on a U-Net architecture (Ronneberger et al., 2015) with two residual units, incorporating attention mechanisms solely in the final layer due to memory constraints. To ensure consistent evaluation, all models are trained using an MSE reconstruction loss and a KL divergence weight of 1×10^{-6}. The Adam optimiser (Kingma and Ba, 2014) is employed, with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and a weight decay of 1×10^{-4} to promote sparsity and improve generalisation. The SiLU (Elfwing et al., 2018) activation function is used throughout the model, consistent with modern diffusion model architectures (Rombach et al., 2022; Ho et al., 2020). Table 4.1 summarises the key hyperparameters for each model, while Figure 4.9 provides examples of their reconstructions.

The first model (VAE-1) prioritises reconstruction accuracy and latent regularisation, serving as a baseline for subsequent experiments. In the absence of high-resolution loss functions, the reconstructions produced by this model are structurally consistent with the input data but are limited to low-to-mid frequency resolutions.

	VAE-1	VAE-2	VAE-3	VAE-4
model_size	3M	3M	4M	3M
num_iter	5K	5K	5K	5K
num_res_blocks	2	2	2	2
num_channels	[64, 128, 128]	[64, 128, 128]	[32, 64, 128, 128]	[64, 128, 128]
attention_levels	[3]	[3]	[4]	[3]
z_shape	(3, 64, 64)	(3, 64, 64)	(1, 32, 32)	(3, 64, 64)
recon_loss	$\mathcal{L}_1$	$\mathcal{L}_1$	$\mathcal{L}_1$	$\mathcal{L}_1$
recon_weight	1.0	1.0	1.0	1.0
kl_weight	1.00×10^{-6}	1.00×10^{-6}	1.00×10^{-6}	1.00×10^{-6}
perceptual_loss	-	-	LPIPS (Zhang et al., 2018a)	LPIPS (Zhang et al., 2018a)
perceptual_weight	-	-	1.0	1.0
lpips_model	-	-	VGG-16 (Simonyan and Zisserman, 2014)	VGG-16 (Simonyan and Zisserman, 2014)
wiener_mode	-	reverse	-	-
wiener_filter_dim	-	2	-	-
wiener_filter_scale	-	2	-	-
wiener_epsilon	-	2.0	-	-
wiener_laplace_alpha	-	-1×10^{-4}	-	-
wiener_laplace_beta	-	5.0	-	-
wiener_weight	-	1.0	-	-
adversarial_mode	-	-	LS-GAN (Mao et al., 2017)	LS-GAN (Mao et al., 2017)
disc_n_layers	-	-	2	2
disc_feature_channels	-	-	16	16
adversarial_weight	-	-	0.1	0.1
batch_size	32	32	32	32
optimiser	Adam (Kingma and Ba, 2014)	Adam (Kingma and Ba, 2014)	Adam (Kingma and Ba, 2014)	Adam (Kingma and Ba, 2014)
weight_decay	1.00×10^{-4}	1.00×10^{-4}	1.00×10^{-4}	1.00×10^{-4}
lr	1.00×10^{-4}	1.00×10^{-4}	1.00×10^{-4}	1.00×10^{-4}
d_lr	5.00×10^{-4}	5.00×10^{-4}	5.00×10^{-4}	5.00×10^{-4}
adam_betas	[0.9, 0.999]	[0.9, 0.999]	[0.9, 0.999]	[0.9, 0.999]

Table 4.1: Summary of hyperparameters that describe the four VAE models explored in this work for perceptual compression.

The second model (VAE-2) incorporates the Wiener Loss, introduced in Chapter 3, to enhance the recovery of high-frequency details while maintaining low computational costs. Due to the inherent properties of the Wiener Loss, as outlined in Section 3, its effectiveness is reduced in scenarios with high-contrast structures. This is because the filter values are proportional to the data amplitudes, allowing the zero-lag matching condition to be satisfied by disregarding low-intensity regions

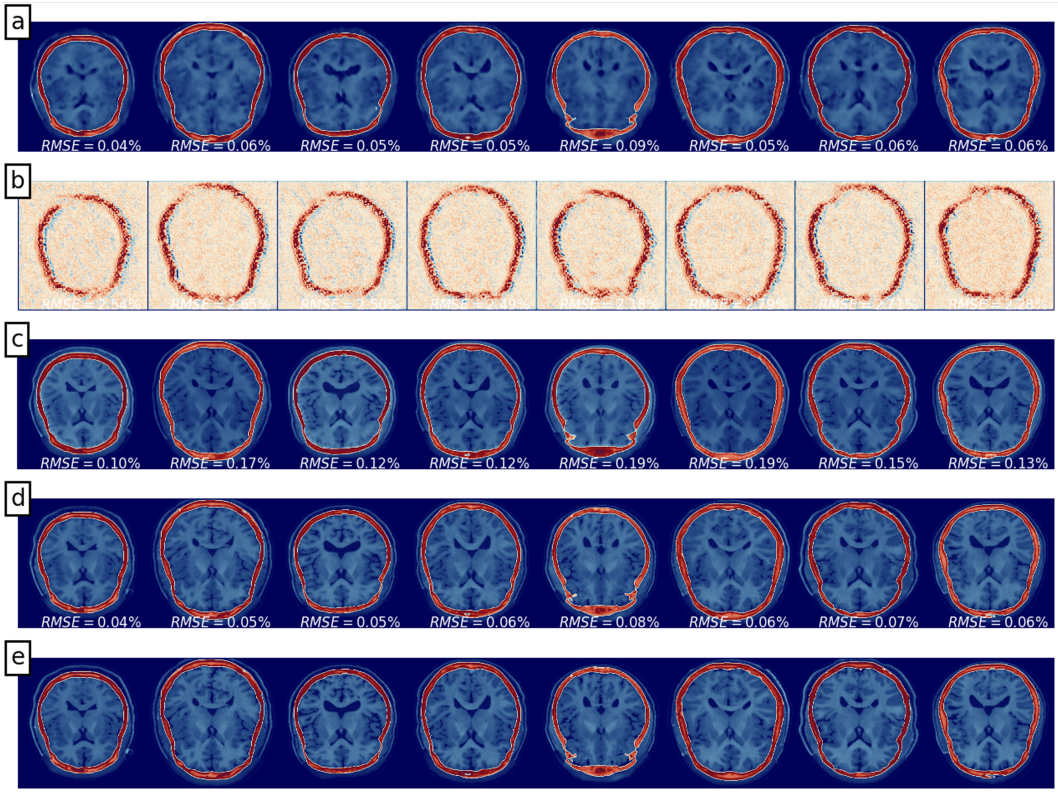


Figure 4.9: Reconstruction of axial slices with associated root mean square errors (RMSE) from (a) VAE-1, (b) VAE-2, (c) VAE-3, (d) VAE-4 and (e) is the ground truth. Reconstructions from VAE-1 lack high-frequency details as no perceptual losses are integrated durings training. Reconstructions from VAE-2 fail to converge due to the Wiener loss' inability to fit the data at high value contrasts; reconstructions from VAE-3 have high perceptuality but the tight latent space bottleneck negatively impacts the fidelity of the images; reconstructions from VAE-4 are the most accurate in the semantic and perceptual recoveries.

of the image. In the context of neuroacoustic imaging, the model predominantly converges on regions representing the skull but fails to reconstruct other meaningful structures. Notably, the Wiener Loss applied to this model predates the stabilisation improvements introduced in Chapter 3. While it is possible that improved stabilisation parameters could yield better results, the underlying challenges associated with Wiener Loss in high-contrast settings remain pertinent.

The third and fourth models (VAE-3 and VAE-4) introduce perceptual and adversarial losses to the autoencoder training process, augmenting the reconstruction and KL divergence losses used in the baseline model. VAE-3 is a larger model with a compact latent space that is eight times smaller than the original input, while VAE-4 features a 3-channel latent space that is four times smaller than the input

dimensions.

The adversarial loss is implemented by co-training a discriminator network alongside the VAE, following the Least Squares Generative Adversarial Network (LS-GAN) scheme (Mao et al., 2017), which challenges the VAE by distinguishing between real and synthetic samples. The discriminator is a simple convolutional neural network with 16 feature channels, as detailed in Table 4.1. The adversarial network is optimised using the same Adam optimiser (Kingma and Ba, 2014) and hyperparameters as those applied to train the VAE.

The perceptual loss in these models is quantified using the Learned Perceptual Image Patch Similarity (LPIPS) metric (Zhang et al., 2018a), leveraging features extracted from a VGG-16 network (Simonyan and Zisserman, 2014) pre-trained on ImageNet (Deng et al., 2009). LPIPS loss is computed by measuring the Euclidean distance between the feature maps of input and reconstructed images, capturing high-level structural aspects such as patterns and textures (see Section 2.1.2). Although the VGG-16 model is originally trained on ImageNet to extract abstract features from natural images, its feature extraction capabilities extend to medical imaging applications. Figure 4.10 illustrates this with an example feature map extracted from the pre-trained VGG-16 for a neuroacoustic image. The feature map highlights high-frequency details of soft tissues and delineates both intra-cranial and outer skull structures, demonstrating the network's ability in promoting the recovery of fine structural details of this dataset within VAE reconstructions.

With the integration of perceptual and adversarial losses, VAE-3 and VAE-4 demonstrate improved capacity to capture high-frequency structures in image reconstructions. However, the constrained latent space of VAE-3 leads to a noticeable reduction in fidelity, reflected by higher root mean squared errors (RMSEs) relative to the original images. The performance of these networks can be further analysed through the resolution of their latent spaces, as shown in Figure 4.11.

VAE-4's latent space more effectively preserves finer anatomical structures, including skin layers, ventricles, and dura mater regions, compared to the other models. Additionally, the delineation of the skull is significantly more prominent and accurate in VAE-4's latent space than in any other model. In contrast, VAE-3's latent

space lacks the capacity to retain such detailed features, while VAE-2's latent space only loosely captures the general shape of the skull, failing to retain the clarity of other anatomical structures.

4.3.3 Generative Modelling

For the generative modelling, I conducted a series of experiments with LDM models based on the VAE architectures outlined in Table 4.1, alongside an additional investigation into the impact of excluding attention mechanisms from an LDM. Each LDM, like the VAE, utilises a U-Net architecture (Ronneberger et al., 2015) with two residual blocks per layer. These models were constructed with four layers of

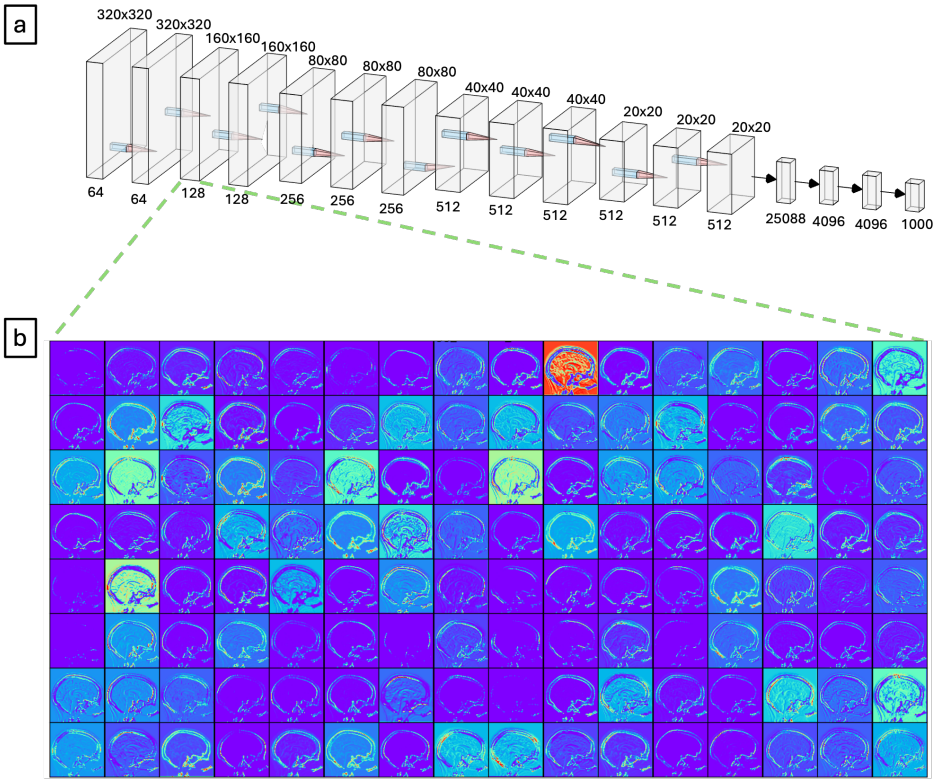


Figure 4.10: Extracted feature maps (b) of a single neuroacoustic image sample of the brain from the VGG-16 (Simonyan and Zisserman, 2014) network (a) pretrained on ImageNet (Deng et al., 2009) dataset of 1,000 classes of natural images. Despite the difference in dataset nature, the pre-trained network is able to select various useful features of the image, highlighting high-frequency textures between the soft tissues and isolating outer and intra-cranial structures. The choice of visualisation of the mid-sagittal plane is arbitrary, and my experiments show that the same principles hold for the mid-axial and mid-coronal planes. Neural network visualisation made with (Lenail, 2024). Figure inspired by Strong (2023).

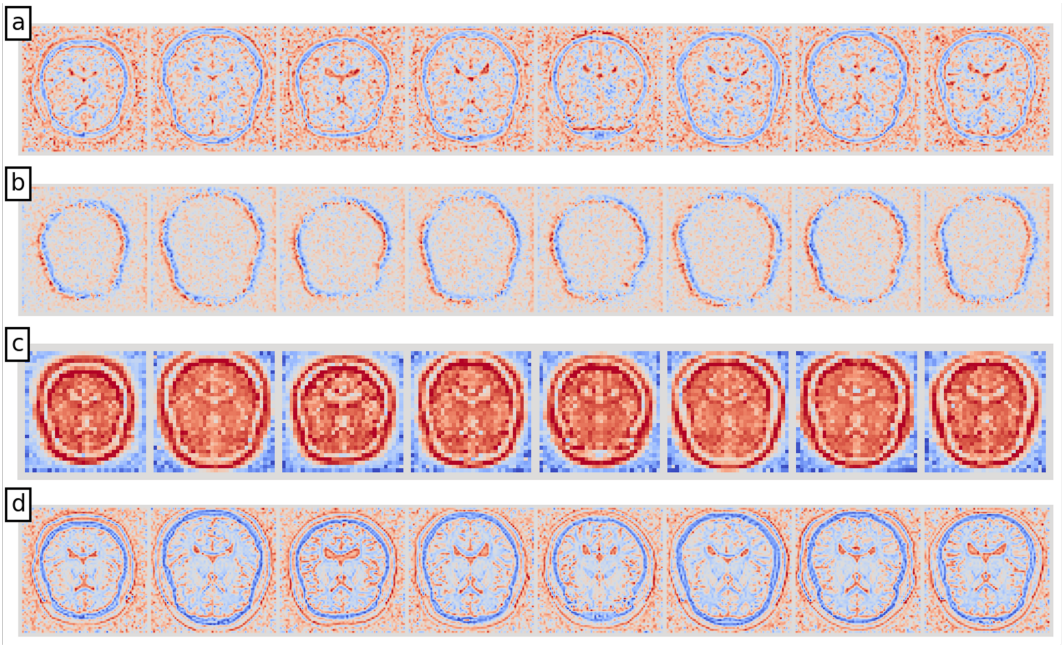


Figure 4.11: Normalised latent spaces of VAE-1 (a), VAE-2 (b), VAE-3 (c) and VAE-4 (d) as described in Table 4.1 and with associated reconstructions in Figure 4.9. For the multi-channel latent spaces (a, b and d) the presented visualisation is averaged in the channel dimension.

increasing channel depth — 64, 128, 256, and 512 — resulting in an architecture substantially larger than that of their corresponding VAEs.

For the diffusion process, hyperparameters were selected to closely align with those used by Rombach et al. (2022). A linear noise schedule was applied to the parameter β, as described in Equations 2.59 and 2.91, where $\alpha := 1 - \beta_t$. The schedule began at 0.015 and increased to 0.0205 over 1,000 steps. MSE loss was employed as the reconstruction objective to quantify the discrepancy between the injected noise and the noise predicted by the network.

The optimisation of the LDMs followed a similar configuration to VAE training, using the Adam optimiser with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, a learning rate of 2×10^{-5}, and a batch size of 32. Weight decay, set to 1×10^{-4}, was applied exclusively to non-bias terms to promote sparsity and improve generalisation, and SiLU (Elfwing et al., 2018) was chosen as the activation function. Three attention layers, each with eight heads, were integrated into all but the first layer of the models, except for the experiment where attention layers were omitted (LDM-4). Each model underwent 25,000 training iterations. A summary of these model configurations is provided in

Table 4.2.

	LDM-1	LDM-2	LDM-3	LDM-4	LDM-5
vae_model	VAE-1	VAE-2	VAE-3	VAE-4	VAE-4
num_res_blocks	2	2	2	2	2
num_channels	[64, 128, 256, 512]	[64, 128, 256, 512]	[64, 128, 256, 512]	[64, 128, 256, 512]	[64, 128, 256, 512]
model_size	63M	63M	63M	57M	63M
attention_levels	[2, 3, 4]	[2, 3, 4]	[2, 3, 4]	-	[2, 3, 4]
num_head_channels	8	8	8	-	8
transformer_num_layers	3	3	3	-	3
recon_loss	$\mathcal{L}_2$	$\mathcal{L}_2$	$\mathcal{L}_2$	$\mathcal{L}_2$	$\mathcal{L}_2$
beta_start	0.0015	0.0015	0.0015	0.0015	0.0015
beta_end	0.0205	0.0205	0.0205	0.0205	0.0205
num_train_timesteps	1000	1000	1000	1000	1000
schedule	linear	linear	linear	linear	linear
num_iter	25K	25K	25K	25K	25K
optimiser	Adam (Kingma and Ba, 2014)	Adam (Kingma and Ba, 2014)	Adam (Kingma and Ba, 2014)	Adam (Kingma and Ba, 2014)	Adam (Kingma and Ba, 2014)
weight_decay	1×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}
lr	2.00×10^{-5}	2.00×10^{-5}	2.00×10^{-5}	2.00×10^{-5}	2.00×10^{-5}
batch_size	32	32	32	32	32
adam_betas	[0.9, 0.999]	[0.9, 0.999]	[0.9, 0.999]	[0.9, 0.999]	[0.9, 0.999]

Table 4.2: Summary of hyperparameters that describe the five LDM models explored in this work.

To evaluate the overall quality and diversity of the generated samples, two widely recognised metrics in the deep generative community are examined: Fréchet Inception Distance (FID) (Heusel et al., 2017) and Kernel Inception Distance (KID) (Bińkowski et al., 2018). Both metrics utilise deep features extracted from real and generated images to assess sample quality and diversity, in a manner similar to LPIPS. KID computes the maximum mean discrepancy between feature distributions using polynomial kernel fitting, while FID measures the Wasserstein-2 distance between Gaussian distributions fitted to the extracted features. FID has been shown to correlate strongly with human perceptual evaluation (Heusel et al., 2017; Borji, 2022), demonstrating sensitivity to visual artifacts such as blurriness. However, FID is known to exhibit bias in the estimation of expected values when the sample size is small (Chong and Forsyth, 2020). In contrast, KID, based on polynomial kernel fitting, is less affected by limited sample counts, providing a more reliable estimate

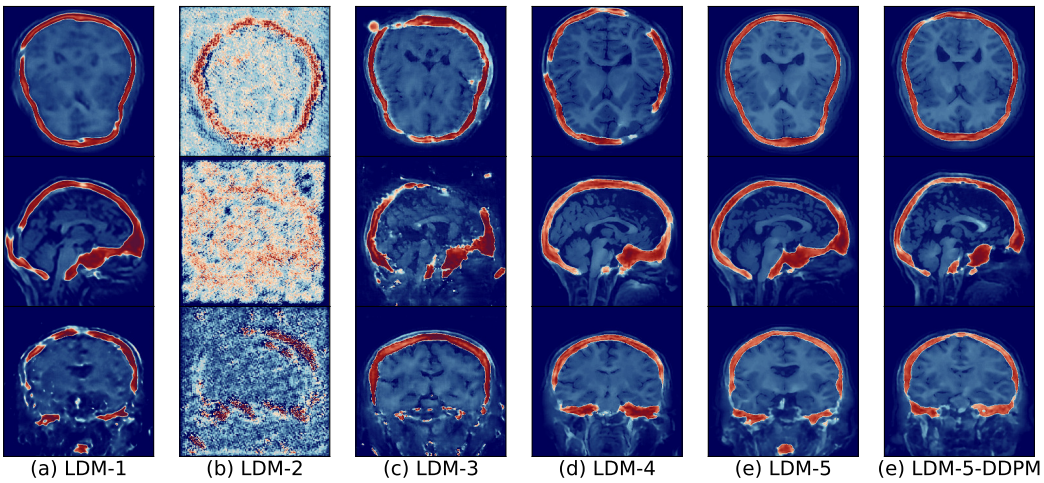


Figure 4.12: Example of samples generated from the five different LDM models (columns) explored in this work in the mid-plane of the axial (top row), sagittal (middle row), and coronal (bottom row) axes. All models were samples with a DDIM scheduler at 25 steps, except for (e) where the samples were generated with a DDPM scheduler at 1,000 steps. While LDM-1 to 4 fail to generate structurally accurate brains, LDM-5, sampled with either DDIM or DDPM are able to construct realistic-looking neuroacoustic brain samples.

in such cases. A detailed discussion of these metrics is presented in Section 2.1.8.

In addition to FID scores computed from pre-trained ImageNet (Deng et al., 2009) features, I present FID scores derived from RadImageNet (Mei et al., 0), a comprehensive multi-modal medical imaging dataset with pre-trained features. The latter FID calculations are expected to more accurately reflect image quality due to the closer alignment between the dataset's domain and the generated samples. The evaluation of these metrics across various LDM models and for all three mid-plane axes of the brain is summarised in Table 4.3, with generated samples shown in Figure 4.12 for a qualitative comparison.

Optimal results are achieved with LDM-5, where perceptual losses are strategically applied during VAE training, and the latent space is sufficiently large to capture the complex anatomical features of brain images. This configuration produced the lowest FID scores, indicating that the synthetic images generated are the most perceptually faithful and anatomically plausible among the experiments.

In contrast, LDM-2, which employed Wiener Loss during VAE training, exhibited the weakest performance across all metrics, failing to generate realistic-looking samples that reflect the underlying data distribution. This outcome is expected, as

	FID ↓ (ImageNet)	FID ↓ (RadImageNet)	KID↓	N_{steps}
LDM-1 (axial)	175.62	8.53	0.28±0.01	25
LDM-2 (axial)	438.19	33.44	0.74±0.01	25
LDM-3 (axial)	306.64	25.77	0.51±0.01	25
LDM-4 (axial)	92.89	5.82	0.13±0.01	25
LDM-5 (axial)	42.66	5.04	0.05±0.00	25
LDM-5-MRI (axial)	86.29	5.84	0.12±0.01	25
LDM-5-DDPM (axial)	**41.66**	**4.36**	**0.04±0.00**	1000
LDM-1 (sagittal)	120.78	6.27	0.13±0.01	25
LDM-2 (sagittal)	416.93	24.17	0.68±0.01	25
LDM-3 (sagittal)	329.32	17.21	0.49±0.01	25
LDM-4 (sagittal)	102.41	4.30	0.12±0.01	25
LDM-5 (sagittal)	76.64	3.12	**0.07±0.01**	25
LDM-5-MRI (sagittal)	119.28	5.12	0.15±0.01	25
LDM-5-DDPM (sagittal)	**76.08**	**3.10**	**0.07±0.01**	1000
LDM-1 (coronal)	197.43	7.60	0.24±0.01	25
LDM-2 (coronal)	481.45	29.91	0.78±0.01	25
LDM-3 (coronal)	367.84	37.92	0.59±0.01	25
LDM-4 (coronal)	128.18	6.51	0.14±0.01	25
LDM-5 (coronal)	99.67	5.75	**0.11±0.01**	25
LDM-5-MRI (coronal)	174.62	8.12	0.23±0.01	25
LDM-5-DDPM (coronal)	**99.19**	**5.29**	**0.11±0.01**	1000

Table 4.3: Fréchet Inception Distance (FID) (Heusel et al., 2017) and Kernel Inception Distance (KID) (Bińkowski et al., 2018) metrics over 128 generated samples of each experiment, compared to 1,000 samples of the original dataset. All metrics performed using features of a pre-trained InceptionV3 (Szegedy et al., 2016) network architecture, consistent with current common practices in generative deep learning. Sampling was obtained with the DDIM equations for reverse Markov chain for 25 steps, except for LDM-5-DDPM where the DDPM equations were used with 1,000 reverse steps. FID (ImageNet) and KID computed with the default parameters in `torch-fidelity` (Obukhov et al., 2020). FID (RadImageNet) computed with Monai Generative (Pinaya et al., 2023).

the latent spaces in LDM-2 do not provide faithful representations of the input samples, limiting the decoder's ability to accurately reconstruct fine details, as shown in Figures 4.9 and 4.11.

Similarly, LDM-1's omission of perceptual and high-frequency losses led to significantly higher FID and KID scores. The limited capacity of the latent space results in anatomical inaccuracies, most notably a lack of continuity in the skull. This issue is also evident in LDM-3, where the latent space size is highly constrained, further limiting the model's ability to capture detailed structural information. Additionally,

LDM-1's scores are adversely affected by image blurriness and a lack of diversity due to poor perceptual quality in the generated samples.

The higher FID and KID scores observed in LDM-4 indicate lower fidelity and reduced diversity in the generated images, attributed to the absence of attention mechanisms. Altogether, these findings underscore the critical role of properly configured latent spaces and the integration of perceptual losses to ensure the generation of more anatomically plausible, diverse, and clinically relevant samples.

However, while FID and KID scores offer a useful indication of distributional similarity between real and synthetic data, the challenge of reliably distinguishing "good" from "bad" samples remains a persistent issue across various scientific domains that utilise generative AI. This difficulty is specially important in medical applications, where ethical considerations and patient safety render the evaluation of generative outputs both critical and exacting. At present, no universally accepted quantitative method exists to determine whether synthetic medical images are anatomically accurate in a clinically meaningful way. Such assessments would require detailed validation by trained medical professionals, as synthetic images that appear anatomically plausible to an untrained observer may still contain clinically significant inaccuracies. In this work, no resources were allocated, nor was it within the scope or intention, to perform such expert qualitative validation. This represents a key limitation in the validation of the model's outputs.

DDIM vs DDPM

The samples used to evaluate the models in Table 4.3 and shown in Figure 4.12 were generated using DDIM sampling (Equation 2.111) with 25 reverse diffusion steps, except for LDM-5-DDPM, which utilised the DDPM reverse process (Equation 2.91) with 1,000 steps. As discussed in Section 2.1.5, the implicit and deterministic nature of DDIM enables a significant reduction in the number of sampling steps required to achieve high-quality image synthesis, whereas DDPM relies on the full set of steps to attain comparable results.

The decision to use 25 timesteps in these experiments was based on a preliminary analysis of sample quality relative to the number of sampling steps, as

summarised in Figure 4.13. For the models evaluated, 10 to 30 sampling steps were sufficient to achieve optimal FID scores with DDIM, whereas DDPM required a minimum of 700 steps. A comparison of metric scores between DDIM and DDPM sampling for LDM-5 in Table 4.3 revealed only marginal differences. However, as shown in Figure 4.13, the computational cost of sampling increases linearly with the number of timesteps. Visual comparisons of sampling results for LDM-5 using both DDIM and DDPM are presented in Figure 4.14, further highlighting the flexibility of DDIM over DDPM. The highest-quality image synthesis in this work was achieved with the LDM-5 architecture using 25 DDIM sampling steps.

4.4 Future Work

A larger batch of finalised samples from LDM-5, generated using DDIM with 25 steps, is presented for the mid-plane axes in Figures 4.15, 4.16, and 4.17. These represent the highest-quality results obtained in the course of this thesis. While the majority of generated samples exhibit anatomical accuracy, a subset displays visible errors, particularly in the continuity of the skull. These structural misalignments likely arise from a combination of dataset biases and limitations in model performance.

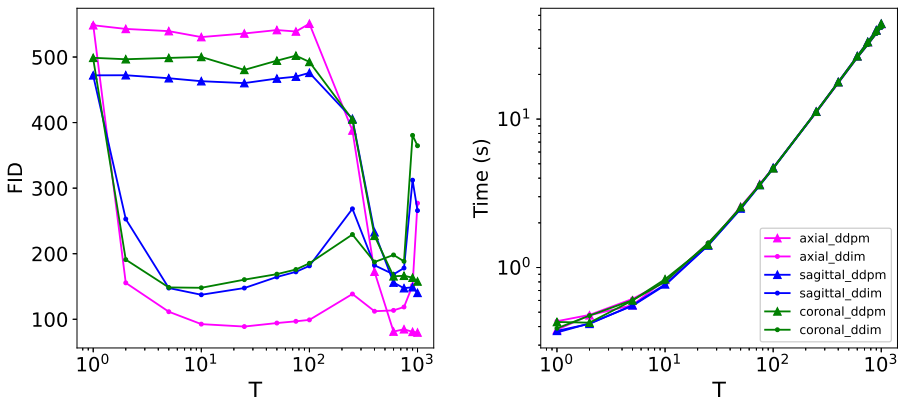


Figure 4.13: (a) FID and (b) sampling time (b) per number of sampling steps averaged for 8 samples generated from LDM-5 using DDPM (△) and DDIM (○). Colors represent different anatomical mid-planes. While DDPM reaches best metrics at >750 (∼30s) sampling steps, DDIM is able to achieve similar metrics starting at 10 (∼0.6s) sampling steps, showing exceptionally better efficiency. Samples were generated with a single NVIDIA GeForce RTX 2080 Ti 11 Gb card.

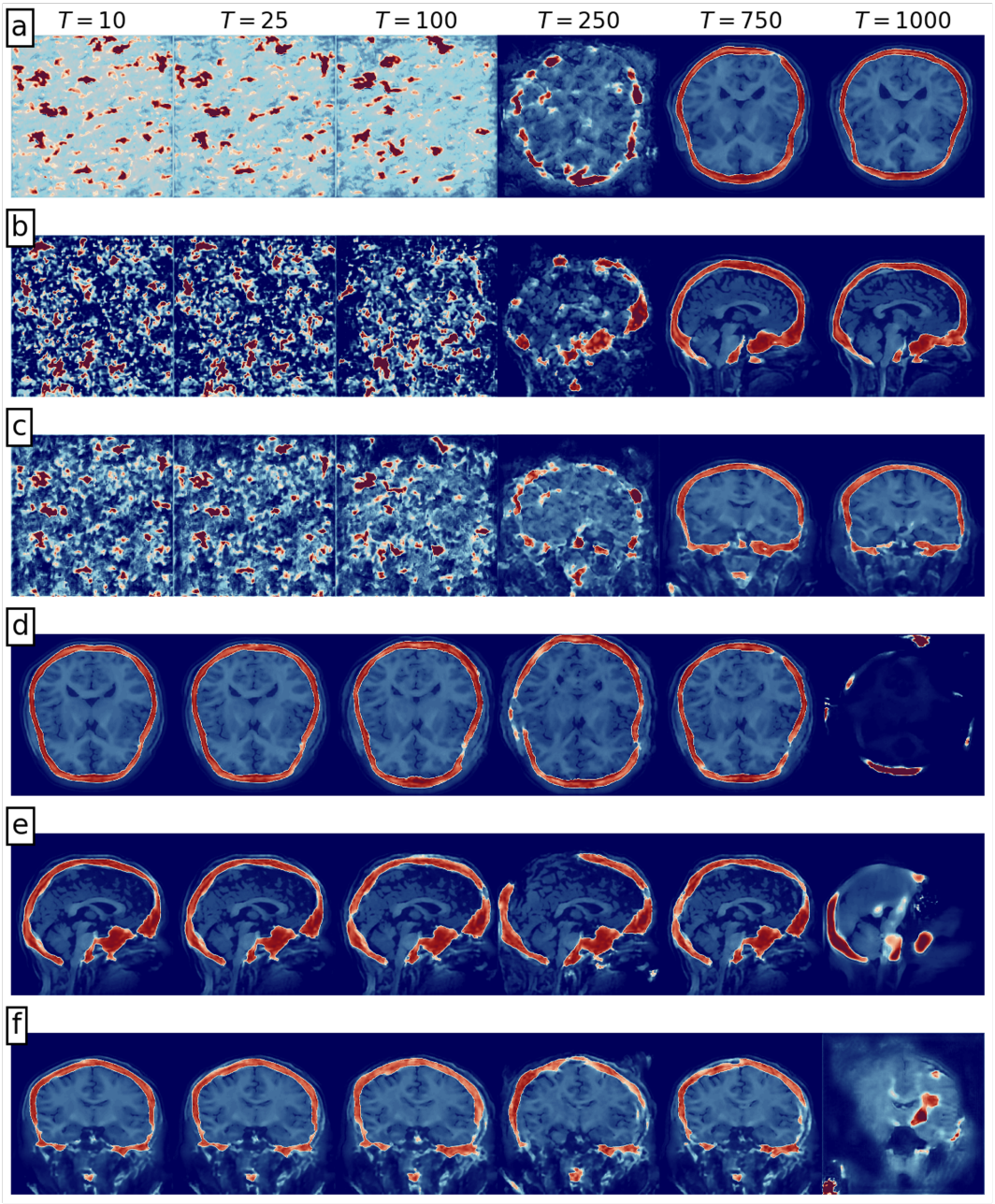


Figure 4.14: Samples generated from LDM-5 using DDPM (a-c) and DDIM (d-f) at different number of sampling steps (columns). While sampling with DDPM can only synthetise high-quality samples at >750 steps, sampling with DDIM achieves the equivalent quality with <30 steps.

To uphold ethical standards and ensure the dataset's utility for medical research, it is essential to establish a rigorous quality-check procedure to identify and exclude erroneous samples prior to public distribution.

Experimenting with untested hyperparameters may further improve the consistency of the generated samples. For example, alternative noise schedulers have been shown to regularise the diffusion sampling process and improve the fidelity of generated distributions (Nichol and Dhariwal, 2021; Sabour et al., 2024). In addition, given the significant improvements in image quality observed with attention-based architectures, I speculate that larger networks with additional attention mechanisms could yield more consistent results, although memory constraints prevented exploration in this direction. Advanced techniques such as diffusion refinement (Podell et al., 2023), weight tracking via exponential moving averages (Karras et al., 2024), and adversarial diffusion models (Sauer et al., 2025) may also improve the models' capacity to generate high-quality and reliable images.

To further enhance the diversity of generated samples, augmenting the dataset with small, localised elastic transformations could facilitate the creation of a broader range of realistic brain images, thereby improving overall model performance. Additionally, conditioning the models on image metadata — such as sex, age, race, and brain volume — as demonstrated by Pinaya et al. (2022), could significantly advance medical research by enabling the generation of more targeted and representative image data.

Finally, extending the current LDMs to generate 3D images represents a natural progression of this work, holding significant potential to advance deep learning research in brain ultrasound imaging and enhance downstream tasks (Zhu et al., 2025). The demonstrated consistency of the 2D models across all principal brain axes, coupled with the successful implementation by Pinaya et al. (2022) for volumetric MRI images, provides strong evidence that 3D neuroacoustic image generation using LDMs is achievable. However, this expansion would require careful consideration of the heightened memory demands associated with 3D latent diffusion models. One way to mitigate such computational constraints is to leverage pre-trained 2D models as foundational components within ensemble-based 3D architectures, as

demonstrated by Zhu et al. (2025).

4.5 Summary

In this chapter, I demonstrated that an unconditional latent diffusion model (LDM) can generate high-quality, diverse, and statistically conforming samples from a limited dataset of neuroacoustic images. Given the challenges in acquiring high-quality ultrasound images of the brain — where costs and technical complexities, such as the non-linearity of wavefield inversion, present great barriers — the results of this work have the potential to drive further advancements in medical imaging research.

To facilitate testing and further development of these LDMs, I presented an open-source framework integrating fast, specialised deep-learning libraries and a highly customizable API. This setup enabled efficient evaluation of four distinct variational autoencoders and five LDMs across various training regimes, underscoring the importance of well-represented and well-regularised latent spaces as essential precursors to stable image generation.

The high contrast of values within neuroacoustic images posed a great challenge, requiring meticulous network designs to accurately capture the complexities of the brain's intra-cranial regions, which are represented by lower numerical values compared to the skull. The proposed preprocessing method aimed to mitigate this imbalance, though more refined techniques could yield substantial improvements to the current implementation. Additionally, further exploration of diffusion hyperparameters and the adoption of more advanced and refined LDM architectures could enhance the overall quality and fidelity of the generated samples.

By extending this research across the three principal axes of the brain, this work demonstrated the feasibility of advancing towards 3D image synthesis. However, achieving this would require further optimisation ensure computational practicality. Additionally, further experimentation is required to evaluate the integrity of the LDM in accurately reproducing pathologies when trained on datasets that include unhealthy brain samples.

Finally, while the LDM's efficient sampling capabilities enable the rapid gener-

ation of a large synthetic image database, I propose the development of a rigorous quality control process to ensure the clinical relevance and ethical integrity of the data. Nonetheless, the work presented in this chapter successfully addresses the shortage of high-quality ultrasound brain samples in the medical field and serves as a strong foundation for more advanced model conditionings explored in the following chapter.

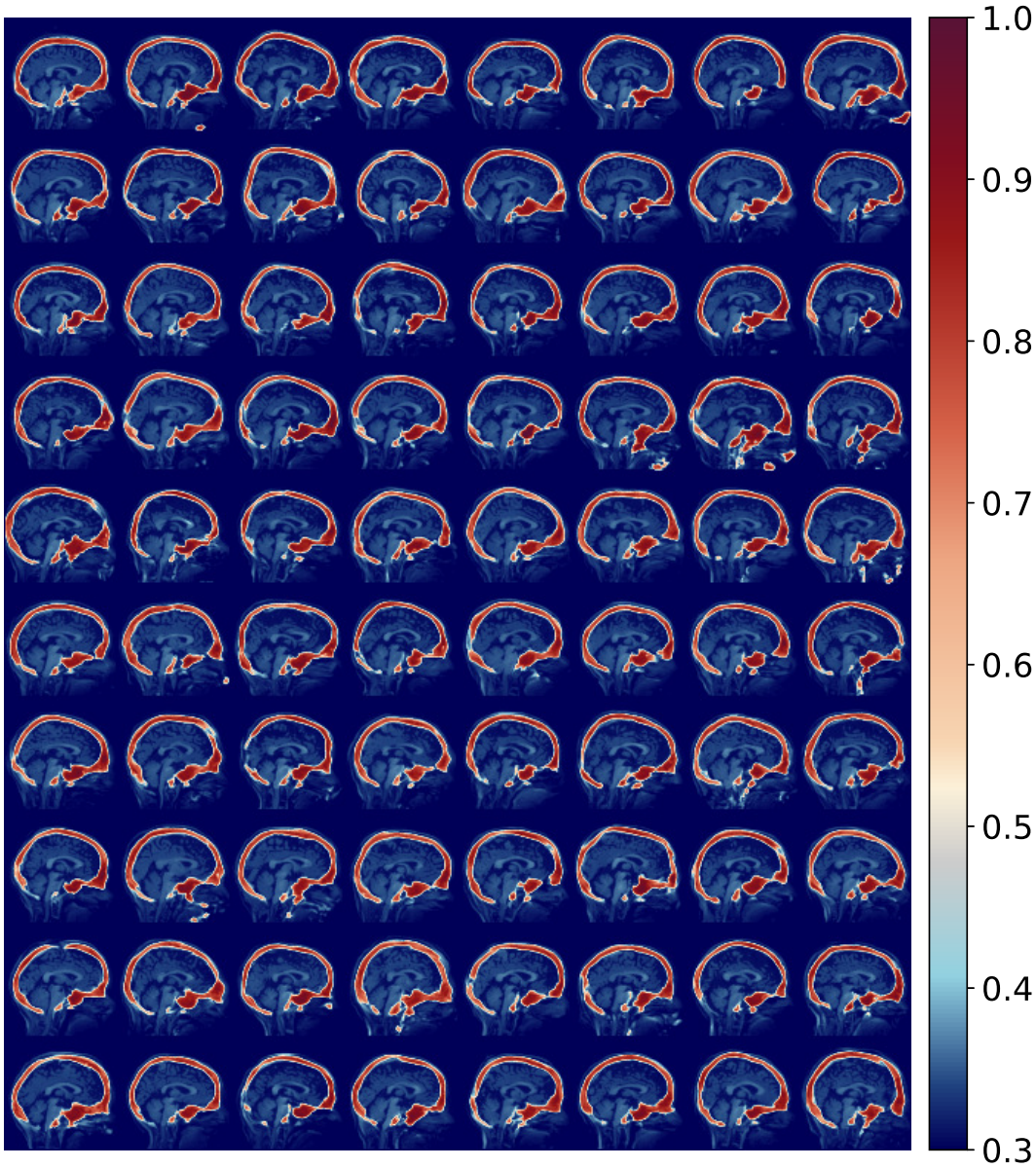


Figure 4.15: Examples of generated neuroacoustic brain samples from LDM-5-DDIM with 25 sampling steps for the mid-sagittal plane of the brain.

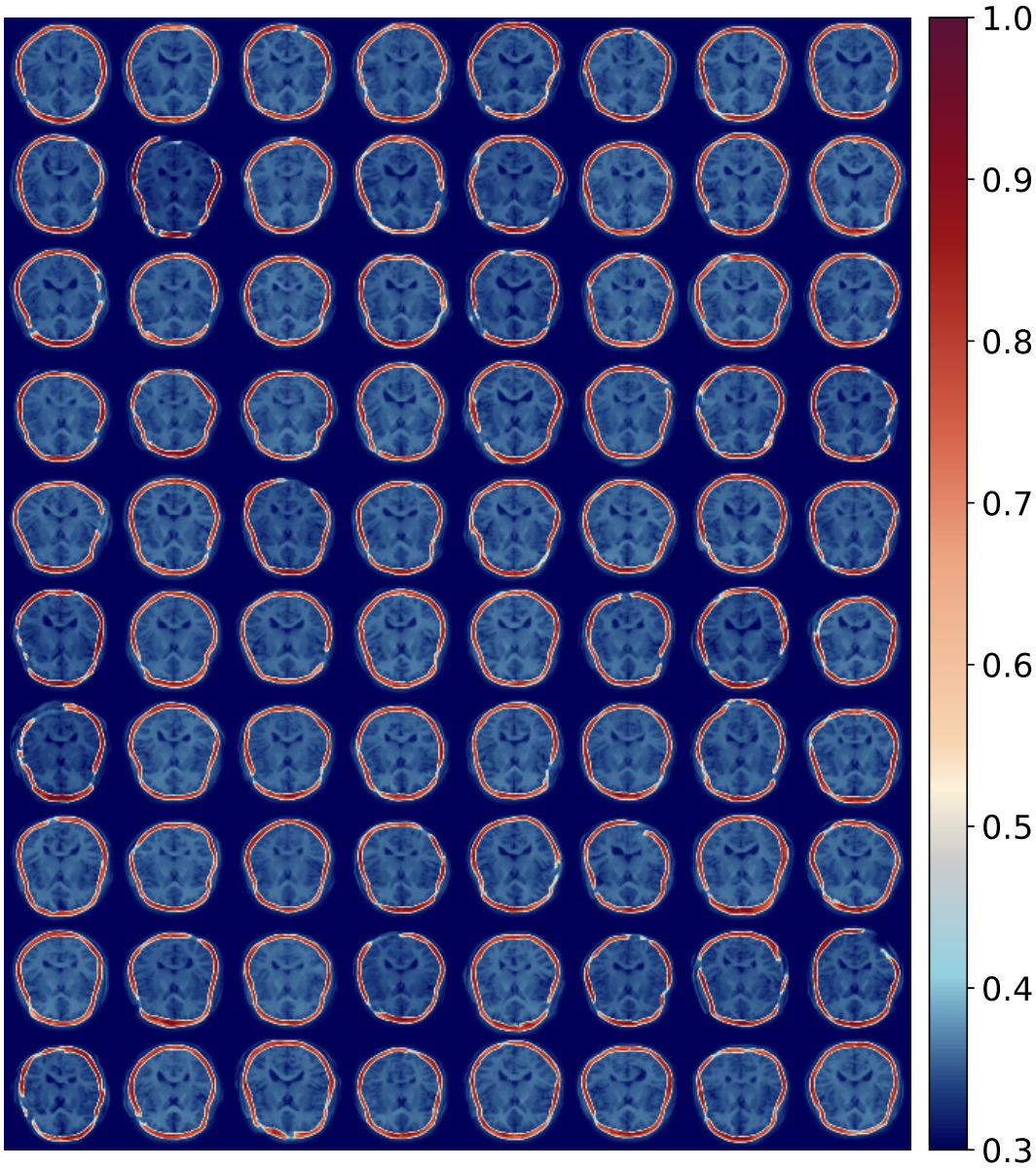


Figure 4.16: Examples of generated neuroacoustic brain samples from LDM-5-DDIM with 25 sampling steps for the mid-axial plane of the brain.

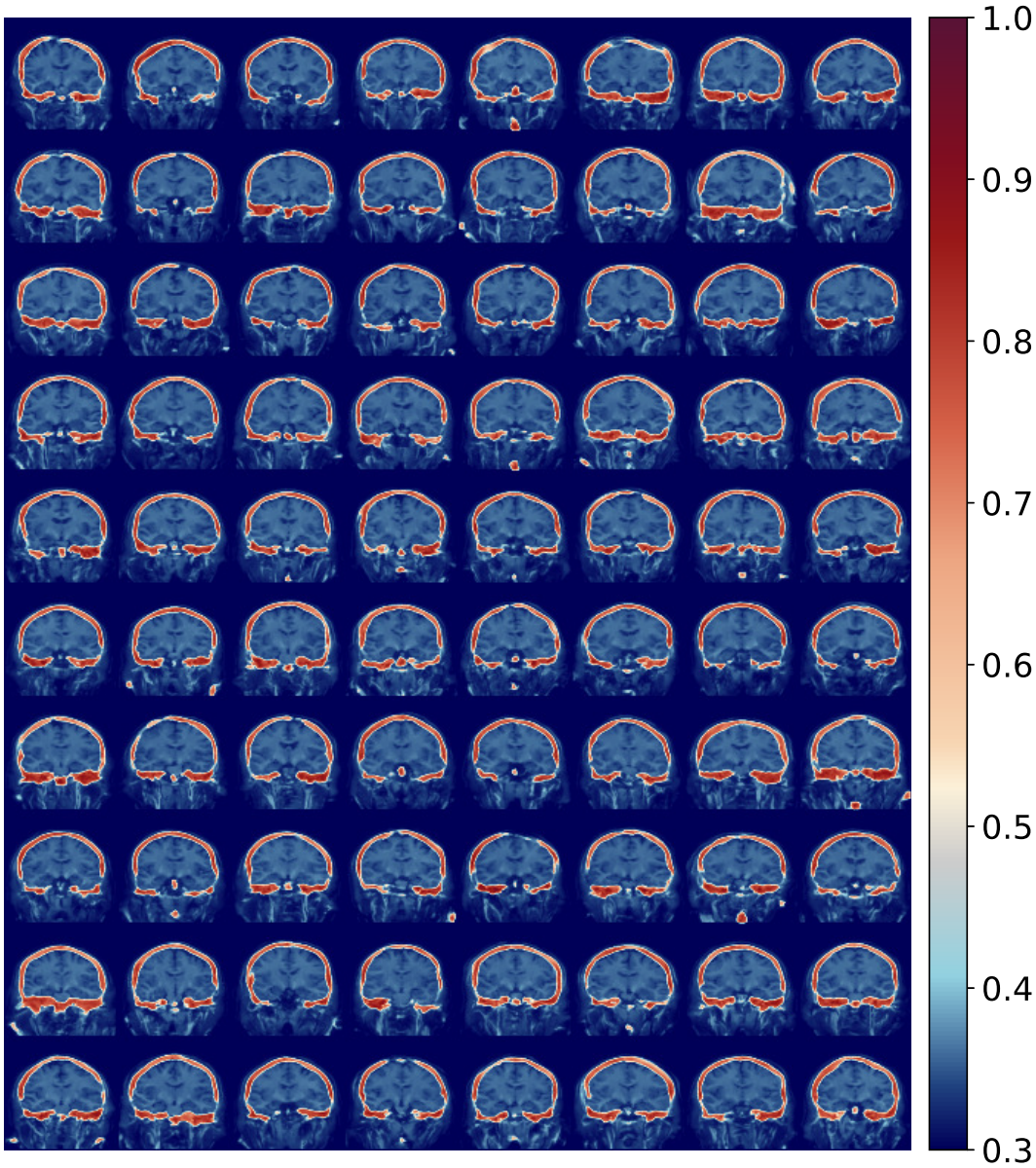


Figure 4.17: Examples of generated neuroacoustic brain samples from LDM-5-DDIM with 25 sampling steps for the mid-coronal plane of the brain.

Chapter 5

Towards Ultrasound Tomography with Generative Models

Magnetic Resonance Imaging (MRI) and X-ray Computed Tomography (CT) are the two principal technologies commercially used for structural brain imaging, whereas conventional ultrasound tomography (UT), though effective for other body parts, cannot resolve intra-cranial structures due to the high impedance contrast between the skull and soft tissues in the brain (von Ramm et al., 1978; Ylitalo et al., 1990; Guasch et al., 2020). UT also struggles to produce high-resolution images due to simplified modelling of non-linear wave propagation. However, UT offers advantages in speed, safety, and portability over MRI and CT.

Recent advances have aimed to enhance UT's imaging capabilities, making it more competitive with other modalities (Guasch et al., 2020; Strong, 2023; Robins, 2021; Calderon Agudo, 2018; Calderon Agudo et al., 2018; Bates et al., 2022). In particular, full-waveform inversion (FWI) of transmitted ultrasound is a technique that addresses UT's limitations by handling high-frequency, non-linear wavefield dynamics to produce images at greater fidelities and resolutions (Tarantola, 1984; Virieux and Operto, 2009). FWI applied to human body imaging has shown exceptional image reconstructions comparable to MRI and, for the first time, offers a viable method for intra-cranial imaging through sufficient processing of transmitted wave signals (Guasch et al., 2020; Strong, 2023; Robins, 2021; Calderon Agudo, 2018;

Calderon Agudo et al., 2018; Bates et al., 2022).

However, FWI is accompanied by significant challenges in medical imaging due to its high computational cost, reliance on low-frequency signals, and sensitivity to initial parametrisation, limiting its feasibility for routine clinical applications (Warner et al., 2013; Shah et al., 2012; Guasch et al., 2020). Overall, FWI remains underexplored in medical contexts and requires further advancements to address its algorithmic and computational complexities (Guasch et al., 2020).

This chapter addresses the central objective of this thesis by exploring the potential of state-of-the-art latent diffusion models (LDMs) (Rombach et al., 2022) to statistically encapsulate complex, non-linear ultrasound data and synthesise high-quality, individualised brain images, thereby overcoming the key limitations of FWI. Building on Chapter 4, which demonstrated that well-trained LDMs can model the intricate distribution of neuroacoustic images and generate randomised samples in near-real-time, this chapter examines whether these models can reconstruct the precise sample corresponding to specific ultrasound data, rather than producing generic outputs.

The transition from generic to tailored image generation in computer vision has proven effective when networks are conditioned with specific prior information about the desired image synthesis (Rombach et al., 2022; Podell et al., 2023). For example, an LDM trained to generate images of dogs could be conditioned with attributes like coat colour and breed to produce an image of a brindle boxer. However, applying this approach to clinical settings is a more delicate task. In the dog image example, there are many possible images of brindle boxers that could satisfy the given condition, and the inherently stochastic nature of LDMs favours this diversity. Such variability, while advantageous in creative tasks, is less suitable in clinical contexts, where achieving high accuracy and precision in data-to-image mapping is essential to ensure diagnostic reliability.

Here, I propose an LDM architecture conditioned on each brain's ultrasound data, embedded in latent representations, to enable targeted image generation. To the best of my knowledge, this approach is entirely novel and has not been attempted before. The primary goal is to develop a trained network that, at inference time,

can synthesise from noise the specific brain image corresponding to the provided ultrasound data, effectively creating a fast and reliable process for data-to-image inversion in a probabilistic framework. This method could potentially eliminate the need for low-frequency content, reliance on *a priori* information, and the computational expense of FWI, while achieving high-resolution and high-fidelity images of the brain.

The chapter begins by presenting the design of the LDM architecture that is conditioned on the ultrasound data, outlining its training and sampling algorithms. It then proceeds by detailing the forward modelling of the ultrasound data used to condition the LDM, followed by a discussion of the processes attempted to encode the resulting data into representative latent spaces. Despite the proposed model not being fully operative by the time-frame of this thesis, the chapter concludes with a critical discussion of the challenges encountered and suggestions for future work to make the proposed approach feasible. The software framework and datasets for this research are consistent with those detailed in Chapter 4, with code presented in the same GitHub repository.

5.1 Conditional Latent Diffusion Models

The extensive development of conditional diffusion models (Saharia et al., 2022; Rombach et al., 2022) has targeted modern computer vision tasks, including text-to-image synthesis, class-conditional generation, layout-to-image transformation, and inpainting. The condition $\mathbf{c}$ is integrated into the denoising U-Net (Ronneberger et al., 2015) either through concatenation or cross-attention (see Section 2.1.6), following encoding by a conditioner network $\mathcal{T}_\chi(\mathbf{c})$ parameterised by trainable weights χ. The network $\mathcal{T}_\chi(\mathbf{c})$ can be optionally pre-trained or trained together with the denoising network.

This conditioning mechanism steers the model to generate outputs that conform to the given context. In text-to-image synthesis, for example, the condition is a figure caption; in super-resolution, it corresponds to the low-resolution input; and in inpainting, it represents the corrupted image. During sampling, the user-specified

condition is combined with a random latent vector to direct the denoising and synthesis process. The original work by Rombach et al. (2022) demonstrated that this approach achieves state-of-the-art performance across these tasks, underscoring the LDM's capacity to generate high-quality, contextually guided outputs.

In image-to-image tasks, the authors applied $\mathcal{T}_\chi(\mathbf{c})$ as the identity function, concatenating a resampled condition directly to the latent space variables. Since in these tasks the condition is structurally aligned with the target latent space (as demonstrated in Chapter 4, where the autoencoder latent space closely mirrors the input image), concatenation provides an effective and computationally efficient conditioning method. Conversely, for text-to-image tasks, $\mathcal{T}_\chi(\mathbf{c})$ is implemented as a pre-trained unmasked transformer (Vaswani et al., 2017), which processes a tokenised representation of the input $\mathbf{c}$, yielding a latent output $\tilde{\mathbf{c}} = \mathcal{T}_\chi(\mathbf{c}) \in \mathbb{R}^{M \times d_t}$. This output is subsequently mapped into the U-Net (Ronneberger et al., 2015) via multi-head cross-attention. The authors found that alternating self-attention with cross-attention layers inside the denoising network allowed the LDM to capture internal dependencies within the data while integrating conditional information. This approach enables the model to produce contextually accurate outputs while maintaining fidelity to the data distribution.

In this work, the conditional LDM is redesigned to incorporate ultrasound data representations as conditioning inputs. Similar to image-to-image tasks, the conditioning is spatially organised, yet, as in text-to-image tasks, it involves a different modality with a non-trivial relationship to the latent images. This complexity stems from the non-linear wave equation, which, without physics-informed guidance, presents a challenge for direct network learning. Furthermore, as discussed in later sections, the dimensionality of ultrasound data is substantially higher than that of typical conditioning inputs in image-to-image or text-to-image tasks. Consequently, the network $\mathcal{T}_\chi$ must encode this data into a lower-dimensional and semantically disentangled space suitable for guiding the LDM. The revised schematics of the conditional architecture are shown in Figure 5.1, with training and sampling procedures detailed in Algorithms 9 and 10, respectively.

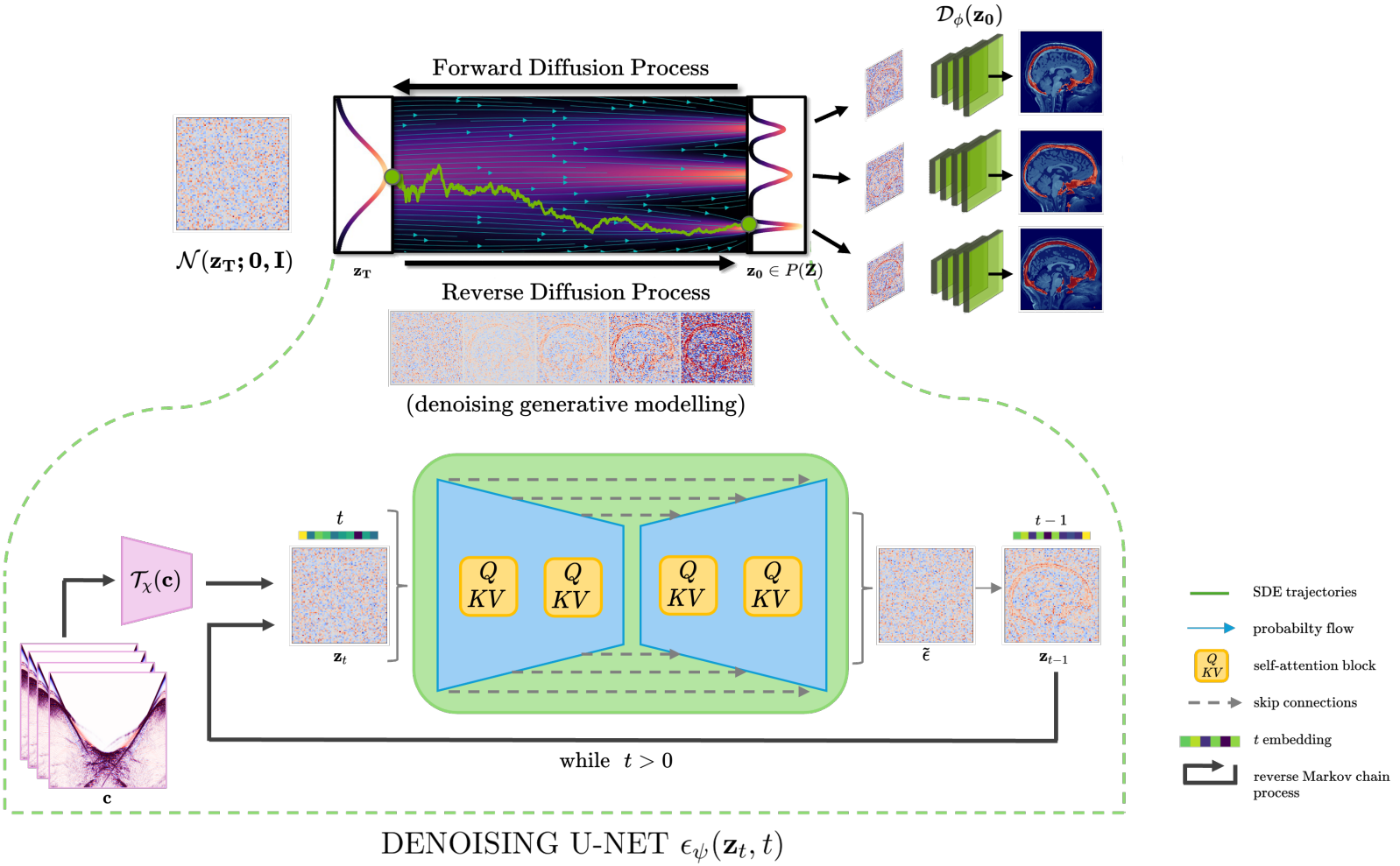


Figure 5.1: Dive into the denoising attention U-Net (Ronneberger et al., 2015) that is trained to predict the reverse diffusion process of the forward SDE. The model takes as input the sample degraded by $\boldsymbol{\epsilon}$ at time step t, defined by the noise scheduler, along with the sinusoidal positional embedding vector of the associated time step, and the embedding $\tilde{\mathbf{c}} = \mathcal{T}_\chi(\mathbf{c})$ of a condition $\mathbf{c}$. The embedded condition is inserted by either concatenation to $\mathbf{z_t}$ in the channel dimension, or by cross-attention in the attention blocks. The model then learns to predict the degradation noise $\tilde{\boldsymbol{\epsilon}} = \boldsymbol{\epsilon}_\psi(\mathbf{z}_t, \tilde{\mathbf{c}}, t)$, at each t, by minimising the Euclidean distance $||\boldsymbol{\epsilon} - \tilde{\boldsymbol{\epsilon}}||_2^2$. Once trained, the model's predicted noise is subsequently used to estimate the previous sample of the diffusion process, given an unseen condition embedding $\tilde{\mathbf{c}}$. Figure adapted (Kreis et al., 2023) and (Rombach et al., 2022).

Algorithm 9 Training Algorithm for Conditional LDMs

Input: Training data distribution $q(\mathbf{x}, \mathbf{c})$, pretrained encoder $\mathcal{E}_\theta$, conditioner network $\mathcal{T}_\chi$, noise scheduler $\bar{\alpha}_t$
Output: Trained denoising model $\boldsymbol{\epsilon}_\psi$
Initialise model parameters ψ, optionally χ if not pre-trained
repeat
  for number of training iterations **do**
    Step 1: Sample data and encode to latent space
    Sample data and condition $(\mathbf{x}, \mathbf{c}) \sim q(\mathbf{x}, \mathbf{c})$
    Encode latent representation:

$$\mathbf{z}_0 \leftarrow \mathcal{E}_\theta(\mathbf{x}) \tag{5.1}$$

    Encode condition:

$$\tilde{\mathbf{c}} \leftarrow \mathcal{T}_\chi(\mathbf{c}) \tag{5.2}$$

    Step 2: Corrupt latent sample and compute loss
    Sample diffusion step $t \sim \text{Uniform}(\{1, \ldots, T\})$
    Sample noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

$$\mathcal{L} = \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\psi \left(\sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon},\, t,\, \tilde{\mathbf{c}} \right) \right\|_2^2 \tag{5.3}$$

    Step 3: Update model parameters

$$\psi, \chi \leftarrow \psi, \chi - \alpha \nabla_{\psi,\chi} \mathcal{L} \tag{5.4}$$

  end for
until convergence of the conditional denoising model

Algorithm 10 Sampling Algorithm for Conditional LDMs

Input: Trained denoising model $\boldsymbol{\epsilon}_\psi$, pretrained decoder $\mathcal{D}_\phi$, conditioner network $\mathcal{T}_\chi$, conditioning data $\mathbf{c}$, noise schedule $\alpha_t, \bar{\alpha}_t, \sigma_t$
Output: Generated sample $\mathbf{x}$
Initialise: Sample latent noise $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
Encode condition:

$$\tilde{\mathbf{c}} \leftarrow \mathcal{T}_\chi(\mathbf{c}) \tag{5.5}$$

for $t = T, \ldots, 1$ **do**
  Sample $\zeta \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else set $\zeta = 0$
  Perform conditional denoising step:

$$\mathbf{z}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{z}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\psi(\mathbf{z}_t, t, \tilde{\mathbf{c}}) \right) + \sigma_t \zeta \tag{5.6}$$

end for
Return: $\mathbf{x} = \mathcal{D}_\phi(\mathbf{z}_0)$ (final generated sample)

5.2 Ultrasound Forward Modelling

I used the Stride package (Cueto et al., 2022) to model the ultrasound dataset employed for conditioning the LDM, with modelling parameters chosen to replicate realistic clinical conditions. Synthetic data was generated by solving the two-dimensional, static-density, isotropic acoustic wave equation in the time domain using a fourth-order finite-difference time-stepping algorithm. The transducers were modelled as unfocused single elements, with 1,000 acting as receivers, 32 of which also served as source emitters. This configuration produced over 32,000 source-receiver trace records of acoustic pressure, each lasting 200 μs, per sample.

The spatial and temporal resolutions were selected to mitigate numerical dispersion and instability, ensuring accuracy for the finite-difference solver and the upper bound of brain sound speed. The spatial grid comprised 480×480 cells with a spacing of 0.5 mm, while the temporal domain was discretised into 2,500 steps with an interval of 0.08 μs. To absorb wavefield boundaries, an additional 40 cells were appended around the periphery of the grid in each dimension (Cueto et al., 2022; Yao et al., 2018). Figure 5.2 provides a visual summary of the experimental setup, depicting a 2D neuroacoustic brain slice submerged in water and encased by a circular array of transducers.

Supporting work from (Barger, 1979; Guasch et al., 2020; Fry and Barger, 1978; Marty et al., 2021) showed that sub-millimetre resolutions of the brain and its vascular system can be imaged at frequencies below 1MHz. Good signal-to-noise ratios are obtained in the range of 100-850kHz where transmission losses in soft tissues and scattering and anelastic losses in the skull are small (Guasch et al., 2020). Accordingly, the synthetic dataset generated utilised a three-cycle tone burst waveform with a peak amplitude at 500 kHz, covering a frequency bandwidth from approximately 100kHz-1MHz. Visualisation of the source waveform used for data modelling and its frequency spectrum are shown in Figure 5.3. Leveraging Stride's (Cueto et al., 2022) GPU support, the synthetic data was efficiently generated on an NVIDIA GeForce RTX 2080 Ti (11 GB), with each sample requiring less than two minutes to compute.

Figure 5.4 presents a normalised ultrasound gather for one of the 32 shots

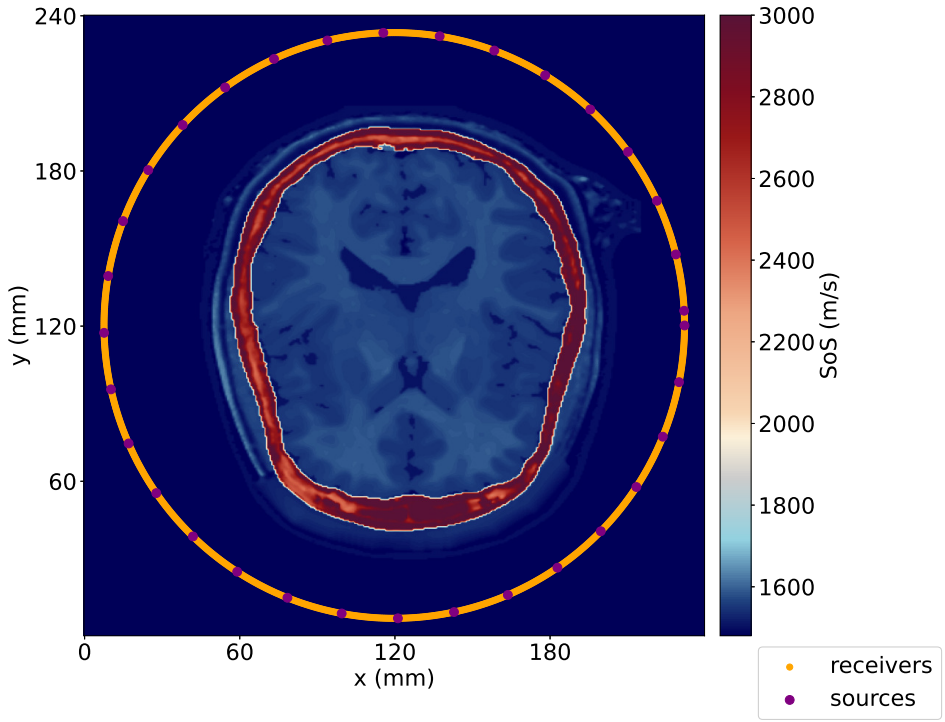


Figure 5.2: Hypothetical acquisition setup displayed on the mid-axial plane of a brain sample, showing the distribution of speed of sound (SoS) across the medium in m/s. The spatial grid used consists of 480×480 cells, each with a spacing of 0.5 mm, capturing detailed structural information within the brain and skull. The yellow markers represents the receiver locations, while the purple markers indicate source positions around the perimeter, outlining the circular transducer arrangement for ultrasound acquisition.

in the example experiment, highlighting key features of ultrasound reflection and transmission in the brain. Visible in the figure are the strong initial arrivals where the source propagates through water directly to the receiving transducers. Also visible are the pronounced reflections at the water-skull interface caused by the significant discrepancy in acoustic impedance between the two mediums, and the guided waves that propagate through the skin and subcutaneous fat. intra-cranial transmissions are also observable, but are limited to weak signals due to low energy penetration caused by the high reflectivity of the skull in relation to intra-cranial soft tissues. The visualisation of intra-cranial transmissions is only made possible when the plotting values are clipped to range $\pm$ 0.01, corresponding to 1% of the maximum normalised shot value (note figure Figure 5.4 compared to the amplitude of the waveform in

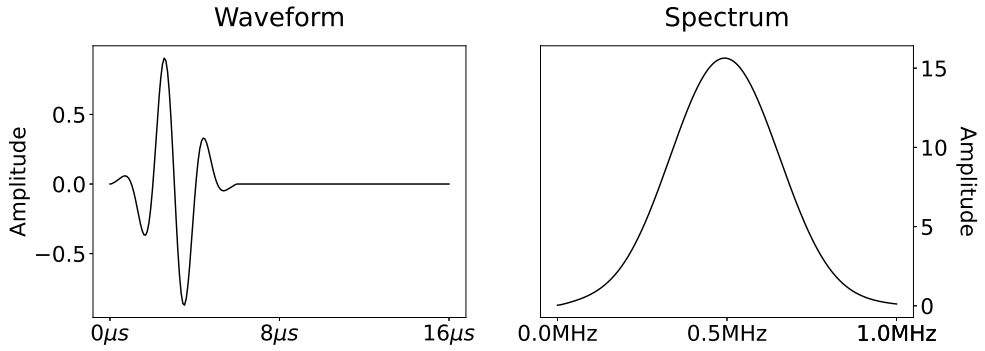


Figure 5.3: Source wavelet (left) and its frequency spectrum (right) used for ultrasound modelling. The waveform is a three-cycle tone burst with a peak amplitude occurring at 500 kHz, covering a frequency bandwidth of approximately 100 kHz to 1 MHz.

Figure 5.3).

5.3 Encoding Ultrasound Data

The computational demands of processing raw ultrasound data at scale render direct application in deep learning models impractical. Conventional high-dimensional datasets in computer vision, typically considered large at approximately $3 \times 1024 \times 1024 \approx 10^6$ data points (Podell et al., 2023), are significantly smaller than the ultrasound datasets examined here, which can reach dimensionalities of $32 \times 1000 \times 2500 \approx 10^7$ — an order of magnitude larger. In clinical acquisitions, the number of shot gathers obtained for FWI can exceed this dimensionality, reaching scales of 10^9 (Guasch et al., 2020), further exacerbating the computational burden. Data reduction approaches are therefore essential to enable the practical application of deep learning methods to ultrasound data.

5.3.1 Shot-Gather Stacking

While successful implementations of FWI often requires processing vast amounts datasets, it is known that such data frequently contains considerable redundancy, especially in spatial information (Hagen, 1982). Ultrasound shot gathers, in particular, exhibit substantial sparsity due to wave propagation through homogeneous media, as evidenced by the sparsity in Figure 5.4.

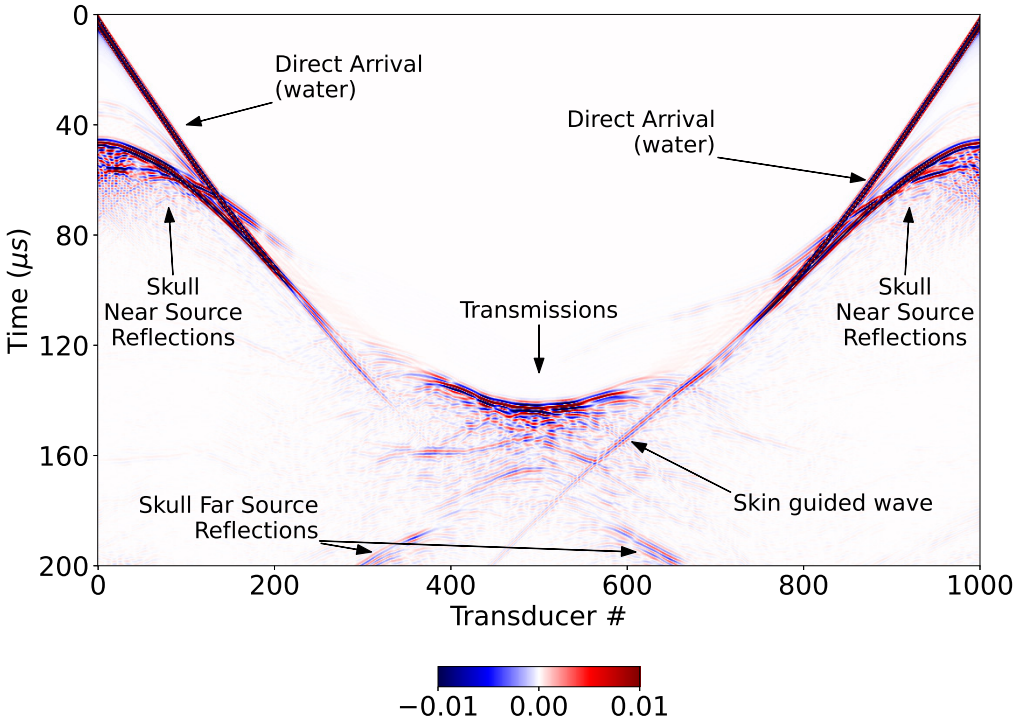


Figure 5.4: Synthetic ultrasound shot gather generated from forward modelling on an acoustic velocity map of a brain, illustrating key features of wave propagation through brain and skull structures. The shot amplitude (colour scale) is normalised to the range ±1 using a min-max scaling approach and visualisation is clipped to range ± 0.01 to increase contrast. Major wave components are annotated, including direct arrivals through water, near-source and far-source reflections from the skull, transmission paths through brain tissue, and guided waves through skin and subcutaneous fat.

Exploiting these redundancies, the initial approach to dimensionality reduction involves superimposing all records from each transducer to form a composite data stack, as shown in Figure 5.5. This composite stack compresses the data by a factor of 32 while preserving the key features and statistical properties of the original data, despite some distortion to their physical meaning. Since the proposed model is not physics-guided, maintaining strict physical interpretations is not a primary concern. Instead, the focus is on preserving or enhancing the statistical distinctions between samples that the network can exploit.

In the following sections, I explore various methods for embedding and further reducing the dimensionality of ultrasound data into lower-dimensional spaces using established machine learning techniques. Notably, previous applications of neural

networks in FWI (see Section 2.2.2) have yet to fully leverage the potential of learned embeddings in this context. The composite stacking approach thus plays a crucial role in making the data manageable for these more advanced workflows. Without this preliminary reduction, the high dimensionality would render subsequent processing approaches computationally infeasible with the resources available.

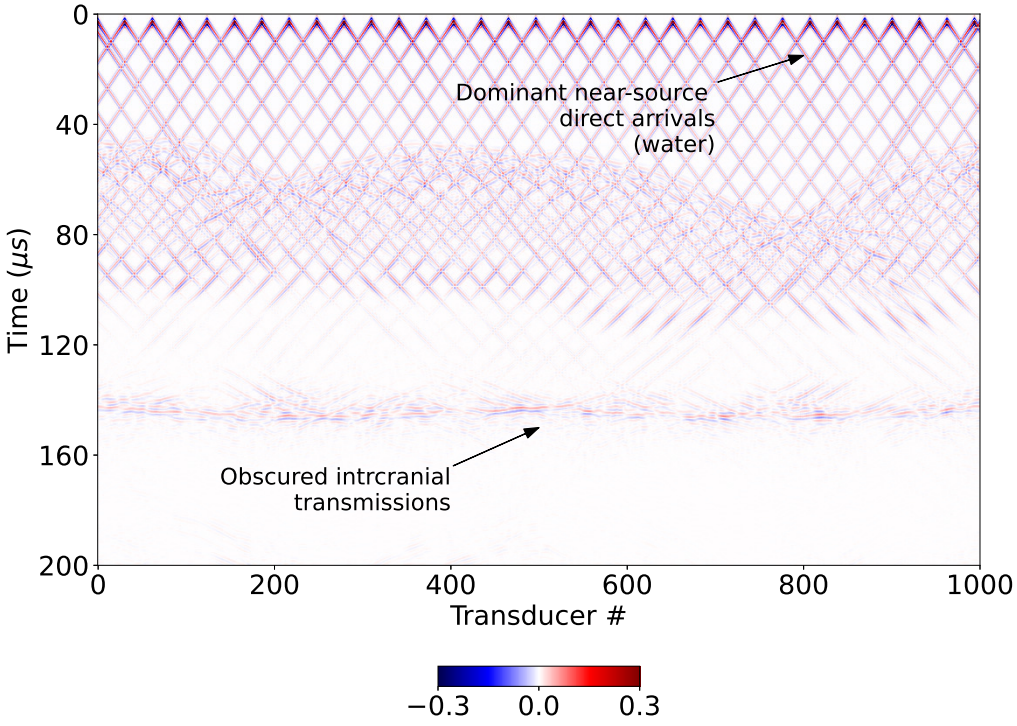
5.3.2 Automatic Gain Control

Adapting ultrasound data for deep learning presents additional challenges due to the dominant reflections caused by the high acoustic contrast between the skull's bone tissues and the surrounding soft tissues, resulting in transmitted waves that retain only approximately 22-28% of their source signal strength (Guasch et al., 2020). This issue is further compounded during the construction of a composite stack, where strong reflections are disproportionately reinforced relative to weaker signals. Such disparities can introduce biases in the network's learning process, as higher values dominate during backpropagation.

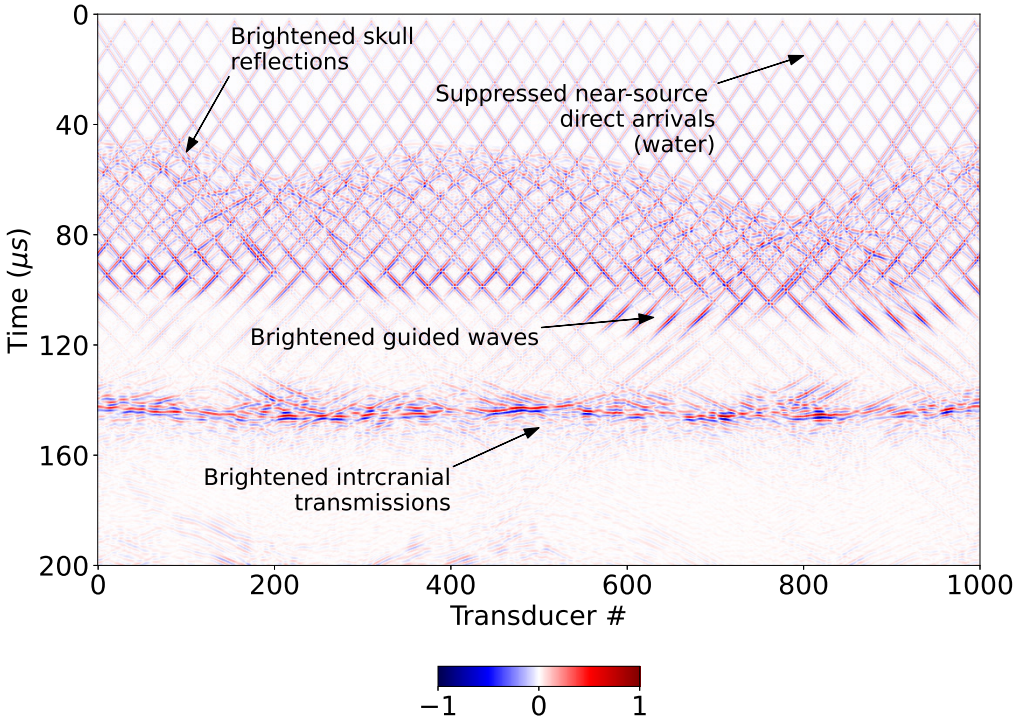
To counteract this bias and reduce the risk of imbalanced learning, a trace-wise least mean squares automatic gain control (LSM-AGC) (Algorithm 11 in Appendix B) is applied to each stack sample, effectively equalising the signals within the data. Figure 5.5 shows the stack before and after equalisation through LSM-AGC (note figure scales), while Figure 5.6 provides a side-by-side comparison of the stack histograms, with and without LSM-AGC. Both figures demonstrate the efficacy of LSM-AGC in enhancing key signal features within the ultrasound data, particularly for intra-cranial transmissions and guided waves, while attenuating more dominant but less informative signals, such as the near-source direct arrival. This adjustment also serves as a preparatory step for embedding the data into a lower-dimensional space, enabling the network to focus on the most distinctive features of the brain.

5.3.3 Principal Component Analysis

While the composite stack could, in principle, be used to condition the network, it is important to acknowledge the dimensional discrepancy between the stack and typical conditioning inputs. Image conditioning data in computer vision typically falls



(a) Stack



(b) Stack + LSM-AGC

Figure 5.5: Stacked shot gathers from 32 sources for a single brain sample. (a) The raw stack, displaying dominant near-source direct arrivals (propagating in water) and obscured intra-cranial transmissions. (b) The stack with processed with LSM-AGC, brightening intra-cranial transmissions, guided waves, and skull reflections while suppressing near-source direct arrivals.

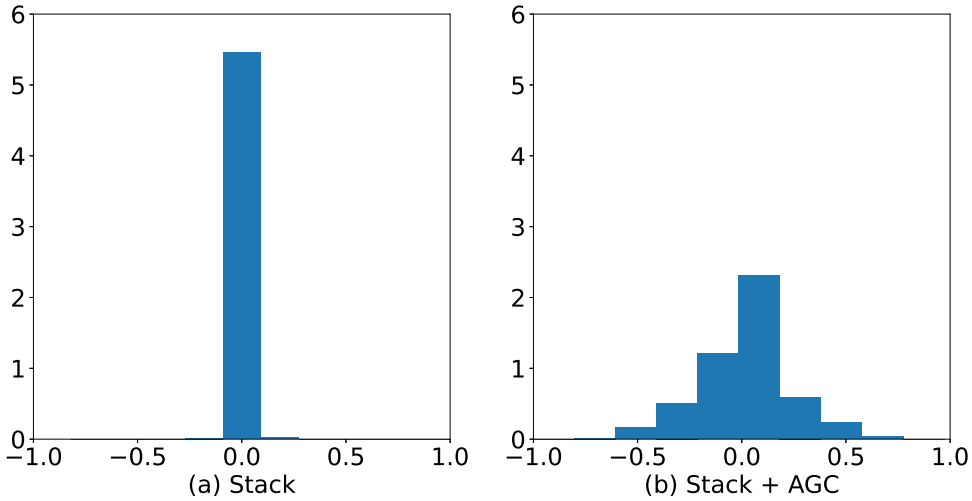


Figure 5.6: Histograms of the normalised amplitude distributions for the stacked shot gathers. (a) Raw stack, showing a narrow distribution of values centred around zero, indicating limited amplitude variation without gain adjustments. (b) Stack processed with LSM-AGC, which results in a broader amplitude distribution with increased variability.

within the dimensional range of 10^3, whereas the stack, representing the full extent of ultrasound information, has a dimensionality of 10^6. This discrepancy presents a significant challenge, requiring the network not only to generate high-resolution brain images but also to process even higher-resolution conditioning inputs. Networks are generally designed to handle conditioning data of similar resolution (e.g., inpainting tasks) or to operate with lower-dimensional conditioning inputs (e.g., text-to-image synthesis). Although theoretically possible, training networks at such high resolutions for both conditioning and output is computationally prohibitive in most research contexts, including this PhD work, where available resources do not permit experiments at this scale.

The most straightforward approach to address this issue is to resample the stack to more manageable dimensions, such as 128x128 or 256x256. However, this downscaling reduces the data's information content directly by a factor of 10^2 and 10^1, respectively, leading to a loss of detail that may undermine the network's ability to differentiate between samples.

To effectively utilise the stack as a conditioning input, more sophisticated methods of dimensionality reduction are necessary — methods that preserve the essential information required to guide the network's sampling towards accurate guided out-

puts. Principal Component Analysis (PCA) is a commonly used approach for this purpose, as it isolates the orthogonal components of the data with the highest variance, capturing the primary structural and statistical features. Unlike deep learning techniques, PCA does not require large models to fit the data and can be quickly applied to new datasets, making it a practical and efficient choice for compressing high-dimensional data while retaining the key characteristics necessary for effective conditioning.

PCA was applied on the stack data through three distinct approaches. These three methods were selected due to PCA's limitation with grid data, which requires one-dimensional vectorisation prior to application. In the first method, each stack sample is vectorised through flattening to allow for PCA computation over the entire dataset, enabling an understanding of variability in the broader context of the samples. However, this approach sacrifices structural information, potentially removing important temporal and spatial variability within individual stacks. The second and third approaches, by contrast, maintain stack-specific structural integrity. In the second method, each stack sample is treated independently, with PCA applied along the offset axis. This approach slices the data horizontally, preserving the full temporal footprint while eliminating spatial redundancies. In the third method, each stack sample is also treated independently, but PCA is applied vertically along the trace dimension, preserving spatial information and addressing temporal redundancies.

In the first approach to dimensionality reduction, where the AGC-stack ultrasound dataset is treated as unstructured vectors, PCA is applied in a 512-dimensional space, reducing the data from $\mathbb{R}^{1000\times2500} \rightarrow \mathbb{R}^{512}$. The left plot in Figure 5.7 presents the cumulative explained variance η^2 as a function of the number of principal components, on a logarithmic scale, showing that 512 components capture nearly all of the dataset's variance ($\eta^2 = 0.998$), and therefore are sufficient to represent the data. On the right, a series of visualisations shows the first 50 principal components learned from the PCA fitting, reshaped to match the original spatial layout of the ultrasound data. These images reveal variations in transmission and reflection patterns, highlighting PCA's ability to capture essential features of the data in a reduced form. Although flattening the dataset for PCA analysis results in a loss of

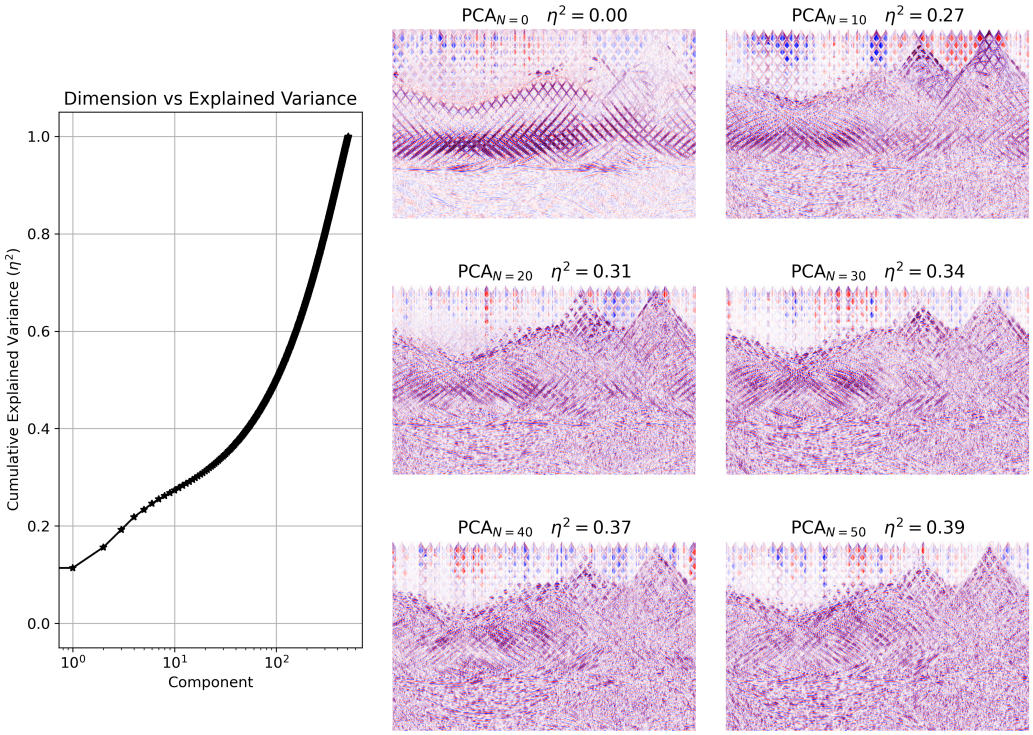


Figure 5.7: Application of PCA to the ultrasound dataset, with each sample flattened into a 1D vector. The left plot shows cumulative explained variance η^2 against the number of principal components on a logarithmic scale, showing that 512 dimensions are sufficient to explain the data ($\eta^2 = 0.998$). The images on the right depict principal components $N \leq 50$ learned by the PCA fitting, reshaped to match the original data dimensions, showing high variability in transmissions and reflections.

its original structural format, it is clear from Figure 5.7 that the structural variations are present in the principal components, capturing inherent spatial heterogeneities of the ultrasound signals.

In the second approach, where PCA is applied along the offset axis treating time-step signals of the AGC-stack as an independent data point, each sample is reduced from $\mathbb{R}^{1000\times2500} \rightarrow \mathbb{R}^{128\times2500}$. A total of 128 PCA components is sufficient to describe major temporal variations, with a sample-averaged cumulative explained variance of $\eta^2 = 0.994$. Figure 5.8 shows the mean of the reduced forms across all samples on the left, where each point in the x-axis corresponds to the average principal component value for a given time step, capturing broad, recurrent structures across

the reduced form of the dataset. The variance map plot on the right displays pixel-wise variance, identifying regions of high variability across the reduced samples. Notably, the largest variances occur in the skull near-source reflections and the guided waves of propagation through skin (time-steps <1500 or approximately 120 μs), and in intra-cranial transmissions around time-step 1800 (144 μs), highlighting global variations in the most prominent zones of acoustic variability across the dataset.

The variability in these critical regions suggests that this form of reduction would provide sufficient statistical discrimination precisely in these essential areas, rather than in the more uniform, non-critical regions of the data. This targeted focus on critical regions could assist the conditioning of the network by encouraging it to focus on specific regions of interest, i.e. skull, skin, and intra-cranial structures, potentially enhancing its capacity to learn the data-to-image mapping more accurately. One caveat, however, is that the reduction from 1000×2500 to 128×2500 does not fully mitigate practical computational constraints, and additional resampling along the time dimension may still be required.

In the third approach, similar to the second, PCA is applied independently to each AGC-stack sample; however, here each trace is treated as a distinct data point, reducing the dimensionality from $\mathbb{R}^{1000\times2500} \rightarrow \mathbb{R}^{1000\times128}$. With 128 components, this method achieves a sample-averaged explained variance of ($\eta^2 = 0.996$), showing that the major variations within each sample are also well captured through a few components of PCA. The mean and variance plots of this approach, shown in Figure 5.9. Unlike the clear structures observed in Figure 5.8, the patterns in these plots are less intuitively interpretable, making it challenging to derive immediate insights. However, the variance map reveals a higher spread of variance across different areas compared to the second approach, which might be beneficial for statistical discrimination and could potentially assist the network in identifying important features.

5.3.4 Feature Learning with Variational Autoencoders

Variational Autoencoders (VAEs) can be interpreted as a mathematical extension of PCA, expanding the technique's ability to represent complex, high-dimensional data by incorporating non-linearity and probabilistic modelling. Whereas PCA applies a

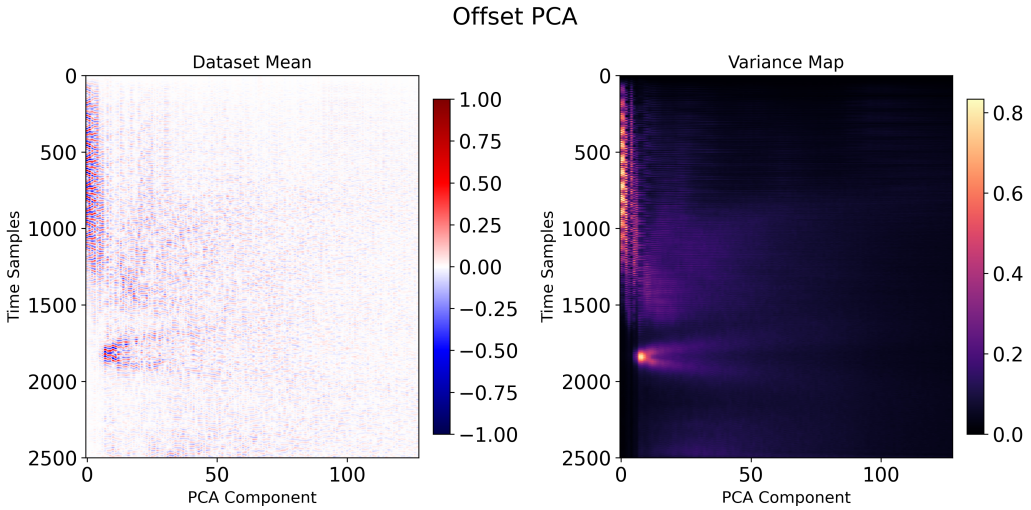


Figure 5.8: PCA applied along the offset axis of the AGC-stack ultrasound dataset, where each trace is treated as an independent data point, reducing each sample from $\mathbb{R}^{1000\times2500}$ to $\mathbb{R}^{128\times2500}$. The left plot displays the mean of the reduced forms across all samples, with each point in the x-axis reflecting the average principal component value for a given time step. The right plot, a variance map, shows pixel-wise variance across the reduced samples, revealing areas of high variability. High variance is observed in the skull near-source reflections and guided waves of propagation through skin (time-steps $<$1500, $\sim$120 μs), and in intra-cranial transmissions around time-step 1800 ($\sim$144 μs), corresponding to critical regions of acoustic variability, showing that the reduction provides strong statistical discrimination precisely in these essential areas. Sample-averaged cumulative explained variance $\eta^2 = 0.994$.

linear transformation to project data onto a lower-dimensional space, VAEs achieve this by employing neural networks as non-linear encoders and decoders, allowing the model to capture intricate relationships and learn meaningful features in the latent space. A key distinction lies in their treatment of the latent space: PCA yields a deterministic, fixed-size representation through linear projections that maximise variance, while VAEs adopt a probabilistic framework, modelling latent variables as random samples from a prior distribution. By leveraging neural networks, VAEs can also naturally integrate convolutional layers to capture structural hierarchies within the data, facilitating the extraction of meaningful representations and positioning them as highly effective feature learners for high-dimensional data.

The architecture of the VAE for this dimensionality reduction problem is a U-Net (Ronneberger et al., 2015) consisting of a feature layer with 32 channels and 2 downsampling and upsampling layers of channel dimensions (32, 64), each containing

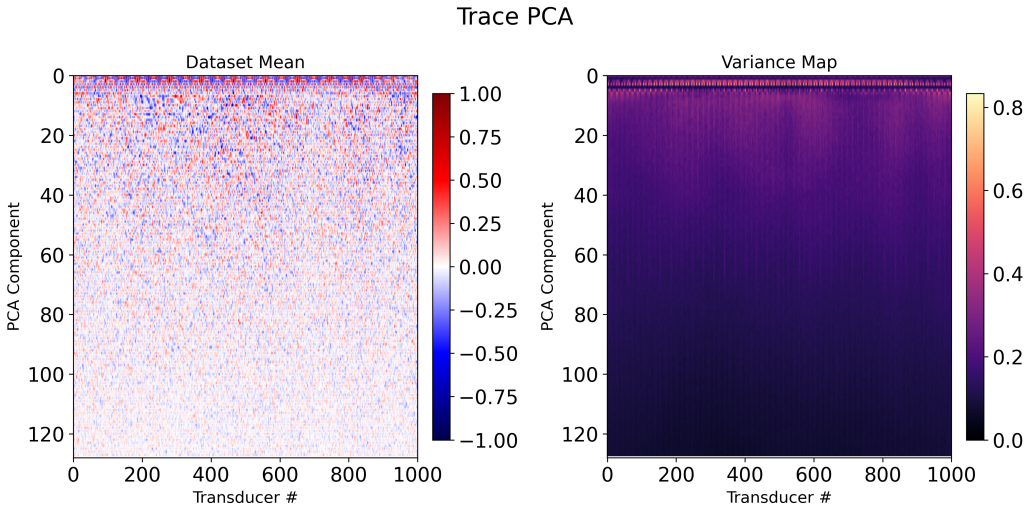


Figure 5.9: PCA applied along the time axis of the ultrasound dataset, where each trace is treated as an independent data point, reducing each sample from $\mathbb{R}^{1000\times2500}$ to $\mathbb{R}^{1000\times128}$. The left plot displays the mean of the reduced forms across all samples, with each position reflecting the average principal component value across transducers. The right plot, a variance map, shows pixel-wise variance across the reduced samples, indicating areas of higher variability. Unlike previous methods, this approach does not reveal clear structural patterns in the mean plot, making interpretability more challenging. However, the variance map shows a higher spread of variance across different areas, which could be beneficial for statistical discrimination and feature identification. Sample-averaged cumulative explained variance $\eta^2 = 0.996$.

2 residual units and a self-attention layer. To accommodate for computational memory capacity, the input AGC-stacks were resized to 200×400 with an antialiasing filter. Each model layer factorised the input size by 2, resulting in a latent space size of 50×100, arbitrarily chosen to be single-channel. Overall, this model contains just over 1 million trainable parameters and was trained using an L2 reconstruction loss and a KL latent space regularisation term with a weight of 1×10^{-2}. The optimiser used for training was Adam (Kingma and Ba, 2014), parameterised by $\beta_1 = 0.9$ and $\beta_2 = 0.999$, with batch size of 4 and SiLU (Elfwing et al., 2018) as activation. The model trained for 300 epochs until convergence. The data was split into 80% training and 20% validation sets, achieving reconstruction losses of 1.778×10^{-3} and 5.667×10^{-3} for training and validation, respectively. The model was trained on a single NVIDIA GeForce RTX 2080 Ti 11 Gb graphics card, taking approximately 4 hours to complete.

Similarly to the discussion on PCA, the mean and pixel-wise variance of the la-

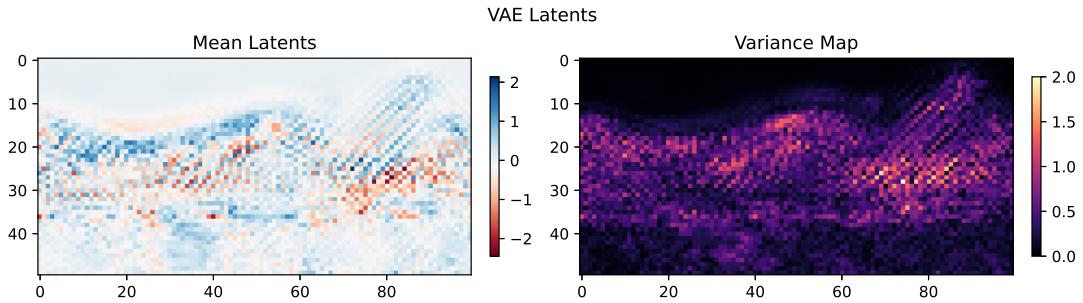


Figure 5.10: A statistical summary of the latent samples from a VAE trained with a U-Net (Ronneberger et al., 2015) architecture on ultrasound data stacks after LSM-AGC, using an L2 reconstruction loss and KL regularisation of the latent space. The mean of the latent samples (left) reveals structural patterns that reflect the preservation of spatial and temporal information from key aspects of the stack, while the variance map (right) highlights regions of high variability, including initial skull reflections and intra-cranial transmissions. This demonstrates the model's ability to capture and differentiate structural variations in the data within a compressed space, offering a more interpretable representation than PCA-based methods. The statistics are derived from 200 latent samples from the validation set.

tent representations were computed and visualised to understand the model's learned representations, presented in Figure 5.10. This figure demonstrates the preservation of spatial and temporal information, with the mean plot showing distinct structural patterns across the latent space that mirror the original data stack. The variance map, meanwhile, reveals a broad spread of variance across key regions like the initial skull reflections and intra-cranial transmissions, highlighting areas of high relevance. Unlike the PCA approaches, this variability is embedded within a structured latent space that more closely aligns with the actual data, improving its interpretability. This high variability also suggests that the model has effectively learned to differentiate between samples in the latent space.

5.3.5 Contrastive Representation Learning

Contrastive learning originated in visual representation learning as a method to tackle dimensionality reduction and derive invariant representations, which are features that preserve their integrity despite transformations or noise in the input data. The objective is straightforward: to map data into a space where similar samples reside in close proximity, while dissimilar ones are pushed apart. The concept was first explored by Hadsell et al. (2006), who applied contrastive learning to the MNIST

dataset to preserve digit similarity in latent representations. The authors demonstrated that this approach could eliminate unwanted variances in the data, enabling the model to learn invariant representations in a disentangled latent space, resilient to certain transformations (Hadsell et al., 2006).

In a supervised contrastive learning framework, the feature learning process helps disentangle the latent space by arranging samples such that those belonging to the same class are clustered closer together, while samples from different classes are positioned further apart. This separation occurs within a continuous semantic latent space, where relationships between classes are also encoded based on their semantic similarities; for example, images of dogs will lie closer to images of cats than to images of houses, as dogs and cats both belong to the broader category of animals.

Contrastive learning has also become a key technique in multi-modal training where representations from different data modalities benefit from aligning in a common latent space (Ramesh et al., 2021, 2022; Rombach et al., 2022; Scotti et al., 2024), being an instrumental backbone of famous generative models like DALL-E 2 by OpenAI, Stable Diffusion by Stability AI, and Midjourney by the Midjourney team.

In text-to-image embedding models such as CLIP (Contrastive Language-Image Pre-Training) (Radford et al., 2021), contrastive learning drives paired embedding samples of each modality towards proximity in the latent space, while unrelated pairs are pushed apart. This structured separation results in discriminative data representations that enhance model performance in downstream guided-generation tasks (Rombach et al., 2022; Ramesh et al., 2022; Alayrac et al., 2022). CLIP implements contrastive learning by optimising a symmetric, bidirectional contrastive loss known as Information Noise Contrastive Estimation (InfoNCE) loss, defined as (Oord et al., 2018):

$$\mathcal{L}_{CL_{i,j}} = \frac{1}{2N}\sum_{i=1}^{N}\left(-\log\frac{\exp(\text{sim}(\mathbf{x}_i,\mathbf{y}_i)/\tau)}{\sum_{j=1}^{N}\exp(\text{sim}(\mathbf{x}_i,\mathbf{y}_j)/\tau)} - \log\frac{\exp(\text{sim}(\mathbf{y}_i,\mathbf{x}_i)/\tau)}{\sum_{j=1}^{N}\exp(\text{sim}(\mathbf{y}_i,\mathbf{z}_j^I)/\tau)}\right), \tag{5.7}$$

where N is the batch size, $\mathbf{x}_i$ and $\mathbf{y}_i$ represent embeddings of each modality — image and text — for the i-th pair, $\text{sim}(\mathbf{x}_i, \mathbf{y}_j)$ denotes the cosine similarity between the embedding $\mathbf{x}_i$ and the embedding $\mathbf{y}_j$, and τ is a temperature parameter that controls the concentration of the distribution. The first term encourages each $\mathbf{x}_i$ embedding to be close to its corresponding $\mathbf{y}_i$ embedding and far from all other $\mathbf{y}_j$ embeddings in the batch, while the second term mirrors the first, pushing each embedding $\mathbf{y}_i$ close to its corresponding $\mathbf{x}_i$ embedding while distancing it from other $\mathbf{x}_j$ embeddings, thereby creating a discriminative feature space. Although the two terms are symmetric, the InfoNCE loss is implemented bidirectionally to ensure that encoders from each modality receive balanced gradient updates, preventing modality collapse and promoting mutual alignment within the shared embedding space (Radford et al., 2021).

Architecturally, contrastive learning frameworks leverage convolutional neural networks to encode representation embeddings, or a combination of Transformers and Vision Transformers to more effectively capture dependencies within the data (Oord et al., 2018; Vaswani et al., 2017; Dosovitskiy et al., 2020). In multimodal tasks, multiple converging networks are employed, one for each modality, mapping their inputs to a shared-dimensional latent space through projection layers at the final stage. This latent space is flat and non-gridded, making it well-suited for the application of contrastive loss.

For this work, I developed a CLIP-inspired architecture to map AGC-processed data stacks and their corresponding brain images into a shared latent space. In the context of conditioning LDMs, the goal is to leverage features from the brain images to guide the data embedding process. This approach shifts the responsibility of aligning these distinct modalities away from the LDM and assigns it to a dedicated network, thereby alleviating the optimisation burden on the main model during training. This strategy is particularly advantageous, as the stacked data and it corresponding image exist in vastly distant spaces, linked solely by the dynamics of the wave equation.

The two networks employed in this study are Visual Transformer architectures designed to map pairs of both modalities into a shared latent space. Each transformer

uses a patch size of 3×3 and consists of three residual transformer layers, each with a single attention head and an embedding dimension of 512. Layer normalisation is applied before and after each transformer layer to stabilise training and improve convergence. The final layer projects the representations into a latent space of size $1 \times 64 \times 64$, ensuring consistent embedding dimensionality across the modalities.

To address memory constraints, the AGC-processed data stack was downsampled to 1000×1000 using an antialiasing filter. Training was performed on a cluster equipped with four NVIDIA A100-SXM4-40GB GPUs, employing data-distributed parallelism over 500 epochs, with a total training duration of approximately 30 hours. The model optimised the contrastive InfoNCE loss, defined in Equation 5.7.

The optimiser used was Adam, with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.98$, a learning rate of 1×10^{-4}, a batch size of 32, SiLU (Elfwing et al., 2018) activation, and a temperature parameter of 0.1 for the contrastive loss. Heuristic experimentation revealed that clipping outliers in the outer quartile of the AGC-stack data improved training losses and overall learning outcomes. Additionally, applying a logarithmic transformation to the input images, similar to the approach in Chapter 4, further improved the learning process.

The network effectively learns the alignment process, as evidenced by the behaviour of the logits for data-to-image and image-to-data embeddings shown in Figure 5.11. The logits matrix displays high values along the diagonal, representing true pairs, while the off-diagonal entries show consistently lower values, indicating the dissociation of negative pairs. This distinct separation highlights the model's ability to correctly align data and image representations in the shared latent space.

This alignment is further visualised through two-dimensional projections of the embeddings using PCA and UMAP in Figure 5.12. The top row shows raw data and image representations projected into 2D space as a baseline. In the bottom row, the aligned embeddings are presented, with the average Euclidean distances between paired embeddings ($\bar{\delta}$) calculated to quantify alignment. The comparison reveals a significant reduction in pairwise distances within the embedding space, demonstrating that the network has learned to aligned data and image representations, preserving their intrinsic relationships within the shared latent space. A compre-

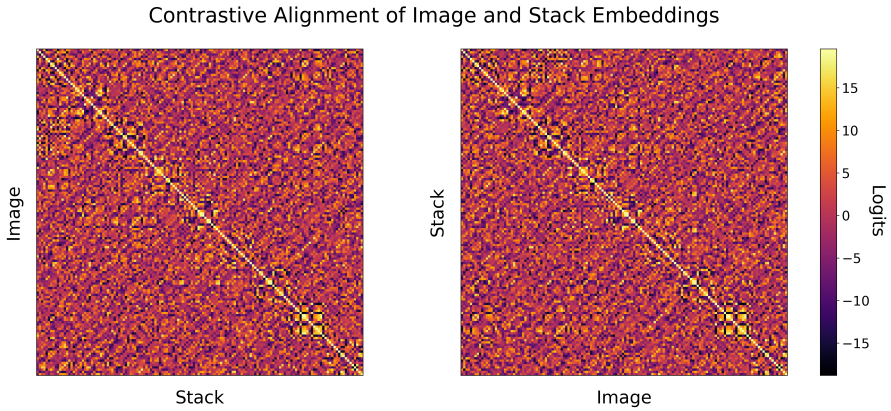


Figure 5.11: Logits of the contrastive loss for the shared embedding space of 256 AGC-stack and image samples. The matrix on the left represents data-to-image alignment, while the matrix on the right corresponds to image-to-data alignment, reflecting the two terms in Equation 5.7. High values along the diagonal represent correctly aligned data-image pairs, while low values on the off-diagonals correspond to negative pairs, indicating cross-modal alignment within the learned latent space.

hensive description of the hyperparameters used, along with a comparison to other trained models, can be found in Appendix C.

The continuity property of the latent space in contrastive learning, where similar samples reside closer together, is illustrated in Figure 5.13. Brain images with similar anatomical structures and their corresponding AGC-stacks are embedded in close proximity, while more distinct samples appear further apart. This relationship is depicted by comparing three samples with high, medium, and low cosine similarities (sim) to a base sample (top), alongside their average relative mean squared error ($\bar{\delta}_{i,j}$), which increases as cosine similarity decreases. This structured organisation enhances the utility of the embeddings by enabling more accurate model guidance. The spatially coherent latent representation makes the relationships between images and data more easily interpretable, potentially allowing the model to better focus on connect corresponding features.

A key extension of this property is its potential to support clinical generalisation. With similar brains residing near each other in the latent space, it becomes possible for deformations, particularly non-rigid transformations, to retain proximity to the original brain embeddings. This could enable the network to develop invariance to variations in acquisition geometry, such as rotations, translations, and other

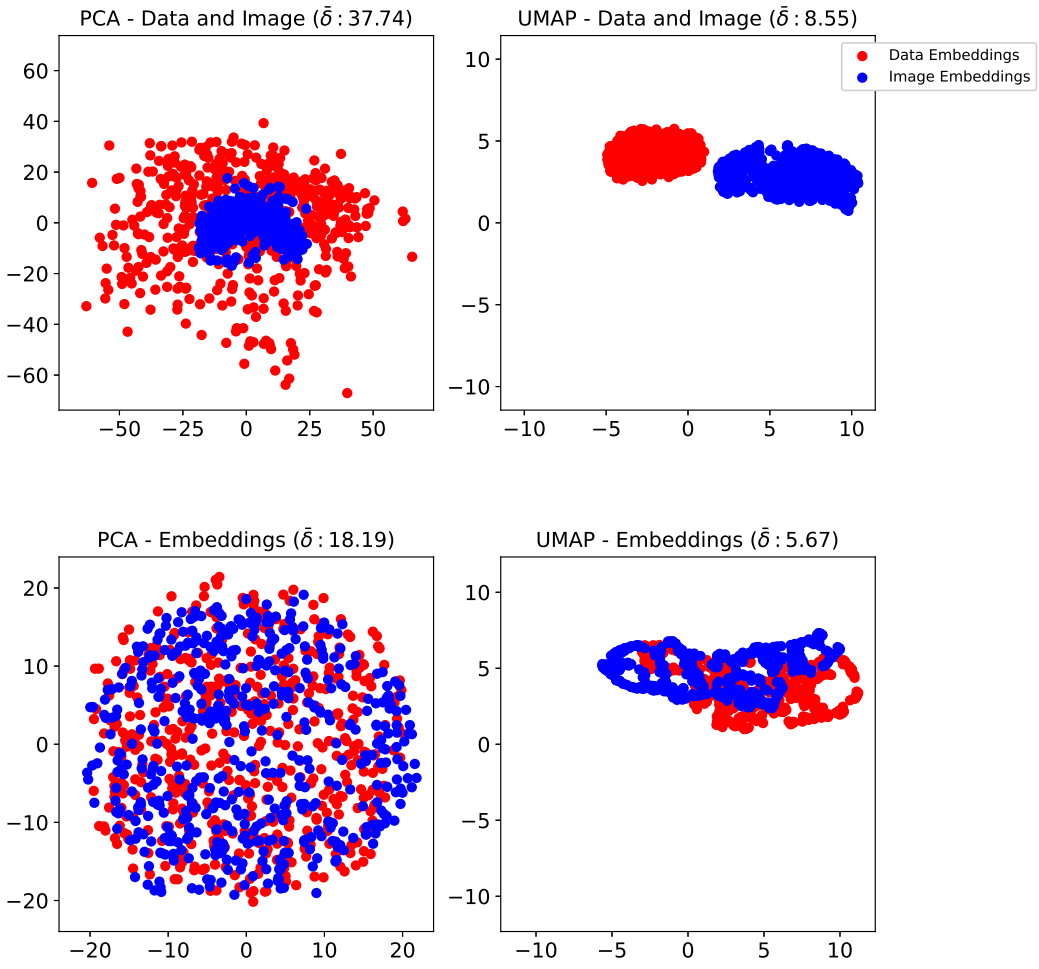


Figure 5.12: Two-dimensional projections of the embeddings of 512 paired samples using PCA (left) and UMAP (right) to visualise the alignment process. In the top row, raw data (red) and image (blue) representations are projected into 2D space as a baseline. In the bottom row, aligned embeddings of data and images are shown, with the pairwise average Euclidean distances ($\bar{\delta}$) calculated to quantify alignment. The drastic reduction in pairwise distances in the embedding space demonstrates that the network successfully aligned data and image representations in a shared latent space. The UMAP optimisation used is parameterised a minimum distance of 0.1, 15 neighbours, Euclidean metric, 2 components, and a learning rate of 1, with 512 samples.

transformations.

Figure 5.14 illustrates this invariance, even in the absence of explicit invariance constraints during optimisation. Translations up to 30 mm and rotations up to 10 degrees maintain high cosine similarity with the original AGC-stack embeddings. Interestingly, the cosine similarity for rotations between 10 and 45 degrees

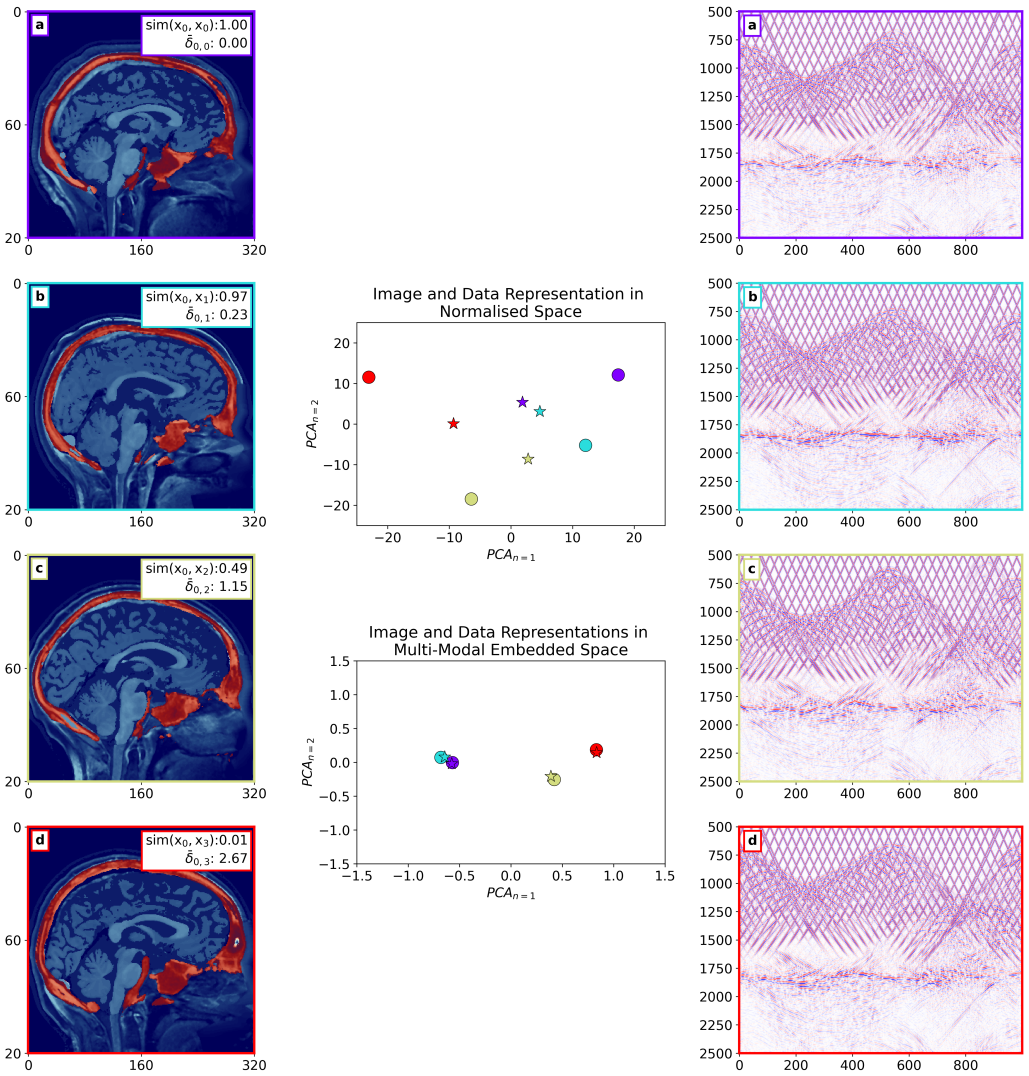


Figure 5.13: Demonstration of the continuity of the latent space in contrastive learning, where brain images with similar anatomical structures and their associated AGC-stacks are embedded closer together, while more distinct samples are positioned further apart. The figure compares three samples with varying cosine similarities (sim) to a base sample (a), displaying a high (b), medium (c), and low (d)similarity. The average relative mean squared error ($\bar{\delta}_{i,j}$) increases as the cosine similarity decreases. This spatial arrangement within the latent space facilitates structured alignment of similar samples, where the latent disentanglement represents meaningful relationships across image and data.

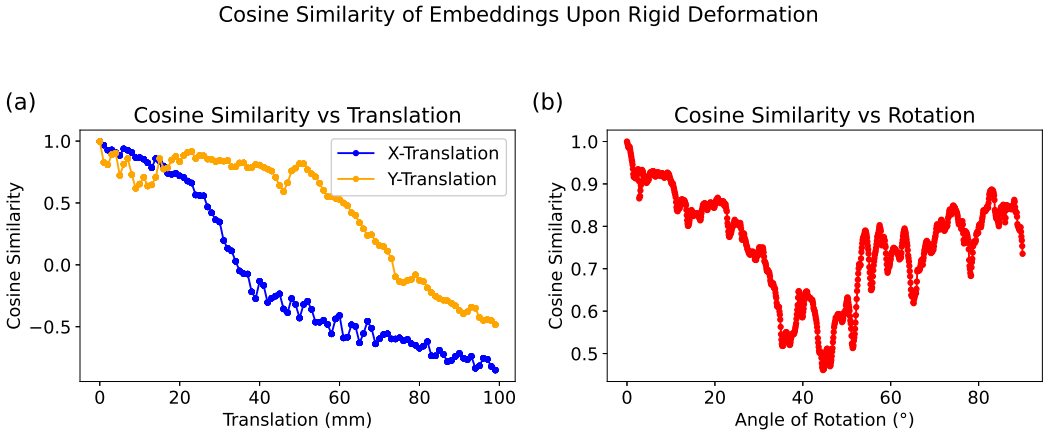


Figure 5.14: Demonstration of the network's potential for clinical generalisation by visualising cosine similarity of embeddings under rigid transformations of translation and rotation. In (a), cosine similarity is plotted against translations, showing that similarity remains high up to approximately 30 mm, beyond which it gradually declines. In (b), cosine similarity is shown relative to rotational angles, with notable similarity retained up to ∼10 degrees. Interestingly, cosine similarity decreases linearly for rotations between 10 and 45 degrees but increases for rotations between 45 and 90 degrees, suggesting the model naturally aligns small and near-perpendicular transformations. These results suggest that the latent space continuity enables the model to maintain alignment between similar brain embeddings under moderate translations and rotations, which can be beneficial in clinical applications.

decreases linearly, while for rotations between 45 and 90 degrees it increases linearly, suggesting that the model can more naturally align small and near-perpendicular transformations.

This preliminary exploration supports the continuity of the latent space under these transformations, though further validation of this generalisability in generation tasks remains untested. Enhanced invariance might be achieved through targeted data augmentation, as proposed by Chen et al. (2020), and explicit invariance constraints, which could further improve the model's invariability to acquisition irregularities and variable designs.

5.4 Results and Discussion

Throughout this research, various frameworks and methodological refinements were explored to implement the LDM proposed in Figure 5.1, designed to generate ultrasound-conditioned brain images. Beginning with simpler approaches, the methodology pro-

gressed through increasingly complex embedded representations, including resampled ultrasound stacks, PCA-derived embeddings, VAE latent vectors, and ultimately, contrastive embeddings. Models with varying depths, parameterisations, and architectural configurations were tested to maximise the potential of each embedded representation, pushing available hardware to their limits.

To maximise guidance in generative sampling, denoising diffusion implicit models (DDIMs) were employed for their deterministic nature. Conditioning was integrated into the network through various techniques, including addition, concatenation, and cross-attention of the embedding representations, each aimed at precisely guiding the generative process with ultrasound data. Additionally, an approach involving joint training of the embedding network with the LDM was explored to evaluate whether co-optimisation could yield improved results.

Despite these comprehensive efforts, the model consistently failed to generate brain images reflecting the conditioning data, producing outputs indistinguishable from those of the unconditional LDM. This outcome suggests that the embedded representations, though statistically validated, did not exert a discernible influence on the diffusion process, raising questions about the compatibility of these conditioning strategies with the diffusion model framework and the overall methodology adopted.

The rationale behind employing a conditional LDM for generating ultrasound-guided brain images was theoretically grounded in recent advancements in diffusion modelling within medical imaging. The success of prior studies in brain imaging, such as that of Pinaya et al. (2022), demonstrated the effectiveness of diffusion models for data-conditioned generation. Their framework achieved reliable outputs utilising simpler metadata conditions to guide image synthesis. This set a precedent to indicate that an LDM, if conditioned on appropriate and meaningful embeddings of ultrasound data, could be configured to produce consistent, and accurate outputs. The speculation was that, even if perfect accuracy was not immediately achieved, the model would exhibit preliminary signs of success, such as coherence in low-frequency structures (e.g skull shape), which could then be incrementally refined with sophisticated correction approaches for high-resolution fidelity.

The feasibility of synthesising images guided by medical data has also been

validated in work by Scotti et al. (2024), who employed contrastive loss to align fMRI data with image embeddings in a shared latent space, using the aligned embeddings as a diffusion prior for image reconstruction directly from fMRI. This study provided a strong basis for using contrastive embeddings in LDMs conditioned by complex data types, reinforcing the plausibility of conditioning the LDM on ultrasound representations.

From a technical perspective, contrastive embeddings were chosen to improve alignment between the ultrasound data and image representations based on their success in image generation models with similar conditioning requirements (Ramesh et al., 2021, 2022; Rombach et al., 2022). Continuous contrastive embedding spaces allow for soft clustering rather than strict class-based compartmentalisation, promoting representational continuity of these conditions that should in theory suit complex, multi-modal generation in diffusion models. Contrastive embeddings should enhance the model's ability to "understand" the connection between modalities by retaining semantic relationships in the embedded space.

Similarly, the methodological foundation of my approach is inspired by the work of Song et al. (2021a) on solving inverse problems in medical imaging with score-based generative models. In their work, Song et al. (2021a) successfully implemented an unsupervised generative framework that reconstructs medical images from partial measurement data by conditioning the generative process on these measurements. A notable distinction in their approach is the incorporation of data measurements directly into the score function, a component intrinsic to score-based diffusion models, which guides the generative process. Additionally, the authors integrated data measurements into the training loss, further constraining and conditioning the reverse pathways, an element that, if time had permitted, would have been valuable to explore in this work. Although, unlike the current study, Song et al. (2021a) focused on solving inverse problems for MRI and X-ray CT data, which have a linear relationship to the modelled image (Liang and Lauterbur, 2000; Hermanek et al., 2018). Having the wave equation as the underlying physics linking the data to the brain images, the complexities which the model must learn are considerably higher.

As a point of interest, initial — and unrecoverable — versions of the code pro-

vided preliminary evidence supporting the viability of this approach, as early tests demonstrated basic resampled stack conditioning effects, generating skull structures similar to the input samples. However, subsequent debugging and adjustments disrupted these results, suggesting that while the foundational model showed potential, it would benefit from thorough investigation and more rigorous testing. This preliminary outcome may indicate that further refinement and troubleshooting of the current code could be key to understanding the model's failure in guided generation.

It is inherently challenging to analyse the internal workings of a neural network, particularly in tasks involving latent variables and embedded data representations that are associated to physical laws. As data is embedded and propagated through the network, it undergoes transformations that abstract much of the physical meaning inherent in the original inputs. Consequently, determining precisely how the network interprets and utilises the conditioning data becomes difficult, often rendering the generation process opaque and complicating efforts to assess or improve the effectiveness of conditioning.

Nevertheless, it is possible to draw on insights from existing work to make educated guesses about factors that may have impacted the model's performance and explain why the conditioning may have failed. One potential reason for the lack of successful conditioning could be the limited capacity of the LDM network itself. The complexity and high non-linearity of the ultrasound data may have exceeded the representational capabilities of the current network architecture, despite the computational exhaustion. Large scale implementations of diffusion models for high-resolution synthesis usually contains billions of trainable parameters (Podell et al., 2023; Rombach et al., 2022), compared to the $\sim$5 million parameters used in this work. Additional layers and attention modules could enhance the model's ability to capture the complex relationships between the embedded ultrasound data and the generated brain images. Without sufficient depth and focused attention on the conditioning data, the network might have struggled to learn the necessary mappings, failing to guide the generative process.

Another factor potentially limiting the success of the conditioning is the size of the training dataset (N=1,000). The number of data samples may have been

insufficient for the network to effectively learn the guidance. While the unconditional LDM demonstrated the ability to capture the distribution with limited samples, the additional complexity introduced by conditioning on ultrasound embeddings may require a larger dataset to enable meaningful learning of the underlying image-data relationships.

In terms of embeddings, particularly those derived from contrastive approaches, this study's findings in Section 5.3.5 indicated that the contrastive embeddings produced a well-structured and semantically meaningful latent representation. Nevertheless, embedding design is a complex and intricate task, requiring extensive study and consideration in its own right. Ensuring that an embedding network adequately captures meaningful, statistically diverse and domain-specific representations is non-trivial. As seen in Figure 5.12, despite the significant decrease in pairwise distance, the distance remains non-zero, suggesting that further reduction may be possible. It remains possible that the embedding network's capacity was insufficient to achieve the depth of feature learning needed for effective conditioning.

One major consideration is that contrastive networks are highly data-intensive, often requiring billions of data samples (Radford et al., 2021; He et al., 2020), and the smaller dataset used in this work may have limited the network's ability to *fully* capture discriminative semantics across samples. Supporting this notion, Chen et al. (2020) demonstrated that contrastive learning benefits greatly from the application of strong, varied data augmentations. The authors explained that such augmentation introduces artificial variability, helping the model focus on meaningful semantic features rather than superficial correlations, which is particularly crucial when training with limited data.

Contrastive learning also typically requires large batch sizes to maximise representation quality — 8192, as reported by Chen et al. (2020), and even up to 32,768 in Radford et al. (2021). For high-dimensional data such as ultrasound embeddings, achieving these batch sizes is practically unfeasible without substantial hardware resources. To overcome this limitation, alternative approaches would need advanced GPU optimisation strategies to support the scale required for improved contrastive embedding learning in this context.

Another limitation could lie in the absence of a physics-based constraint within the embedding process. Without leveraging the underlying physical model linking ultrasound data to image generation, learning a shared latent space between these domains becomes highly abstract and complex. Incorporating known physics constraints, as demonstrated in Song et al. (2021a), could help the network learn better representations by anchoring it to meaningful patterns in the data. This approach could potentially aid in disentangling useful information in the embedded space and facilitate conditioning in the LDM.

It is also possible that the visual transformer architecture, which segments images into spatial patches for processing, may not be the optimal approach for representing ultrasound data, which has both temporal and spatial dimensions. Unlike typical visual data, ultrasound sequences contain temporal dynamics that are integral to the data's overall structure and interpretation. Effectively incorporating temporal elements would typically involve processing the data as sequential vectors, which, however, poses a greater computational challenges in transformer-based architectures due to increased memory requirements.

Ultimately, the core methodology of using ultrasound data as a direct conditioning input, without further considerations beyond embeddings, could potentially be too simplistic for the complexity of the task. The link between ultrasound data and the desired brain images is deeply intricate due to the dynamics of wave propagation, as outlined in Section 2.2, and may require a more advanced approach to fully guide the generation process. A more effective design might involve a customised loss term to better enforce consistency between the generated images and the conditioning data, as seen in Song et al. (2021a), where a data-driven loss function is used to regularise and steer the output toward the correct sample. Alternatively, incorporating physics-based guiding mechanisms or designing hybrid approaches from other methods in the field, such as those in Section 2.2.2, could facilitate better integration of the conditioning data, enabling the model to align ultrasound signals more closely with the generated images.

While these challenges were considerable, the extensive efforts invested in this project offer valuable insights for future research. Each obstacle and empirical finding

here brings to light key factors to consider in developing more tailored approaches, paving a clearer path forward for generative models in ultrasound imaging. This work not only reveals the complexities involved in conditioning generative models on ultrasound data but also provides a solid foundation that could support breakthroughs in data-informed generation, bringing us closer to practical, reliable applications in clinical settings.

5.5 Future Work

An important preliminary step in further work would involve developing a more rigorous testing strategy to ensure the conditional LDM model accurately learns the intended features and that the conditioning mechanism is functioning as expected. One approach would be to test the model on simpler conditions, such as metadata-based conditioning, as demonstrated by Pinaya et al. (2022). By first validating the conditional implementation on such straightforward conditions, it would be possible to verify whether the model responds correctly to conditional cues and rule out the possibility that the observed failure is due to undetected bugs or implementation issues. This initial testing phase would allow for refinement of the conditional setup in a controlled environment, reducing ambiguity and ensuring that any adjustments made later are based on a more stable, functioning codebase.

Another important development of this work would be to use the unconditional LDM presented in Chapter 4 to expand the training dataset of the conditioned LDM, potentially mitigating the lack of conditional guidance due to limited variability. The unconditional LDM could also be used to generate additional paired samples for the embedding model, providing better variability and greater disentanglement of the data-image relationships.

Given the results and discussions presented in this chapter, I strongly believe that improving the embeddings represents the most significant area for further work. A priority would be implementing well-designed data augmentation strategies, as shown to severely impact contrastive learning in Chen et al. (2020). This would involve rigid transformations to increase the diversity within the dataset and allow the

embedding model to learn a more semantic, but also more continuos representation through different input variations. This could be further imposed in the network by adding a regularisation term that enforces consistency among these variants.

Another useful augmentation approach would involve expanding the dataset by reusing the current images with varied acquisition geometries, which I also theorise would improve the generalisability of the embedding model. This strategy could improve the model's understanding of different acquisition setups, a practical advantage for adapting to diverse clinical conditions. Additionally, associating each image with multiple variants of the data exhibiting different levels of sparsity could further enhance the generalisation capacity of the model. This approach, functioning similarly to a dropout method, would also help the model adapt to scenarios with missing or low-sampled data, simulating conditions encountered in low-resolution or resource-limited acquisition environments.

Extending this regularisation further, the embedding model could be trained to also achieve invariance to wave-equation parameterisation by being exposed to modelling that incorporate elasticity dynamics and accounts for anelastic losses. Incorporating noisy data and enforcing consistency between noisy and clean embeddings would also be beneficial, promoting robustness to real-world variability in acquisition conditions and enhancing the model's adaptability to unpredictable clinical scenarios.

It should be noted, however, that such augmentation strategies would increase the demand for computational resources; with this small dataset, training embeddings took approximately 30 hours parallelised on four A100 GPUs, suggesting that larger-scale augmentation would require more powerful compute or suffer from longer training times.

Drawing inspiration from Song et al. (2021a), another direction for future work would be the development of a customised loss term designed to directly regularise the conditioning of the diffusion model, ensuring that generated images remain consistent with the conditioning data. Such tailored loss could provide better conditioning guidance to the model's training process and reinforce alignment with the conditioned information, providing more targeted feedback to increase the model's

responsiveness to the conditional data throughout generation. Along these lines, the work by Nichol et al. (2021) suggests perturbing the reverse-process mean that founds diffusion models with the gradient of the dot product of the image and the model condition, which is an interesting approach to explore.

If the model had succeeded, two key speculative testing strategies could have provided deeper insights into its performance and potential for clinical application. First, given the high-frequency synthesis capability of latent diffusion models, it would be worth exploring the model's ability to generate high-resolution brain images in the absence of low-frequency information, potentially overcoming a prominent problem of FWI. Although this was not tested due to the model's current limitations, such analysis could reveal how well the model compensates for missing low-frequency components and the extent to which it can retain image fidelity under these conditions. Second, rigorous testing strategies would be have to be put in place to assess the fidelity and generalisability of the model's outputs, particularly as this fidelity is of critical importance for any potential clinical applications. In the context of probabilistic generative modelling, tests would need to evaluate consistency and accuracy across generated samples to ensure that the model's outputs meet the reliability standards required in medical imaging and can, in full, accurately recover fine structures of the brain that might be associated with any pathologies.

As an extension of this work, future research could explore further an alternative approach inspired by the architecture of DALL-E Ramesh et al. (2022), which has demonstrated notable success in generating high-resolution images from complex, multi-modal pairings. While LDMs and DALL-E share the goal of generative image synthesis, DALL-E employs a distinctive methodology that leverages diffusion models in combination with multiple specialised networks, each tasked with a specific sub-problem. This architecture enables DALL-E to capture and manage the intricate relationships between modalities more effectively. DALL-E primarily differs from the original LDM architecture in that it introduces an intermediate prior network to generate image embeddings from text embeddings, bridging the text-image relationship in a complementary manner to the embedding network. The model comprises three separately trained networks:

1. **CLIP Network**: This transformer-based network is trained to align text and image embeddings within a shared latent space, allowing for coherent associations between the modalities.

2. **Prior Network**: This diffusion model is conditioned on the text captions to produce an image embedding that aligns with the caption's semantics. This step is important because even though the text and image embeddings align, they are not equal. The prior network serves as a constraint, ensuring that the generated image embedding reflects the style and content of the text prompt, while also controlling the variability of generation within the text prompt (Ramesh et al., 2022).

3. **Decoder Network**: Also a diffusion model, the decoder network is conditioned on the generated image embedding from the prior network. It translates this embedding into a high-resolution image, and due to the alignment, it adheres to the caption's essential semantic and stylistic elements. This layered conditioning approach allows the decoder to focus on refining details and maintaining fidelity to the input while permitting variation in secondary aspects.

The process for generating an image in this architecture is structured: the text caption is first encoded into an embedding by the CLIP network, capturing the core semantics. This text embedding then serves as a condition for the prior network, which produces an intermediate image embedding that carries the intended visual characteristics. Finally, this embedding is passed to the decoder network, which reconstructs the high-resolution image based on the prior's output.

The authors of DALL-E (Ramesh et al., 2022) demonstrated that, without a prior, the generated images exhibit reduced semantic alignment with the input text and lower coherence in visual quality. This is quantitatively reflected in the FID scores, where the score improves from 16.55 to 9.16 with the use of a prior. Additionally, human evaluation shows that participants prefer images generated with the prior $\sim$57% of the time for photorealism and $\sim$53% for caption similarity, underscoring the role of the prior in enhancing both semantic fidelity and visual coherence.

Applying this multi-step framework to ultrasound image generation could pro-

vide a promising path forward. In this adapted model, the CLIP network would serve as the data-to-image embedding model, similar to the one used in this chapter's work. Here, text tokens would be replaced by ultrasound data to condition a prior network that generates corresponding brain image embeddings. This step could be viewed as a corrective term to the embedding network, where the remaining pairwise distances, as seen in Figure 5.12, would be resolved by the prior network. To ensure a deterministic image-to-data embedding mapping, the prior network could be simplified to a U-Net model, or equivalent. The decoder network would then take these image embeddings from the prior as input, producing the final high-resolution brain images. While such architectures have primarily been demonstrated at a large scale with large datasets, adapting this approach to smaller datasets remains an open challenge. Nonetheless, this extension could provide a compelling avenue for improving conditioning and generating high-fidelity ultrasound-based brain images with diffusion models.

Since the completion of this thesis, a new paper has been released by Wang et al. (2024) (currently pending revision for ICLR 2025), demonstrating that the joint distribution of acoustic data and velocity maps can be effectively modelled using a diffusion framework, closely following the structure of LDMs. In this work, the authors employ a dual autoencoder architecture — one encoder for each modality — to map ultrasound data and corresponding velocity images into a shared latent space, using CNNs and element-wise metrics instead of transformers and contrastive losses. Following this, they train an unconditioned LDM as a prior network that refines these coarse latent representations from the data, which are subsequently decoded to produce the final velocity maps. Their approach enables flexible mappings from image-to-data and data-to-image, that largely satisfies the wave equation (Wang et al., 2024). This recent work validates the direction taken in this thesis, showing that the ideas developed here align well with emerging advancements in the field. It is encouraging to see other researchers exploring similar approaches and achieving promising results, suggesting that generative models may indeed hold the key to tackling inverse problems in medical imaging. If these models can be made resilient enough for real-world deployment, they have the potential to bring

about meaningful changes in clinical imaging, opening the door to more accurate, accessible, and transformative diagnostic tools for the future.

5.6 Summary

In this research chapter, I proposed an ambitious methodology using latent diffusion models to establish a near-real-time and reliable process for data-to-image ultrasound tomography, replacing the full-waveform inversion (FWI) pipeline. The methodology was inspired by recent advancements in diffusion models conditioned on simpler metadata, which have shown remarkable success in computer vision and medical imaging. The goal was to demonstrate that a well-designed network could embed ultrasound data in a way that effectively guides the LDM to generate brain images that consistently align with the associated data.

For that, various embedding techniques were explored to facilitate the transformation of ultrasound signals into compact forms suitable for conditioning the diffusion model. These techniques include principal component analysis, variational autoencoders, and contrastive representation learning, each chosen for its potential to capture essential features of the high-dimensional ultrasound data that could enable guidance and conditioning within the LDM framework.

In particular, embeddings through contrastive learning were thoroughly explored in this chapter, inspired by the current state-of-the-art developments in guided generative modelling. The embedding model presented here successfully created latent representations that grouped similar anatomical features of the brain and ultrasound data together in a continuous, semantically meaningful space — an achievement that, to the best of my knowledge, has not been previously applied in the context of ultrasound imaging. While this study does not exhaust the possibilities for embedding techniques, it offers a promising foundation for future research, suggesting that a well-designed embedding model has the potential to capture complex relationships between ultrasound data and the physical models that generated them.

Despite these advanced embedding strategies, the model encountered challenges in accurately guiding the network towards reconstructing brain images from ultra-

sound data, resulting in outputs that did not fully align with the intended conditioning. Several factors are suggested to have influenced the model's capacity to leverage the conditioning data effectively, including limitations in model capacity, the relatively small dataset size, and the complexity of the embedding tasks.

Reflecting on these outcomes, both the potential and current limitations of using LDMs for ultrasound-guided brain imaging become evident. Future work could benefit from refining embedding strategies, implementing more robust data augmentation techniques, and exploring more advanced diffusion frameworks inspired by architectures such as DALL-E to overcome the encountered challenges.

Looking ahead, this research opens the door to a new approach in high-resolution brain ultrasound inversion. By eliminating the need for a predefined starting model, this method enables the network to learn and apply a deep understanding of brain anatomy directly from training data, capturing natural and accurate anatomical distributions without relying on often inaccessible *a priori* information. With a model dedicated to crafting intelligent latent representations of the data, this approach could also overcome the low-frequency limitations that constrain traditional FWI, while adapting to varied acquisition equipment and environments. Most importantly, this setup shifts computational demands to the pre-clinical training phase, enabling quick, efficient inference in clinical settings. Such a system could deliver portable, near-instant imaging and diagnosis, radically transforming clinical practices. The potential for real-time, point-of-care monitoring could greatly expand ultrasound's role in both emergency and routine care, supporting clinicians in delivering timely, precise diagnostics for a broader patient population.

Chapter 6

Conclusions

The human brain, with its intricate structure and essential role in cognitive and physiological functions, has long challenged the scientific community to develop viable methods for non-invasive imaging. Despite advancements in MRI, CT, and PET scans, limitations persist in terms of cost, accessibility, and patient safety, particularly in scenarios requiring rapid diagnosis and intervention. Among imaging modalities, ultrasound tomography (UT) presents promising advantages due to its portability, safety, and real-time imaging capabilities. Yet, conventional UT lacks the computational performance needed to achieve the resolution and tissue differentiation necessary for accurate brain characterisation. Recent advancements in Full-Waveform Inversion (FWI) have sought to address these limitations by leveraging complete waveform data to capture more intricate wave dynamics, enabling, for the first time, the imaging of fine structures of the brain using ultrasound data alone. Nonetheless, FWI introduces significant computational and practical challenges, driven largely by the complex and costly modelling required to account for the acoustic contrast between the human skull and intracranial tissues. This thesis investigates the integration of advanced generative deep learning models, particularly diffusion models, to address these challenges. By modelling the intricate relationship between ultrasound data and underlying brain structures, this work sought to extend the clinical utility of UT, particularly in critical and point-of-care settings where safe, reliable, and rapid brain imaging is essential.

6.1 Summary of Contributions

The first contribution of this thesis is an in-depth literature review that I conducted to meticulously explore the principles and current research in the fields of deep learning and acoustic imaging. In this review, I provide a comprehensive study of fundamental deep learning concepts and present a thorough examination of common loss functions, focusing on their mathematical formulations and their significance in generative modelling. I give detailed coverage to prominent generative architectures, including variational autoencoders, generative adversarial networks, diffusion models, and transformers, highlighting the mathematical principles that drive these models, and discussing key metrics for quality assessment. Finally, the literature review introduces ultrasound tomography and provides a thorough exploration of the mathematics and heuristics of FWI as an ultrasound processing technique, as well as the specific challenges it poses.

Following the literature review, I presented three research chapters, each designed to address a distinct sub-problem within the overarching aim of ultrasound brain imaging with diffusion models. I have made all developed models available as open-source software to promote reproducibility and encourage contributions from the scientific community, with links to the code repositories provided in the respective chapters. The modular implementation of each chapter allowed for independent validation, tailored in-depth discussions, and the promotion of further work. Their contributions are summarised below.

In Chapter 3, I tackled the limitations of traditional data comparison methods in ultrasound tomography, focusing specifically on the need for high-resolution synthesis in medical imaging. Existing metrics, such as Mean Squared Error (MSE), often fail to preserve high-frequency details and structural fidelity, both of which are essential for accurate brain imaging in ultrasound. To address these shortcomings, I proposed a novel metric, the Wiener Loss, using convolutional Wiener filters to encode contextual dependencies and promote full-broadband frequency recovery. The Wiener loss is developed from a least-squares framework, preserving the smoothness, differentiability and computational efficiency of MSE. I validated the metric

across multiple domains, with a primary focus on medical imputation tasks, where Wiener Loss notably improved the fidelity of brain scan reconstructions by preserving high-frequency components of key anatomical structures with limited hallucination. These outcomes were further supported by experiments on standard datasets such as MNIST and CelebA, establishing Wiener Loss as a robust loss function capable of enhancing perceptual quality across various domains. The improved structural recovery achievable with Wiener Loss underscores its utility in clinical and diagnostic imaging, where precise detail preservation is integral to accurate assessment of pathologies and other high-resolution imaging needs. I further discussed how the Wiener Loss can be reframed in the context of a non-parametric diffusion model, a novel attention mechanism, and in tasks that require translational invariance.

In Chapter 4, I addressed the pervasive issue of data scarcity in brain ultrasound imaging, a key limitation hindering the implementation of deep learning models in this field. I proposed an approach for synthetic data generation using an unconditional Latent Diffusion Model (LDM), designed to expand and diversify the limited imaging dataset (N=1,000) of neuroacoustic images. The model leveraged the denoising process inherent in diffusion models and their strong inductive bias to generate high-quality, structurally accurate, and statistically conforming images across all three principal axes of the brain. Quantitative evaluation confirmed the variability and quality of the generated samples across a range of metrics, supporting the model's potential to augment the existing dataset despite its size constraints. Testing across each brain axis demonstrated that extending this methodology to 3D image synthesis is a feasible task. Additionally, integrating metadata, such as patient demographics, was identified as a promising next step to enhance the clinical relevance of the generated samples.

In Chapter 5, I extended the research from Chapter 4 to conditional LDMs, focusing on the goal of synthesising high-resolution, ultrasound-guided brain images as an alternative to the computationally intensive FWI approach. To address the high dimensionality and complexity of ultrasound data, I explored a variety of embedding and conditioning strategies aimed at enabling the LDM to generate brain images conditioned on ultrasound data. I tested several embedding methods

including principal component analysis, variational autoencoders, and contrastive representation learning, to transform ultrasound data into a manifold compatible with the LDM's diffusion process. The contrastive representation learning approach yielded particularly promising results by producing embeddings that were physically meaningful. However, embedding design remains a complex and nuanced task that warrants dedicated research. This initial progress suggests that a carefully crafted embedding model could effectively capture intricate relationships between ultrasound data and the physical models that generate it, potentially incorporating an implicit understanding of wave propagation dynamics. Despite extensive efforts, none of the conditioning strategies produced outputs that accurately reflected the conditioning ultrasound data; instead, the generated images resembled those from an unconditional LDM. While the intended conditional generation was not realised, I presented a thorough reflection on the findings and proposed areas for future work, including physics-based guidance, data augmentation, and more advanced architectures, each offering potential avenues for advancing ultrasound-guided image synthesis using diffusion models.

6.2 Future Work

In each chapter, I provide a detailed discussion of potential directions for future research specific to that study. Here, I will summarise the primary areas for further work that emerged from Chapter 5, as it addresses the most significant challenges related to the main goal of this thesis: performing ultrasound tomography using generative models.

Findings from Chapter 5 suggest several promising directions for future work. Developing a rigorous testing strategy will be essential for ensuring that the current implementation of the conditional LDM captures intended features while responding to conditional cues. Testing with simpler metadata-based conditions, as seen in related works, could verify the conditional setup and help identify potential bugs or issues.

Implementing data augmentation methods could enhance model performance,

as state-of-the-art models have been shown to benefit from increased data diversity due to their data-intensive nature. Using the unconditional LDM from Chapter 4 and applying rigid transformations to expand the dataset is a promising approach, as it may also improve the embedding model's ability to learn more continuous and semantically rich representations.

It is possible that the current embedding representations were insufficient to adequately condition the network in this complex problem, making improvements to the embedding model a key direction for future work. For instance, incorporating physics-based guidance, particularly techniques that integrate wave propagation dynamics, could reinforce the relationship between ultrasound signals and the generated brain images, potentially enhancing both the quality of the embeddings and the accuracy of the diffusion process. Further embedding improvements could involve training the model for invariance to ultrasound-specific variations, such as different acquisition geometries, elasticity dynamics, and anelastic losses, while also enforcing consistency between noisy and clean data embeddings to increase robustness against the variability encountered in clinical data.

Lastly, architectural extensions inspired by multi-modal frameworks, such as the intermediate prior network in DALL-E, could provide an enhanced conditioning pathway for generating reliable images, aligning the model's output more accurately with complex conditional data inputs. Collectively, I believe these approaches would bring us closer to achieving brain ultrasound imaging within the diffusion model framework, making reliable and high-quality results more attainable in this challenging problem.

6.3 Impact & Final Remarks

With the rapid growth of research in deep learning and generative modelling, alongside the increasing accessibility of advanced technologies, the landscape of scientific possibilities is evolving faster than ever. This research embodies that shift. It has aimed to reach a goal once so inconceivable that researchers had not yet considered exploring. As I complete this thesis, I realise that the approaches developed here

may themselves lay the groundwork for methods that could eventually surpass and replace them, turning it obsolete. This prospect fills me with optimism.

When I began this PhD in 2019, generative adversarial networks were regarded as the forefront of generative modelling, dominating discussions and aspirations in the field. Today, while GANs remain relevant, their place has been challenged by the emergence of new architectures like diffusion models, which continue to reshape the field of generative AI in ways we could not have anticipated. I hope and expect that the work I have done here will be part of that same progression, evolving in ways that could help improve healthcare delivery and, in time, bring meaningful changes in how we approach patient care.

Medical imaging is essential in clinical practice, offering non-invasive methods to diagnose a broad spectrum of conditions, from minor illnesses to life-threatening diseases. In this research, I focused specifically on ultrasound imaging, a modality with unique advantages, including portability, safety, and real-time imaging, benefits that are not readily available with other imaging techniques. However, ultrasound's strength in accessibility comes with a trade-off: its resolution and ability to differentiate tissue types are often limited, which constrains its diagnostic potential. For brain imaging in particular, ultrasound has historically been ruled out as a viable technology due to the computational challenges of accurately modelling wave interactions between the skull and the soft tissues inside it. This research builds on prior works to address these challenges and move toward making brain imaging with ultrasound possible.

I leveraged the capabilities of deep learning to address the high non-linearity inherent in ultrasound data, which arises from the complex physics of wave propagation. By applying diffusion models, I used a probabilistic, data-driven generative approach designed to overcome the typical low resolution and structural infidelity limitations of ultrasound. My aim was to create an alternative imaging solution with the potential for widespread use in critical and point-of-care settings, where time, precision, and portability are essential. This technology could also become part of routine clinical checks, removing the need for specialised imaging rooms and making preventative care more accessible, particularly in a world where access to healthcare

is limited by socioeconomic disparities.

Within the framework I developed, this methodology could also be readily extended to support 3D image synthesis, which would enhance diagnostic utility within the medical community. Additionally, the embeddings I created hold considerable potential for broader applications of ultrasound data that are today hindered by computational constraints.

Despite the performance limitations of my proposed model, I expect that my work will have set a strong foundation to make this once unimaginable goal achievable. By providing a thorough and critical analysis, I hope to have outlined a clear path for myself or other researchers to build upon and succeed in this area in the future. Ultimately, I have aimed for my research to make a meaningful impact on healthcare delivery and to influence how we approach and think about emerging medical technologies in the years to come.

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Ackley, D. (1987). *A connectionist machine for genetic hillclimbing*. Kluwer Academic Publishers.

Aganj, I., Yeo, B. T. T., Sabuncu, M. R., and Fischl, B. (2012). On removing interpolation and resampling artifacts in rigid image registration. *IEEE Transactions on Image Processing*, 22(2):816–827.

Ahmadian, S. and Khanteymoori, A. R. (2015). Training back propagation neural networks using asexual reproduction optimization. In *2015 7th conference on information and knowledge technology (IKT)*, pages 1–6. IEEE.

Aki, K. and Richards, P. G. (2002). *Quantitative seismology*. University Science Books.

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. (2022). Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

Alqahtani, H., Kavakli-Thorne, M., and Kumar, G. (2021). Applications of generative adversarial networks (gans): An updated review. *Archives of Computational Methods in Engineering*, 28:525–552.

Anderson, B. D. (1982). Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326.

Araya-Polo, M., Adler, A., Farris, S., and Jennings, J. (2020). Fast and accurate seismic tomography via deep learning. *Deep learning: Algorithms and applications*, pages 129–156.

Araya-Polo, M., Jennings, J., Adler, A., and Dahlke, T. (2018). Deep-learning tomography. *The Leading Edge*, 37(1):58–66.

Arjovsky, M. and Bottou, L. (2017). Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. *International conference on machine learning*, pages 214–223.

Arlot, S. and Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79.

Arnold, M. J., Jonas, C. E., and Carter, R. E. (2020). Point-of-care ultrasonography. *American family physician*, 101(5):275–285.

Arora, K. (2024). Everything you must know about data normalization in machine learning. https://www.markovml.com/blog/normalization-in-machine-learning?utm_source=chatgpt.com. [Accessed 22-12-2024].

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bahrini, A., Khamoshifar, M., Abbasimehr, H., Riggs, R. J., Esmaeili, M., Majdabadkohne, R. M., and Pasehvar, M. (2023). Chatgpt: Applications, opportunities, and threats. In *2023 Systems and Information Engineering Design Symposium (SIEDS)*, pages 274–279. IEEE.

Barger, J. E. (1979). Attenuation and dispersion of ultrasound in cancellous bone. *Ultrasonic Tissue Characterization*, 11:197–201.

Barratt, S. and Sharma, R. (2018). A note on the inception score. *arXiv preprint arXiv:1801.01973*.

Bates, O., Guasch, L., Strong, G., Robins, T. C., Calderon-Agudo, O., Cueto, C., Cudeiro, J., and Tang, M. (2022). A probabilistic approach to tomography and adjoint state methods, with an application to full waveform inversion in medical ultrasound. *Inverse Problems*, 38(4):045008.

Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M. (2017). Automatic differentiation in machine learning: a survey. *Journal of Marchine Learning Research*, 18(153):1–43.

Beck-Razi, N., Fischer, D., Michaelson, M., Engel, A., and Gaitini, D. (2007). The utility of focused assessment with sonography for trauma as a triage tool in multiple-casualty incidents during the second lebanon war. *Journal of Ultrasound in Medicine*, 26(9):1149–1156.

Ben-Kiki, O., Evans, C., and Ingerson, B. (2009). Yaml ain't markup language (yaml™) version 1.2. https://yaml.org/. Accessed: 2023.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305.

Bernard, S., Monteiller, V., Komatitsch, D., and Lasaygues, P. (2017). Ultrasonic

computed tomography based on full-waveform inversion for bone quantitative imaging. *Physics in Medicine & Biology*, 62(17):7011.

Berthelot, D., Raffel, C., Roy, A., and Goodfellow, I. (2018). Understanding and improving interpolation in autoencoders via an adversarial regularizer. *arXiv preprint arXiv:1807.07543*.

Bińkowski, M., Sutherland, D. J., Arbel, M., and Gretton, A. (2018). Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*.

Bishop, C. M. (1995). Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116.

Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, volume 4. Springer.

Blau, Y. and Michaeli, T. (2018). The perception-distortion tradeoff. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6228–6237.

Bond-Taylor, S., Leach, A., Long, Y., and Willcocks, C. G. (2021). Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models. *IEEE transactions on pattern analysis and machine intelligence*, 44(11):7327–7347.

Borji, A. (2019). Pros and cons of gan evaluation measures. *Computer vision and image understanding*, 179:41–65.

Borji, A. (2022). Pros and cons of gan evaluation measures: New developments. *Computer Vision and Image Understanding*, 215:103329.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine learning*, 45:5–32.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Brudfors, M., Balbastre, Y., Nachev, P., and Ashburner, J. (2019). A tool for super-resolving multimodal clinical mri. *arXiv preprint arXiv:1909.01140*.

Calderon Agudo, O. (2018). *Acoustic full-waveform inversion in geophysical and medical imaging*. PhD thesis, Imperial College London Department of Earth Science and Engineering.

Calderon Agudo, O., Guasch, L., Huthwaite, P., and Warner, M. (2018). 3d imaging of the breast using full-waveform inversion. In *Proceedings of the International Workshop on Medical Ultrasound Tomography*, pages 99–110.

Calderón Agudo, Ò., Vieira da Silva, N., Warner, M., Kalinicheva, T., and Morgan, J. (2018). Addressing viscous effects in acoustic full-waveform inversion. *Geophysics*, 83(6):R611–R628.

Cardoso, M. J., Li, W., Brown, R., Ma, N., Kerfoot, E., Wang, Y., Murrey, B., Myronenko, A., Zhao, C., Yang, D., et al. (2022). Monai: An open-source framework for deep learning in healthcare. *arXiv preprint arXiv:2211.02701*.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, pages 213–229. Springer.

Chan, W., Jaitly, N., Le, Q., and Vinyals, O. (2016). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4960–4964. IEEE.

Che, T., Zhang, R., Sohl-Dickstein, J., Larochelle, H., Paull, L., Cao, Y., and Bengio, Y. (2020). Your gan is secretly an energy-based model and you should use

discriminator driven latent sampling. *Advances in Neural Information Processing Systems*, 33:12275–12287.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29.

Child, R. (2020). Very deep vaes generalize autoregressive models and can outperform them on images. *arXiv preprint arXiv:2011.10650*.

Chong, M. J. and Forsyth, D. (2020). Effectively unbiased fid and inception score and where to find them. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6070–6079.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27.

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., and Bharath, A. A. (2018). Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65.

Crevier, D. (1993). Ai: The tumultuous history of the search for artificial intelligence. *Basic Book*.

Cruz, D. P., Strong, G., Bates, O., Cueto, C., Yao, J., and Guasch, L. (2023a). Convolve and conquer: Data comparison with wiener filters. *arXiv preprint arXiv:2311.06558*.

Cruz, D. P., Strong, G., Bates, O., Cueto, C., Yao, J., and Guasch, L. (2023b). Convolve and conquer: Data comparison with wiener filters. In *Medical Imaging meets NeurIPS Workshop*.

Cudeiro-Blanco, J., Cueto, C., Bates, O., Strong, G., Robins, T., Toulemonde, M., Warner, M., Tang, M.-X., Agudo, O. C., and Guasch, L. (2022). Design and construction of a low-frequency ultrasound acquisition device for 2-d brain imaging using full-waveform inversion. *Ultrasound in Medicine & Biology*, 48(10):1995–2008.

Cueto, C., Bates, O., Strong, G., Cudeiro, J., Luporini, F., Agudo, Ò. C., Gorman, G., Guasch, L., and Tang, M.-X. (2022). Stride: A flexible software platform for high-performance ultrasound computed tomography. *Computer methods and programs in biomedicine*, 221:106855.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314.

da Silva, F. L. (2013). Eeg and meg: relevance to neuroscience. *Neuron*, 80(5):1112–1128.

Dalca, A. V., Bouman, K. L., Freeman, W. T., Rost, N. S., Sabuncu, M. R., and Golland, P. (2017). Population based image imputation. In *International Conference on Information Processing in Medical Imaging*, pages 659–671. Springer.

Dalca, A. V., Guttag, J., and Sabuncu, M. R. (2019). Unsupervised data imputation via variational inference of deep subspaces. *arXiv preprint arXiv:1903.03503*.

Dao, T. (2023). Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*.

Dao, T., Fu, D. Y., Ermon, S., Rudra, A., and Ré, C. (2022). Flashattention: Fast and memory-efficient exact attention with io-awareness. *arXiv preprint arXiv:2209.11135*.

De Basabe, J. D. and Sen, M. K. (2015). A comparison of finite-difference and spectral-element methods for elastic wave propagation in media with a fluid-solid interface. *Geophysical Journal International*, 200(1):278–298.

de Choudhury, P., Zhu, Z., Nema, P., et al. (2021). Transformers are graph neural networks. In *Neural Information Processing Systems (NeurIPS)*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE.

Detlefsen, N. S., Borovec, J., Schock, J., Harsh, A., Koker, T., Liello, L. D., Stancl, D., Quan, C., Grechkin, M., and Falcon, W. (2022). Torchmetrics - measuring reproducibility in pytorch.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019a). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019b). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Dinh, L., Krueger, D., and Bengio, Y. (2014). Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2016). Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*.

Domingos, P. (2000). A unified bias-variance decomposition and its applications. In *Proceedings of the Seventeenth International Conference on Machine Learning*.

Dong, C., Loy, C. C., He, K., and Tang, X. (2015). Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307.

Dong, L., Xu, S., and Xu, B. (2018). Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5884–5888. IEEE.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Dosovitskiy, A. and Brox, T. (2016). Generating images with perceptual similarity metrics based on deep networks. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, pages 658–666.

Dozat, T. (2016). Incorporating nesterov momentum into adam. *ICLR Workshop*.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Dumoulin, V. and Visin, F. (2016). A guide to convolution arithmetic for deep learning. *ArXiv e-prints*.

Durbin, J. (1960). The fitting of time-series models. *Revue de l'Institut International de Statistique*, pages 233–244.

Duric, N., Littrup, P., Babkin, A., Chambers, D., Azevedo, S., Kalinin, A., Pevzner, R., Tokarev, M., Holsapple, E., Rama, O., et al. (2005). Development of ultrasound tomography for breast imaging: Technical assessment. *Medical Physics*, 32(5):1375–1386.

Elfwing, S., Uchibe, E., and Doya, K. (2018). Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11.

Esser, P., Rombach, R., and Ommer, B. (2021). Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12873–12883.

Fefferman, C., Mitter, S. K., and Narayanan, H. (2016). Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049.

Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., and Greenspan, H. (2018). Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification. *Neurocomputing*, 321:321–331.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Frogner, C., Zhang, C., Mobahi, H., Araya, M., and Poggio, T. A. (2015). Learning with a wasserstein loss. *Advances in neural information processing systems*, 28.

Fry, F. J. and Barger, J. E. (1978). Acoustical properties of the human skull. *The Journal of the Acoustical Society of America*, 63(5):1576–1590.

Giannakis, I., Giannopoulos, A., and Warren, C. (2019). A machine learning-based fast-forward solver for ground penetrating radar with application to full-waveform inversion. *IEEE Transactions on Geoscience and Remote Sensing*, 57(7):4417–4426.

Girod, B. (1993). What's wrong with mean-squared error? *Digital images and human vision*, pages 207–220.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings.

Golub, G. H. and Van Loan, C. F. (1996). Matrix computations johns hopkins university press. *Baltimore and London*, 41:62.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, volume 27, pages 2672–2680.

Grenander, U. and Miller, M. I. (1994). Representations of knowledge in complex systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 56(4):549–581.

Groos, L., Schäfer, M., Forbriger, T., and Bohlen, T. (2014). The role of attenuation in 2d full-waveform inversion of shallow-seismic body and rayleigh waves. *Geophysics*, 79(6):R247–R261.

Guasch, L., Calderón Agudo, O., Tang, M.-X., Nachev, P., and Warner, M. (2020). Full-waveform inversion imaging of the human brain. *NPJ digital medicine*, 3(1):28.

Guasch, L., Warner, M., Nangoo, T., Morgan, J., Umpleby, A., Stekl, I., and Shah, N. (2012). Elastic 3d full-waveform inversion. In *SEG International Exposition and Annual Meeting*, pages SEG–2012. SEG.

Hadsell, R., Chopra, S., and LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

Hagen, D. C. (1982). The application of principal components analysis to seismic data sets. *Geoexploration*, 20(1-2):93–111.

Hansen, P. C. (2002). Deconvolution and regularization with toeplitz matrices. *Numerical algorithms*, 29(4):323–378.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

He, Q. and Wang, Y. (2021). Reparameterized full-waveform inversion using deep neural networks. *Geophysics*, 86(1):V1–V13.

Hendrycks, D. and Gimpel, K. (2016). Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Hermanek, P., Rathore, J. S., Aloisi, V., and Carmignato, S. (2018). Principles of x-ray computed tomography. *Industrial X-ray computed tomography*, pages 25–67.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637.

Higgins, I., Matthey, L., Pal, A., Burgess, C. P., Glorot, X., Botvinick, M. M., Mohamed, S., and Lerchner, A. (2017). beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR (Poster)*, 3.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800.

Hinton, G. E. and Zemel, R. (1993). Autoencoders, minimum description length and helmholtz free energy. *Advances in neural information processing systems*, 6.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.

Hochreiter, S., Bengio, Y., Frasconi, P., and Schmidhuber, J. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257.

Horst, R., Pardalos, P. M., and Van Thoai, N. (2000). *Introduction to global optimization*. Springer Science & Business Media.

Hu, W., Jin, Y., Wu, X., and Chen, J. (2021). Progressive transfer learning for low-frequency data prediction in full-waveform inversion. *Geophysics*, 86(4):R369–R382.

Ikotun, A. M., Ezugwu, A. E., Abualigah, L., Abuhaija, B., and Heming, J. (2023). K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*, 622:178–210.

Innes, M. (2018). Flux: Elegant machine learning with julia. *Journal of Open Source Software*, 3(25):602.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 448–456.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1125–1134.

Ivakhnenko, A. G. (1971). Polynomial theory of complex systems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-1(4):364–378.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning*. Springer.

Jensen, J. L. W. V. (1906). On convex functions and inequalities between mean values. *Acta mathematica*, 30(1):175–193.

Jiang, L., Dai, B., Wu, W., and Loy, C. C. (2021). Focal frequency loss for image reconstruction and synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13919–13929.

Johnson, C. R. and Horn, R. A. (1985). *Matrix analysis*. Cambridge university press Cambridge.

Johnson, J., Alahi, A., and Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer.

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer, New York, NY, 2nd edition.

Kannala, J., Mallet, V., and Finnie-Ansley, W. (2018). Attention augmented agent: A reinforcement learning agent with action-adaptive attention for atari games. In *Neural Information Processing Systems (NeurIPS)*.

Karras, T., Aittala, M., Lehtinen, J., Hellsten, J., Aila, T., and Laine, S. (2024). Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24174–24184.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. (2020). Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119.

Kazeminia, S., Baur, C., Kuijper, A., van Ginneken, B., Navab, N., Albarqouni, S., and Mukhopadhyay, A. (2020). Gans for medical image analysis. *Artificial intelligence in medicine*, 109:101938.

Kebaili, A., Lapuyade-Lahorgue, J., and Ruan, S. (2023). Deep learning approaches for data augmentation in medical imaging: a review. *Journal of Imaging*, 9(4):81.

Kessy, A., Lewin, A., and Strimmer, K. (2018). Optimal whitening and decorrelation. *The American Statistician*, 72(4):309–314.

Kim, J., Lee, J. K., and Lee, K. M. (2016). Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654.

Kingma, D., Salimans, T., Poole, B., and Ho, J. (2021). Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P. and Dhariwal, P. (2018). Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kitaev, N., Kaiser, L., and Levskaya, A. (2020). Reformer: The efficient transformer. In *International Conference on Learning Representations (ICLR)*.

Kreis, K., Gao, R., and Vahdat, A. (2023). Latent diffusion models: Is the generative ai revolution happening in latent space? NeurIPS 2023 Tutorial.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Kukreja, N., Louboutin, M., Vieira, F., Luporini, F., Lange, M., and Gorman, G. (2016). Devito: Automated fast finite difference computation. In *2016 Sixth International Workshop on Domain-Specific Languages and High-Level Frameworks for High Performance Computing (WOLFHPC)*, pages 11–19. IEEE.

Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., and Aila, T. (2019). Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32.

Larsen, A. B. L., Sønderby, S. K., Larochelle, H., and Winther, O. (2016). Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning*, pages 1558–1566. PMLR.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and

Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F., et al. (2006). A tutorial on energy-based learning. *Predicting structured data*, 1(0).

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690.

Lee, H., Kang, S., and Chung, K. (2022). Robust data augmentation generative adversarial network for object detection. *Sensors*, 23(1):157.

Lenail, A. (2024). Nn-svg: Alexnet. Accessed: 2024-06-10.

Levinson, N. (1946). The wiener (root mean square) error criterion in filter design and prediction. *Journal of Mathematics and Physics*, 25(1-4):261–278.

Lewis, W. and Vigh, D. (2017). Deep learning prior models from seismic images for full-waveform inversion. In *SEG International Exposition and Annual Meeting*, pages SEG–2017. SEG.

Li, D., Gao, F., and Williamson, P. (2019). A deep learning approach for acoustic fwi with elastic data. In *SEG International Exposition and Annual Meeting*, page D033S039R006. SEG.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. (2018). Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31.

Li, K., Wang, X., Xu, Y., and Wang, J. (2015). Density enhancement-based long-range pedestrian detection using 3-d range data. *IEEE Transactions on Intelligent Transportation Systems*, 17(5):1368–1380.

Li, Y., Alkhalifah, T., and Zhang, Z. (2021). Deep-learning assisted regularized elastic full waveform inversion using the velocity distribution information from wells. *Geophysical Journal International*, 226(2):1322–1335.

Li, Z., Xia, B., Zhang, J., Wang, C., and Li, B. (2022). A comprehensive survey on data-efficient gans in image generation. *arXiv preprint arXiv:2204.08329*.

Liang, Z.-P. and Lauterbur, P. C. (2000). *Principles of magnetic resonance imaging*. SPIE Optical Engineering Press Bellingham.

Lin, T.-Y., Dollar, P., Girshick, R. B., He, K., Hariharan, B., and Belongie, S. (2017a). Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017b). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. *European conference on computer vision*, pages 740–755.

Liu, B. and Sacchi, M. D. (2004). Minimum weighted norm interpolation of seismic records. *Geophysics*, 69(6):1560–1568.

Liu, L., Muelly, M., Deng, J., Pfister, T., and Li, L.-J. (2019). Generative modeling for small-data object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6073–6081.

Liu, Y., Zhang, K., Li, Y., Yan, Z., Gao, C., Chen, R., Yuan, Z., Huang, Y., Sun, H., Gao, J., et al. (2024). Sora: A review on background, technology, limitations, and opportunities of large vision models. *arXiv preprint arXiv:2402.17177*.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Liutkus, A., Simsekli, U., Majewski, S., Durmus, A., and Stöter, F.-R. (2019). Sliced-wasserstein flows: Nonparametric generative modeling via optimal transport and diffusions. In *International Conference on Machine Learning*, pages 4104–4113. PMLR.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440.

Lucas, J., Tucker, G., Grosse, R., and Norouzi, M. (2019). Understanding posterior collapse in generative latent variable models.

Luo, L., Xiong, Y., Liu, Y., and Sun, X. (2018). Adaptive gradient methods with dynamic bound of learning rate. *arXiv preprint arXiv:1902.09843*.

Maas, A. L., Hannun, A. Y., Ng, A. Y., et al. (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Atlanta, Georgia, USA.

MacKay, D. J. (2003). *Information Theory, Inference and Learning Algorithms*. Cambridge University Press.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, CA. University of California Press.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. (2016). Adversarial autoencoders. In *International Conference on Learning Representations (ICLR)*.

Mantiuk, R. K., Tomaszewska, A., and Mantiuk, R. (2012). Comparison of four subjective methods for image quality assessment. In *Computer graphics forum*, volume 31, pages 2478–2491. Wiley Online Library.

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., and Smolley, S. P. (2017). Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802.

Marty, P., Boehm, C., and Fichtner, A. (2021). Acoustoelastic full-waveform inversion for transcranial ultrasound computed tomography. In *Medical Imaging 2021: Ultrasonic Imaging and Tomography*, volume 11602, pages 210–229. SPIE.

Marty, P., Boehm, C., and Fichtner, A. (2022). Elastic full-waveform inversion for transcranial ultrasound computed tomography using optimal transport. In *2022 IEEE International Ultrasonics Symposium (IUS)*, pages 1–4. IEEE.

McAllester, D. (2023). On the mathematics of diffusion models. *arXiv preprint arXiv:2301.11108*.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

McLachlan, G. J. (1999). Mahalanobis distance. *Resonance*, 4(6):20–26.

Mei, X., Liu, Z., Robson, P. M., Marinelli, B., Huang, M., Doshi, A., Jacobi, A., Cao, C., Link, K. E., Yang, T., Wang, Y., Greenspan, H., Deyer, T., Fayad, Z. A., and Yang, Y. (0). Radimagenet: An open radiologic deep learning research dataset for effective transfer learning. *Radiology: Artificial Intelligence*, 0(ja):e210315.

Meng, C., Rombach, R., Gao, R., Kingma, D., Ermon, S., Ho, J., and Salimans, T. (2023). On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14297–14306.

Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., et al. (2017). Mixed precision training. *arXiv preprint arXiv:1710.03740*.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Misra, D. (2019). Mish: A self regularized non-monotonic neural activation function. *arXiv preprint arXiv:1908.08681*, 4(2):10–48550.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D.,

and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Moseley, B., Nissen-Meyer, T., and Markham, A. (2020). Deep learning for fast simulation of seismic waves in complex media. *Solid Earth*, 11(4):1527–1549.

Mosser, L., Dubrule, O., and Blunt, M. J. (2020). Stochastic seismic waveform inversion using generative adversarial networks as a geological prior. *Mathematical Geosciences*, 52(1):53–79.

Mosser, L., Kimman, W., Dramsch, J., Purves, S., De la Fuente Briceño, A., and Ganssle, G. (2018). Rapid seismic domain transfer: Seismic velocity inversion and modeling using deep generative neural networks. In *80th eage conference and exhibition 2018*, volume 2018, pages 1–5. European Association of Geoscientists & Engineers.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT press.

Musicus, B. R. (1988). Levinson and fast choleski algorithms for toeplitz and almost toeplitz matrices.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814.

Neter, J., Kutner, M. H., Nachtsheim, C. J., Wasserman, W., et al. (1996). Applied linear statistical models.

Nguyen, D. (2023). An in depth introduction to variational bayes note. *Available at SSRN*.

Nichol, A. and Dhariwal, P. (2021). Improved denoising diffusion probabilistic models. *arXiv preprint arXiv:2102.09672*.

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. (2021). Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*.

Nichyporuk, B., Cardinell, J., Szeto, J., Mehta, R., Falet, J.-P. R., Arnold, D. L., Tsaftaris, S. A., and Arbel, T. (2022). Rethinking generalization: the impact of annotation style on medical image segmentation. *arXiv preprint arXiv:2210.17398*.

Nilsson, J. and Akenine-Möller, T. (2020). Understanding ssim. *arXiv preprint arXiv:2006.13846*.

Nocedal, J. and Wright, S. (2006). *Numerical Optimization*. Springer Science & Business Media.

Obukhov, A., Seitzer, M., Wu, P.-W., Zhydenko, S., Kyl, J., and Lin, E. Y.-J. (2020). High-fidelity performance metrics for generative models in pytorch. Version: 0.3.0, DOI: 10.5281/zenodo.4957738.

Oksendal, B. (2013). *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media.

Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N. Y., Kainz, B., et al. (2018). Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*.

Olah, C. (2014). Neural networks, manifolds, and topology. Accessed: 2024-08-06.

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Orozco, R., Louboutin, M., Siahkoohi, A., Rizzuti, G., van Leeuwen, T., and Herrmann, F. (2023). Amortized normalizing flows for transcranial ultrasound with uncertainty quantification. *arXiv preprint arXiv:2303.03478*.

Osborne, M. J. and Rubinstein, A. (1994). *A course in game theory*. MIT press.

Osogami, T. (2017). Boltzmann machines and energy-based models. *arXiv preprint arXiv:1708.06008*.

Pambrun, J.-F. and Noumeir, R. (2015). Limitations of the ssim quality metric in the context of diagnostic imaging. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 2960–2963. IEEE.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64.

Pappas, T. N., Safranek, R. J., and Chen, J. (2000). Perceptual criteria for image quality evaluation. *Handbook of image and video processing*, 110.

Parisotto, E. and Salakhutdinov, R. (2018). Neural map: Structured memory for deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library.

Peyré, G., Cuturi, M., et al. (2019). Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607.

Pezoa, F., Reutter, J. L., Suarez, F., Ugarte, M., and Vrgoč, D. (2016). Foundations of json schema. In *Proceedings of the 25th International Conference on World Wide Web*, pages 263–273. International World Wide Web Conferences Steering Committee.

Pinaya, W. H., Graham, M. S., Kerfoot, E., Tudosiu, P.-D., Dafflon, J., Fernandez, V., Sanchez, P., Wolleb, J., da Costa, P. F., Patel, A., et al. (2023). Generative ai for medical imaging: extending the monai framework. *arXiv preprint arXiv:2307.15208*.

Pinaya, W. H., Tudosiu, P.-D., Dafflon, J., Da Costa, P. F., Fernandez, V., Nachev, P., Ourselin, S., and Cardoso, M. J. (2022). Brain imaging generation with latent

diffusion models. In *MICCAI Workshop on Deep Generative Models*, pages 117–126. Springer.

Plessix, R.-E. (2006). A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophysical Journal International*, 167(2):495–503.

Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., and Rombach, R. (2023). Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*.

Pratt, R. G., Huang, L., Duric, N., and Littrup, P. (2007). Sound-speed and attenuation imaging of breast tissue using waveform tomography of transmission ultrasound data. In *Medical Imaging 2007: Physics of Medical Imaging*, volume 6510, pages 1523–1534. SPIE.

Pratt, R. G., Shin, C., and Hick, G. (1998). Gauss–newton and full newton methods in frequency–space seismic waveform inversion. *Geophysical journal international*, 133(2):341–362.

Prechelt, L. (1998). Automatic early stopping using cross validation: quantifying the criteria. *Neural networks*, 11(4):761–767.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training.

Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., and Courville, A. (2019). On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR.

Raknes, E. B. and Arntsen, B. (2017). Challenges and solutions for performing 3d time-domain elastic full-waveform inversion. *The leading edge*, 36(1):88–93.

Ramachandran, P., Zoph, B., and Le, Q. V. (2017). Searching for activation functions. *arXiv preprint arXiv:1710.05941*.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021). Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr.

Rashid, H., Tanveer, M. A., and Khan, H. A. (2019). Skin lesion classification using gan based data augmentation. In *2019 41St annual international conference of the IEEE engineering in medicine and biology society (EMBC)*, pages 916–919. IEEE.

Rasht-Behesht, M., Huber, C., Shukla, K., and Karniadakis, G. E. (2022). Physics-informed neural networks (pinns) for wave propagation and full waveform inversions. *Journal of Geophysical Research: Solid Earth*, 127(5):e2021JB023120.

Razavi, A., Oord, A. v. d., Poole, B., and Vinyals, O. (2019a). Preventing posterior collapse with delta-vaes. *arXiv preprint arXiv:1901.03416*.

Razavi, A., Van den Oord, A., and Vinyals, O. (2019b). Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32.

Reddi, S. J., Kale, S., and Kumar, S. (2019). On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.

Ren, J., Li, J., and Liu, Y. (2024). Physics-embedded neural networks for enhanced ultrasonic brain tomography. In *Journal of Physics: Conference Series*, volume 2822, page 012016. IOP Publishing.

Ren, S., He, K., Girshick, R. B., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 91–99.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR.

Richardson, A. (2018). Seismic full-waveform inversion using deep learning tools and techniques. *arXiv preprint arXiv:1801.07232*.

Richardson, A. and Feller, C. (2019). Seismic data denoising and deblending using deep learning. *arXiv preprint arXiv:1907.01497*.

Robins, T. (2021). *Towards Ultrasound Full-Waveform Inversion in Medical Imaging*. PhD thesis, Imperial College London Department of Bioengineering.

Robins, T., Camacho, J., Agudo, O. C., Herraiz, J. L., and Guasch, L. (2021). Deep-learning-driven full-waveform inversion for ultrasound breast imaging. *Sensors*, 21(13):4570.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

Rossky, P. J., Doll, J. D., and Friedman, H. L. (1978). Brownian dynamics as smart monte carlo simulation. *The Journal of Chemical Physics*, 69(10):4628–4633.

Rousseau, F., Initiative, A. D. N., et al. (2010). A non-local approach for image super-resolution using intermodality priors. *Medical image analysis*, 14(4):594–605.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Saad, O. M., Harsuko, R., and Alkhalifah, T. (2024). Siamesefwi: A deep learning network for enhanced full waveform inversion. *Journal of Geophysical Research: Machine Learning and Computation*, 1(3):e2024JH000227.

Sabour, A., Fidler, S., and Kreis, K. (2024). Align your steps: Optimizing sampling schedules in diffusion models. *arXiv preprint arXiv:2404.14507*.

Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. J., and Norouzi, M. (2022). Image super-resolution via iterative refinement. *IEEE transactions on pattern analysis and machine intelligence*, 45(4):4713–4726.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., and Brock, A. (2016). Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242.

Salmon, E., Ir, C. B., and Hustinx, R. (2015). Pitfalls and limitations of pet/ct in brain imaging. 45(6):541–551.

Samala, R. K., Chan, H.-P., Hadjiiski, L. M., Helvie, M. A., and Richter, C. D. (2020). Generalization error analysis for deep convolutional neural network with transfer learning in breast cancer diagnosis. *Physics in Medicine & Biology*, 65(10):105002.

Sandhu, G., Li, C., Roy, O., Schmidt, S., and Duric, N. (2015). Frequency domain ultrasound waveform tomography: breast imaging using a ring transducer. *Physics in Medicine & Biology*, 60(14):5381.

Santurkar, S., Tsipras, D., Ilyas, A., and Madry, A. (2018). How does batch normalization help optimization? In *Advances in Neural Information Processing Systems*, volume 31.

Sauer, A., Lorenz, D., Blattmann, A., and Rombach, R. (2025). Adversarial diffusion distillation. In *European Conference on Computer Vision*, pages 87–103. Springer.

Sayed, A. H. (2011). *Adaptive filters*. John Wiley & Sons.

Schreiner, M. (2023). Gpt-4 architecture, datasets, costs, and more leaked. Accessed: 2024-08-06.

Scotti, P., Banerjee, A., Goode, J., Shabalin, S., Nguyen, A., Dempster, A., Verlinde, N., Yundler, E., Weisberg, D., Norman, K., et al. (2024). Reconstructing the mind's eye: fmri-to-image with contrastive learning and diffusion priors. *Advances in Neural Information Processing Systems*, 36.

Seo, J. K. and Woo, E. J. (2012). *Nonlinear inverse problems in imaging*. John Wiley & Sons.

Shah, N., Warner, M., Nangoo, T., Umpleby, A., Stekl, I., Morgan, J., and Guasch, L. (2012). Quality assured full-waveform inversion: Ensuring starting model adequacy. In *SEG International Exposition and Annual Meeting*, pages SEG–2012. SEG.

Shazeer, N. (2020). Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*.

Shen, T.-C., Yang, C.-P., Yen, I. E.-H., and Lin, S.-D. (2022). Towards l1 regularization for deep neural networks: Model sparsity versus task difficulty. In *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–9. IEEE.

Shin, C., Yoon, K., Marfurt, K. J., Park, K., Yang, D., Lim, H. Y., Chung, S., and Shin, S. (2001). Efficient calculation of a partial-derivative wavefield using reciprocity for seismic imaging and inversion. *Geophysics*, 66(6):1856–1863.

Shin, H.-C., Zhang, Y., Chefd'hotel, C., Conjeti, S., Cui, Z., Kim, L., Sollee, I., Pan, L., Funkhouser, T., and Mansi, T. (2018). Medical image synthesis for data augmentation and anonymization using generative adversarial networks. *International Workshop on Simulation and Synthesis in Medical Imaging*, pages 1–11.

Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48.

Siahkoohi, A., Orozco, R., Rizzuti, G., and Herrmann, F. J. (2022). Wave-equation-based inversion with amortized variational bayesian inference. *arXiv preprint arXiv:2203.15881*.

Siahkoohi, A., Rizzuti, G., and Herrmann, F. (2020). A deep-learning based bayesian approach to seismic imaging and uncertainty quantification. In *EAGE 2020 Annual Conference & Exhibition Online*, volume 2020, pages 1–5. European Association of Geoscientists & Engineers.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sirgue, L. (2003). *Inversion de la forme d'onde dans le domaine fréquentiel de données sismiques grands offsets*. PhD thesis, Ecole Normale Supérieure Paris.

Sirgue, L. (2006). The importance of low frequency and large offset in waveform inversion. In *68th EAGE conference and exhibition incorporating SPE EUROPEC 2006*, pages cp–2. European Association of Geoscientists & Engineers.

Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2256–2265. PMLR.

Song, J., Meng, C., and Ermon, S. (2020). Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Song, Y. (2021). Generative modeling by estimating gradients of the data distribution. Accessed: 2024-12-21.

Song, Y., Dhariwal, P., Chen, M., and Sutskever, I. (2023). Consistency models. *arXiv preprint arXiv:2303.01469*.

Song, Y. and Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*.

Song, Y. and Kingma, D. P. (2021). How to train your energy-based models. *arXiv preprint arXiv:2101.03288*.

Song, Y., Shen, L., Xing, L., and Ermon, S. (2021a). Solving inverse problems in medical imaging with score-based generative models. *arXiv preprint arXiv:2111.08005*.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021b). Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Staudemeyer, R. C. and Morris, E. R. (2019). Understanding lstm–a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*.

Strong, G. (2023). *Machine learning for ultrasound brain imaging*. PhD thesis, Imperial College London Department of Earth Science and Engineering.

Sun, H. and Demanet, L. (2020). Extrapolated full-waveform inversion with deep learning. *Geophysics*, 85(3):R275–R288.

Sun, H. and Demanet, L. (2021). Deep learning for low-frequency extrapolation of multicomponent data in elastic fwi. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–11.

Sun, J., Niu, Z., Innanen, K. A., Li, J., and Trad, D. O. (2020). A theory-guided deep-learning formulation and optimization of seismic waveform inversion. *Geophysics*, 85(2):R87–R99.

Sundaram, S. and Hulkund, N. (2021). Gan-based data augmentation for chest x-ray classification. *arXiv preprint arXiv:2107.02970*.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 1057–1063.

Suzuki, K. (2017). Overview of deep learning in medical imaging. *Radiological Physics and Technology*, 10(3):257–273.

Szabo, T. L. (2013). *Diagnostic ultrasound imaging: inside out*. Academic press.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Tarantola, A. (1984). Inversion of seismic reflection data in the acoustic approximation. *Geophysics*, 49(8):1259–1266.

Theis, L., Oord, A. v. d., and Bethge, M. (2015). A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*.

Thorpe, M., Park, S., Kolouri, S., Rohde, G. K., and Slepčev, D. (2017). A transportation l^p distance for signal analysis. *Journal of mathematical imaging and vision*, 59(2):187–210.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

Tieleman, T. (2013). affnist. https://www.cs.toronto.edu/~tijmen/affNIST/.

Tieleman, T. and Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning.

Vakanski, A. and Xian, M. (2021). Evaluation of complexity measures for deep learning generalization in medical image analysis. In *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE.

Van Den Oord, A., Kalchbrenner, N., and Kavukcuoglu, K. (2016). Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR.

Van Essen, D. C., Ugurbil, K., Auerbach, E., Barch, D., Behrens, T. E., Bucholz, R., Chang, A., Chen, L., Corbetta, M., Curtiss, S. W., et al. (2012). The human connectome project: a data acquisition perspective. *Neuroimage*, 62(4):2222–2231.

Vantassel, J. P., Kumar, K., and Cox, B. R. (2022). Using convolutional neural networks to develop starting models for near-surface 2-d full waveform inversion. *Geophysical Journal International*, 231(1):72–90.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Vig, J. (2019). A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1906.05714*.

Vincent, P. (2011). A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674.

Virieux, J. and Operto, S. (2009). An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 74(6):WCC1–WCC26.

von Ramm, O., Smith, S., and Kisslo, J. (1978). Ultrasound tomography of the adult brain. In *Ultrasound in Medicine: Volume 4*, pages 261–267. Springer.

Wang, B., Zhang, N., Lu, W., and Wang, J. (2019). Deep-learning-based seismic data interpolation: A preliminary result. *Geophysics*, 84(1):V11–V20.

Wang, F., Huang, X., and Alkhalifah, T. A. (2023). A prior regularized full waveform inversion using generative diffusion models. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–11.

Wang, H., Chen, Y., Kang, J., Wu, Y., Kim, Y.-J., and Lin, Y. (2024). Wavediffusion:

Exploring full waveform inversion via joint diffusion in the latent space. *arXiv preprint arXiv:2410.09002*.

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., and Catanzaro, B. (2018). High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807.

Wang, Y., Pan, Y., and Li, H. (2020). What is brain health and why is it important? *bmj*, 371.

Wang, Z. and Bovik, A. C. (2006). Modern image quality assessment. *Synthesis Lectures on Image, Video, and Multimedia Processing*, 2(1):1–156.

Wang, Z. and Bovik, A. C. (2009). Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE signal processing magazine*, 26(1):98–117.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612.

Wangz, B. A., Sheikh, H., et al. (2004). Image quality assessment: From error visibility to structural similarity. *IEEETransactionsonImageProcessing*, 13(4):600G612.

Warner, M. and Guasch, L. (2016). Adaptive waveform inversion: Theory. *Geophysics*, 81(6):R429–R445.

Warner, M., Ratcliffe, A., Nangoo, T., Morgan, J., Umpleby, A., Shah, N., Vinje, V., Štekl, I., Guasch, L., Win, C., et al. (2013). Anisotropic 3d full-waveform inversion. *Geophysics*, 78(2):R59–R80.

Warner, M., Stekl, I., Umpleby, A., Morgan, J., Pain, C., and Wang, Y. (2008). 3d wavefield tomography: Problems, opportunities and future directions. In *70th EAGE Conference and Exhibition-Workshops and Fieldtrips*, pages cp–41. European Association of Geoscientists & Engineers.

Watson, N. J. (2022). Ultrasound tomography. In *Industrial tomography*, pages 245–273. Elsevier.

Wei, J. and Zou, K. (2019). Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 6383–6389.

Weng, L. (2017). From gan to wgan. https://lilianweng.github.io/posts/2017-08-20-gan/.

Weng, L. (2021). What are diffusion models? *lilianweng.github.io*.

Wiener, N. (1942). The interpolation, extrapolation and smoothing of stationary time series. ndrc report.

Williamson, P. (1991). A guide to the limits of resolution imposed by scattering in ray tomography. *Geophysics*, 56(2):202–207.

Wiskin, J., Borup, D., Johnson, q. S., Berggren, M., Abbott, T., and Hanover, R. (2007). Full-wave, non-linear, inverse scattering: high resolution quantitative breast tissue tomography. In *Acoustical Imaging*, pages 183–193. Springer.

Wood, J. L. (1971). Dynamic response of human cranial bone. *Journal of biomechanics*, 4(1):1–12.

Wu, H. R. and Rao, K. R. (2017). *Digital video image quality and perceptual coding*. CRC press.

Wu, Y. and He, K. (2018). Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19. Springer.

Wu, Y. and Lin, Y. (2019). Inversionnet: An efficient and accurate data-driven full waveform inversion. *IEEE Transactions on Computational Imaging*, 6:419–433.

Wu, Y. and McMechan, G. A. (2019). Parametric convolutional neural network-domain full-waveform inversion. *Geophysics*, 84(6):R881–R896.

Xiao, Z., Kreis, K., Kautz, J., and Vahdat, A. (2020). Vaebm: A symbiosis between variational autoencoders and energy-based models. *arXiv preprint arXiv:2010.00654*.

Xu, Z.-Q. J., Zhang, Y., Luo, T., Xiao, Y., and Ma, Z. (2019). Frequency principle: Fourier analysis sheds light on deep neural networks. *arXiv preprint arXiv:1901.06523*.

Yan, F. et al. (2010). Sparse gaussian process regression via l1 penalization. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 1183–1190. Citeseer.

Yang, F. and Ma, J. (2019). Deep-learning inversion: A next-generation seismic velocity model building method. *Geophysics*, 84(4):R583–R599.

Yang, F. and Ma, J. (2023). Fwigan: Full-waveform inversion via a physics-informed generative adversarial network. *Journal of Geophysical Research: Solid Earth*, 128(4):e2022JB025493.

Yang, Y., Zhang, X., Guan, Q., and Lin, Y. (2022). Making invisible visible: Data-driven seismic inversion with spatio-temporally constrained data augmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–16.

Yao, G., Da Silva, N. V., and Wu, D. (2018). An effective absorbing layer for the boundary condition in acoustic seismic wave simulation. *Journal of Geophysics and Engineering*, 15(2):495–511.

Yao, J., Guasch, L., and Warner, M. (2023a). Neural networks as a tool for domain translation of geophysical data. *Geophysics*, 88(3):V267–V275.

Yao, J., Guasch, L., Warner, M., Lin, T., and Percak-Dennett, E. (2020). Geophysical data and gradient translation using deep neural networks. In *SEG International Exposition and Annual Meeting*, page D041S101R003. SEG.

Yao, J. and Wang, Y. (2022). Building a full-waveform inversion starting model from wells with dynamic time warping and convolutional neural networks. *Geophysics*, 87(2):R223–R230.

Yao, J., Warner, M., and Wang, Y. (2023b). Regularization of anisotropic full-waveform inversion with multiple parameters by adversarial neural networks. *Geophysics*, 88(1):R95–R103.

Yin, Z., Orozco, R., and Herrmann, F. J. (2024a). Wiser: multimodal variational inference for full-waveform inversion without dimensionality reduction. *arXiv preprint arXiv:2405.10327*.

Yin, Z., Orozco, R., Louboutin, M., and Herrmann, F. J. (2024b). Wise: Full-waveform variational inference via subsurface extensions. *Geophysics*, 89(4):A23–A28.

Ylitalo, J., Koivukangas, J., and Oksman, J. (1990). Ultrasonic reflection mode computed tomography through a skullbone. *IEEE Transactions on Biomedical Engineering*, 37(11):1059–1066.

You, Y., Li, J., Reddi, S., Hseu, J., Kumar, S., Bhojanapalli, S., Song, X., Demmel, J., and Hsieh, C.-J. (2020). Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*.

Yu, J., Li, X., Koh, J. Y., Zhang, H., Pang, R., Qin, J., Ku, A., Xu, Y., Baldridge, J., and Wu, Y. (2021). Vector-quantized image modeling with improved vqgan. *CoRR*, abs/2110.04627.

Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., et al. (2020). Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018a). The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018b). The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*.

Zhang, W., Gao, J., Gao, Z., and Chen, H. (2020). Adjoint-driven deep-learning seismic full-waveform inversion. *IEEE Transactions on Geoscience and Remote Sensing*, 59(10):8913–8932.

Zhao, H., Gallo, O., Frosio, I., and Kautz, J. (2016). Loss functions for image restoration with neural networks. *IEEE Transactions on computational imaging*, 3(1):47–57.

Zhao, J., Xu, H., Liu, H., Wu, J., Zheng, Y., and Wu, D. (2019). Detection and tracking of pedestrians and vehicles using roadside lidar sensors. *Transportation research part C: emerging technologies*, 100:68–87.

Zhao, S., Song, J., and Ermon, S. (2017). Towards deeper understanding of variational autoencoding models. *arXiv preprint arXiv:1702.08658*.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232.

Zhu, W., Xu, K., Darve, E., Biondi, B., and Beroza, G. C. (2022). Integrating deep neural networks with full-waveform inversion: Reparameterization, regularization, and uncertainty quantification. *Geophysics*, 87(1):R93–R109.

Zhu, X., Kwark, D. H., Zhu, R., Hong, K., Tao, Y., Luo, S., Li, Y., Liang, Z.-P., and Kindratenko, V. (2025). Introducing 3d representation for medical image volume-to-volume translation via score fusion.

Appendix A

Structural Elements of the `InverseLDM` Package

A.1 Example Configuration file `config.yml`

Example of a configuration file in YAML format for the `InverseLDM` package, showing the high-level of customisation available to the user:

```
data:
    dataset: Brain2D
    data_path:
    ↪ /scratch_hive/dp4018/data/ultrasound-data/Ultrasound-Vp-axial-models/
    mode: vp
    maxsamples: null
    slowness: false
    resize:
    - 256
    - 256
    clip_outliers: false
    log: true
    to_tensor: false
    normalise: false
```

```
    condition:
        mode: null
        resize: null
        antialias: false
        scale: null
        clip_outliers: false
        to_tensor: false
        normalise: false
    dataset_path: invldm/datasets
    antialias: true
    scale: null
    sampling:
        in_size:
        - 64
        - 64
        in_channels: 3
    sampling_only: false
logging:
    tool: JSON
    plot:
        vmin: 0.3
        vmax: 1.0
        cmap: oscar
    track_folder: exps/thesis/axial/base/JSON
    logger: !!python/object/apply:logging.getLogger []

autoencoder:
    model:
        spatial_dims: 2
        in_channels: 1
        out_channels: 1
        num_res_blocks: 2
        num_channels:
        - 64
        - 128
```

```
      - 128
      attention_levels:
      - false
      - false
      - true
      latent_channels: 3
      norm_num_groups: 32
      norm_eps: 1.0e-06
      with_encoder_nonlocal_attn: false
      with_decoder_nonlocal_attn: false
      use_flash_attention: false
      use_checkpointing: false
      use_convtranspose: false
      checkpoint: null
  params:
      recon_loss: l1
      recon_weight: 1.0
      kl_weight: 1.0e-06
      perceptual_loss: lpips
      perceptual_weight: 1.0
      lpips_model: vgg
      wiener_mode: reverse
      wiener_filter_dim: 2
      wiener_filter_scale: 2
      wiener_epsilon: 2.0
      wiener_laplace_alpha: -0.2
      wiener_laplace_beta: 1.5
      wiener_weight: 0.0
      adversarial_mode: least_squares
      disc_n_layers: 2
      disc_feature_channels: 16
      adversarial_weight: 0.1
  training:
      batch_size: 16
      n_epochs: 75
```

```
        warm_up_epochs: 5
        save_recon_freq: 2000
        ckpt_freq: 500
        ckpt_last_only: true
        sampling_freq: 0
    validation:
        split: 0.0
        batch_size: 0
        freq: 0
        save_recon_freq: 0
    optim:
        weight_decay: 0.0
        optimiser: Adam
        lr: 0.0001
        d_lr: 0.0005
        beta1: 0.9
        amsgrad: false
        eps: 1.0e-08
        grad_clip: null
        lr_scheduler:
            scheduler: null
    name: autoencoder
    log_path: exps/thesis/axial/base/logs/autoencoder
    ckpt_path: exps/thesis/axial/base/logs/autoencoder/checkpoints
    recon_path: exps/thesis/axial/base/logs/autoencoder/recons
    sampling_only: false

diffusion:
    model:
        num_res_blocks: 2
        num_channels:
        - 64
        - 128
        - 256
        - 512
```

```
        attention_levels:
        - false
        - true
        - true
        - true
        norm_num_groups: 32
        norm_eps: 1.0e-06
        resblock_updown: false
        num_head_channels: 8
        transformer_num_layers: 3
        upcast_attention: false
        use_flash_attention: false
        checkpoint: null
    params:
        recon_loss: l2
        beta_start: 0.0015
        beta_end: 0.0205
        num_train_timesteps: 1000
        num_inference_timesteps: 50
        schedule: linear_beta
        sampler: ddim
    training:
        batch_size: 16
        n_epochs: 400
        ckpt_freq: 100
        ckpt_last_only: true
        sampling_freq: 2000
    validation:
        split: 0.0
        batch_size: 0
        freq: 0
        save_recon_freq: 0
    optim:
        weight_decay: 0.0001
        optimiser: Adam
```

```
        lr: 2.0e-05
        beta1: 0.9
        amsgrad: false
        eps: 1.0e-08
        grad_clip: null
        lr_scheduler:
            scheduler: null
    name: diffusion
    log_path: exps/thesis/axial/base/logs/diffusion
    ckpt_path: exps/thesis/axial/base/logs/diffusion/checkpoints
    samples_path: exps/thesis/axial/base/logs/diffusion/samples
    sampling_only: false

run:
    config: exps/thesis/config_axial.yml
    logdir: exps/thesis/axial
    name: base
    comment: ''
    jobid: 7990
    device: cuda
    gpu_ids:
    - 0
    seed: 42
    resume_training: false
    overwrite: true
    y: true
    verbose: info
    exp_folder: exps/thesis/axial/base
    log_folder: exps/thesis/axial/base/logs
    pid: 835743
```

A.2 Customisable `Dataset` classes

The following code snippet shows the implementation of the `BaseDataset` and `Brain2DDataset` classes in the `InverseLDM` package. The `BaseDataset` class is an abstract class that defines the structure of the dataset classes in the package. The `Brain2DDataset` class is a concrete implementation of the `BaseDataset` class, which is used to load 2D brain ultrasound and MRI images from given data locations. From `BaseDataset`, the user can define their own custom dataset classes by inheriting from it and implementing the abstract methods.

```
class BaseDataset(Dataset):
    def __init__(self, args):
        self.args = args
        self.transform = self._get_transform(**namespace2dict(self.args))
        self.cond_transform =
        ↪  self._get_transform(**namespace2dict(self.args.condition))

    @abstractmethod
    def __len__(self):
        raise NotImplementedError

    @abstractmethod
    def __getitem__(self, index):
        raise NotImplementedError

    def _get_transform(self, **kwargs):
        resize = kwargs.pop("resize", None)
        antialias = kwargs.pop("antialias", True)
        to_tensor = kwargs.pop("to_tensor", True)
        outliers = kwargs.pop("clip_outliers", False)
        scale = kwargs.pop("scale", [0, 1])
        normalise = kwargs.pop("normalise", False)
```

```
        transform_list = []
        if resize:
            transform_list.append(Resize(resize, antialias=antialias))
        if to_tensor:
            transform_list.append(ToTensor())
        if outliers:
            transform_list.append(ClipOutliers(fence=outliers))
        if scale:
            scale = scale
            transform_list.append(Scale(scale=scale))
        if normalise:
            transform_list.append(Normalize(normalise[0],
                                            normalise[1]))

        return Compose(transform_list)


class Brain2DDataset(BaseDataset):
def __init__(self, args):
    BaseDataset.__init__(self, args)

    # Save args in object
    self.args = args

    # Checking mode
    try:
        mode = self.args.mode
        if mode.lower() in ["mri", "vp"]:
            self.mode = mode
        else:
            raise ValueError("mode must be 'mri' or 'vp', \
                             but got '{}'".format(mode))
    except KeyError:
        self.mode = "mri"
```

```
    # Check maxsamples
    try:
        maxsamples = self.args.maxsamples
    except KeyError:
        maxsamples = None

    # Get image paths
    self.data_paths = []
    self._get_image_paths(self.args.data_path, maxsamples)
    return None

def _get_image_paths(self, path, maxsamples=None):
    prefix = "m" if self.mode == "mri" else "vp"
    suffix = (".npy", ".npy.gz")

    # Loop through folders and subfolders
    for subdir, _, files in os.walk(path):
        for filename in files:
            if filename.lower().startswith(prefix) and \
               filename.lower().endswith(suffix):
                self.data_paths.append(os.path.join(subdir, filename))
    self.data_paths = self.data_paths[:maxsamples]
    assert len(self.data_paths) > 0, f" Found no data samples to load in
    ↪  {path} with prefix {prefix} and suffixes {suffix}"
    return None

def _get_condition_path(self, data_path):
    """
    Finds inside self.args.condition.path a .npy or .npy.gz file
    containing the name of the data item of data_path with suffix
    ↪  '-self.args.condition.mode'
    """

    name = os.path.splitext(os.path.split(data_path)[-1])[0]
```

```
    cond_path = os.path.join(self.args.condition.path, name +
    ↪  f"-{self.args.condition.mode}")

    if os.path.isfile(cond_path + ".npy"):
        return cond_path + ".npy"
    elif os.path.isfile(cond_path + ".npy.gz"):
        return cond_path + ".npy.gz"
    else:
        raise FileNotFoundError(f"Condition file {cond_path} does not exist
        ↪  in neither .npy or .npy.gz format")


def _read_npy_data(self, data_path):
    # Load .npy or .npy.gz data into torch tensor
    if data_path.endswith(".npy.gz"):
        with gzip.open(data_path, 'rb') as f:
            y = torch.from_numpy(np.load(f))
    elif data_path.endswith(".npy"):
        y = torch.from_numpy(np.load(data_path))
    else:
        raise TypeError

    # Check channel dimension exists
    if len(y.shape) < 3:
        y = y.unsqueeze(0)
    return y

def __getitem__(self, index):
    # Data item path
    y_path = self.data_paths[index]

    # Read data
    y = self._read_npy_data(y_path)

    # slowness
```

```
    if self.args.slowness:
        y = 1./ y

    # Apply transform
    if self.transform:
        y = self.transform(y)

    # Get condition, apply steps above
    if self.args.condition.mode is not None and self.args.condition.path is
    ↪  not None:
        cond_path = self._get_condition_path(y_path)
        cond = self._read_npy_data(cond_path)
        cond = cond / torch.abs(torch.max(cond))
        if self.cond_transform:
            cond = self.cond_transform(cond)
        return y, cond
    return y

def __len__(self):
    return len(self.data_paths)

def __str__(self):
    dic = {}
    dic["name"] = self.__class__.__name__
    dic.update(self.__dict__)
    dic.pop("data_paths")
    dic["len"] = self.__len__()
    return "{}".format(dic)

def info(self, nsamples=None):
    nsamples = self.__len__() if nsamples is None else nsamples
    sample = self.__getitem__(0)[0]
    idx = torch.randint(0, self.__len__(), [nsamples])
    arr = torch.empty_like(sample).unsqueeze(0).repeat(nsamples, 1, 1, 1)
    for i in range(nsamples):
```

```
        arr[i] = self.__getitem__(idx[i])[0]
    stats = {"max": arr.max().item(), "min": arr.min().item(),
             "mean": arr.mean().item(), "std": arr.std().item(),
             "shape": sample.shape}
    return stats
```

A.3 Custom Loggers

The following code snippet shows the implementation of the `BaseLogger`, `JSONLogger` and `TensorboardLogger` classes in the `InverseLDM` package. The `BaseLogger` class is an abstract class that defines the structure of the logger classes in the package. The `JSONLogger` and `TensorboardLogger` classes are concrete implementations of the `BaseLogger` class, which are used to log scalar values, figures and hyperparameters to JSON and Tensorboard files respectively. From `BaseLogger`, the user can define their own custom logger classes by inheriting from it and implementing the abstract methods.

```
class BaseLogger():
    def __init__(self, args):
        self.args = args

    def log_scalar(self, tag, val, step, **kwargs):
        pass

    def log_figure(self, tag, fig, step, **kwargs):
        pass

    def log_hparams(self, hparam_dict, metric_dict, **kwargs):
        pass
```

```
class JSONLogger(BaseLogger):
    def __init__(self, args):
        super().__init__(args)
        self.hparams = {}
        self.scalars = {}

        self.log_file = os.path.join(
            self.args.run.exp_folder,
            "JSON/log.json"
        )
        return None

    def save_logger(self):
        json_dic = {"hyperparameters": self.hparams}
        for key, val in self.scalars.items():
            json_dic.update({key: val})

        # print(json_dic)
        with open(self.log_file, 'w') as f:
            json.dump(json_dic, f)

    def log_scalar(self, tag, val, step, **kwargs):
        try:
            self.scalars[tag].append(val)
            self.scalars[tag+"_step"].append(step)
        except KeyError:
            self.scalars[tag] = [val]
            self.scalars[tag+"_step"] = [step]
        self.save_logger()
        return None

    def log_hparams(self, hparam_dict, metric_dict, **kwargs):
        # hparam_dict.update(metric_dict)
        self.hparams.update(clear_dic_for_json(hparam_dict))
        self.save_logger()
```

```
        return None


class TensorboardLogger(BaseLogger):
    def __init__(self, args):
        super().__init__(args)

        self.logger = tb.SummaryWriter(
            log_dir=os.path.join(
                self.args.run.exp_folder,
                "tensorboard"
            )
        )
        return None

    def log_scalar(self, tag, val, step, **kwargs):
        self.logger.add_scalar(tag, val, step)
        return None

    def log_figure(self, tag, fig, step, **kwargs):
        self.logger.add_figure(tag, fig, step)
        return None
```

Appendix B

Least Mean Square Automatic Gain Control Algorithm

The following pseudocode details the AGC algorithm based on LMS adjustment, designed to maintain a target amplitude level for a given input signal.

Algorithm 11 Automatic Gain Control with Least-Mean-Square (LMS) Adjustment

Input: Signal x, step size $\lambda = 0.2$, reference amplitude ref $= 5$
Output: Gain-controlled signal y
Initialise: AGCMult $\leftarrow 0$
$N \leftarrow$ length of x
Initialise output array y of same length as x
Initialise error array err of same length as N
for $n = 0$ to $N - 1$ **do**
    $y[n] \leftarrow x[n] \times \text{AGCMult}$ ▷ Apply the gain multiplier to the current sample
    $y_{\text{mag}} \leftarrow |y[n]|$ ▷ Compute the magnitude of the current output sample
    $\text{err}[n] \leftarrow \text{ref} - y_{\text{mag}}$ ▷ Calculate the current error from the reference amplitude
    $\text{AGCMult} \leftarrow \text{AGCMult} + \lambda \times \text{err}[n]$ ▷ Update the gain multiplier using the error and step size
end for
Return y ▷ Return the gain-controlled signal

Appendix C

Contrastive Embedding Models

In this appendix, we present the visualisation of the embeddings from contrastive learning models in a two-dimensional (2D) space, using Principal Component Analysis (PCA) and Uniform Manifold Approximation and Projection (UMAP). The visualisation of nine models, selected from a total of 116 experiments conducted for contrastive learning, is depicted in Figures C.1 and C.2. The primary model discussed in the main text is SG-222, and it is illustrated here to show the true Euclidean distances within the embedded space.

It is important to emphasise that the true Euclidean distances between data points are not preserved when projected into a 2D space. In the main text, Euclidean distances were computed within this projected space to enable a comparison between the non-embedded image data and the corresponding embeddings. Since the original data and embeddings have different dimensions, the true Euclidean distances could not be directly calculated for the non-embedded data. Thus, the computation of these distances in the projected space was necessary to facilitate a meaningful comparison. In contrast, this appendix highlights the true Euclidean distances within the embedded space to better illustrate the similarities between the model embeddings. The aim is to provide a clearer understanding of the relationships among the different embeddings in the learned space, revealing how closely or distantly the embeddings are positioned relative to one another.

For the projection of both raw data and images, a normalisation step was applied to scale the values to the range [0, 1] prior to projection. However, no pre-processing

was carried out on the embeddings of the models before their projection.

The hyperparameters used for each visualised model are presented in Table C.1. Both the data and images were scaled to the range [0, 1] before being inserted into the model, and the same image pre-processing pipeline described in Chapter 4 was applied here as well.

The table provides detailed descriptions of each hyperparameter. The `GPU Count` refers to the number of GPUs utilised during the training process. `Runtime` indicates the total amount of time, measured in days, required for the model's training. The `batch_size` is the number of samples processed in each training batch. For image data, the `image_clip_outliers` parameter specifies whether outer or inner quartile outliers were clipped, while `data_clip_outliers` performs the same operation for the data. The `data_resize` parameter indicates the AGC stack data after resizing, which is used as input to the model.

The `vision_layers` parameter denotes the number of layers in the vision transformer encoder, and `vision_patch_size` specifies the size of the patches in the encoder. The `embed_size` defines the spatial size of the embeddings, while `embed_dim` refers to the channel dimension of the embeddings. The `vision_width` parameter defines the width of the vision transformer encoder.

Further, the `learning_rate` indicates the rate at which the optimiser updates the model's parameters. The `optimiser` parameter identifies which optimiser was employed during training, and `adam_betas` refers to the betas used in the Adam optimiser. The `warm_restart_steps` parameter defines the number of warm-up restart steps for the learning rate scheduler, while the `epochs` parameter specifies the total number of epochs the model was trained for. Finally, `training_loss` represents the value of the contrastive loss, as described in Equation 5.7, at the conclusion of the training process.

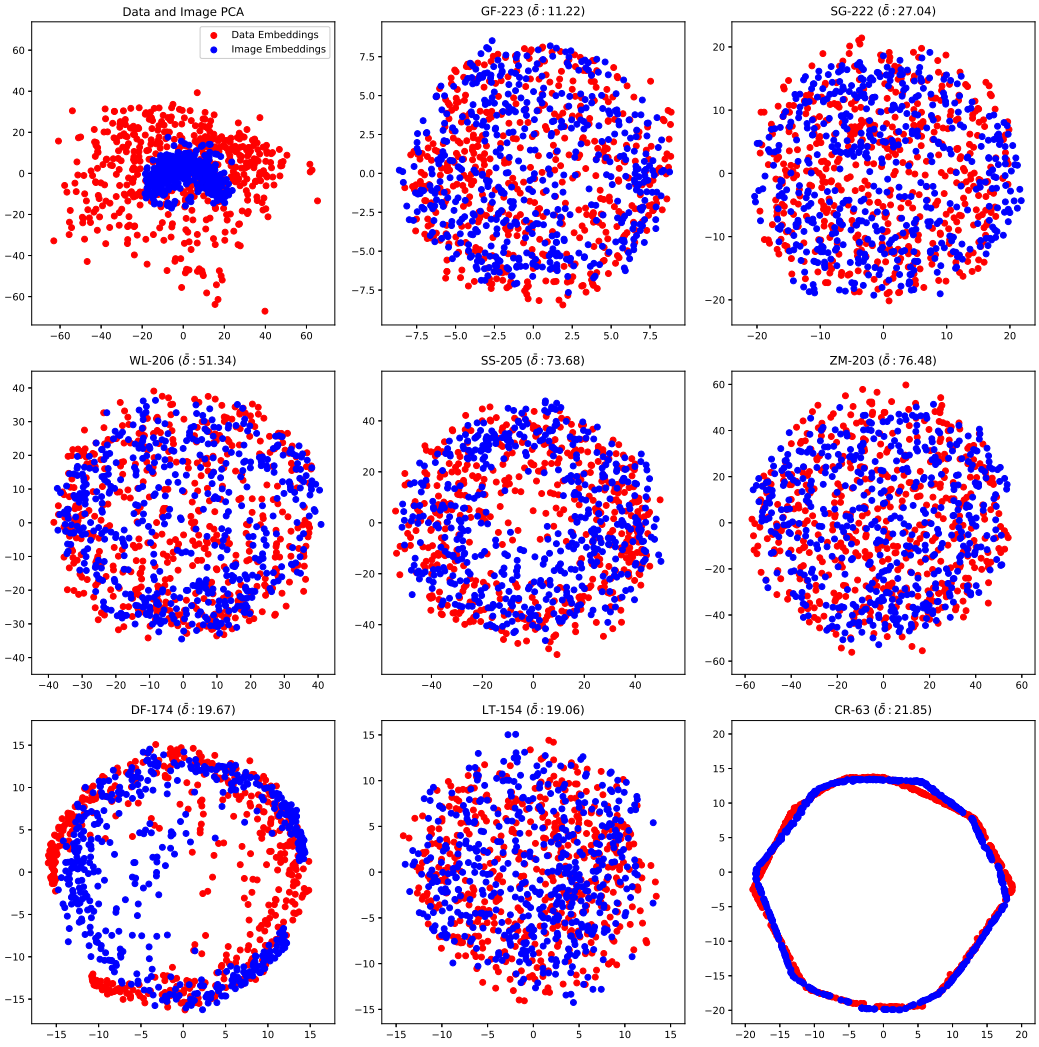


Figure C.1: Visualisation of the PCA projections for both data and image embeddings across different models. The red dots represent data embeddings, while the blue dots represent image embeddings. The projections are plotted for the raw image and data (first plot) and for the embeddings from models GF-223, SG-222, WL-206, SS-205, ZM-203, DF-174, LT-154, and CR-63, with their respective average pair-wise Euclidean distances ($\bar{\delta}$) indicated in parentheses. Distances between data and image embeddings within the projections illustrate how the model encodes similarities between the two modalities.

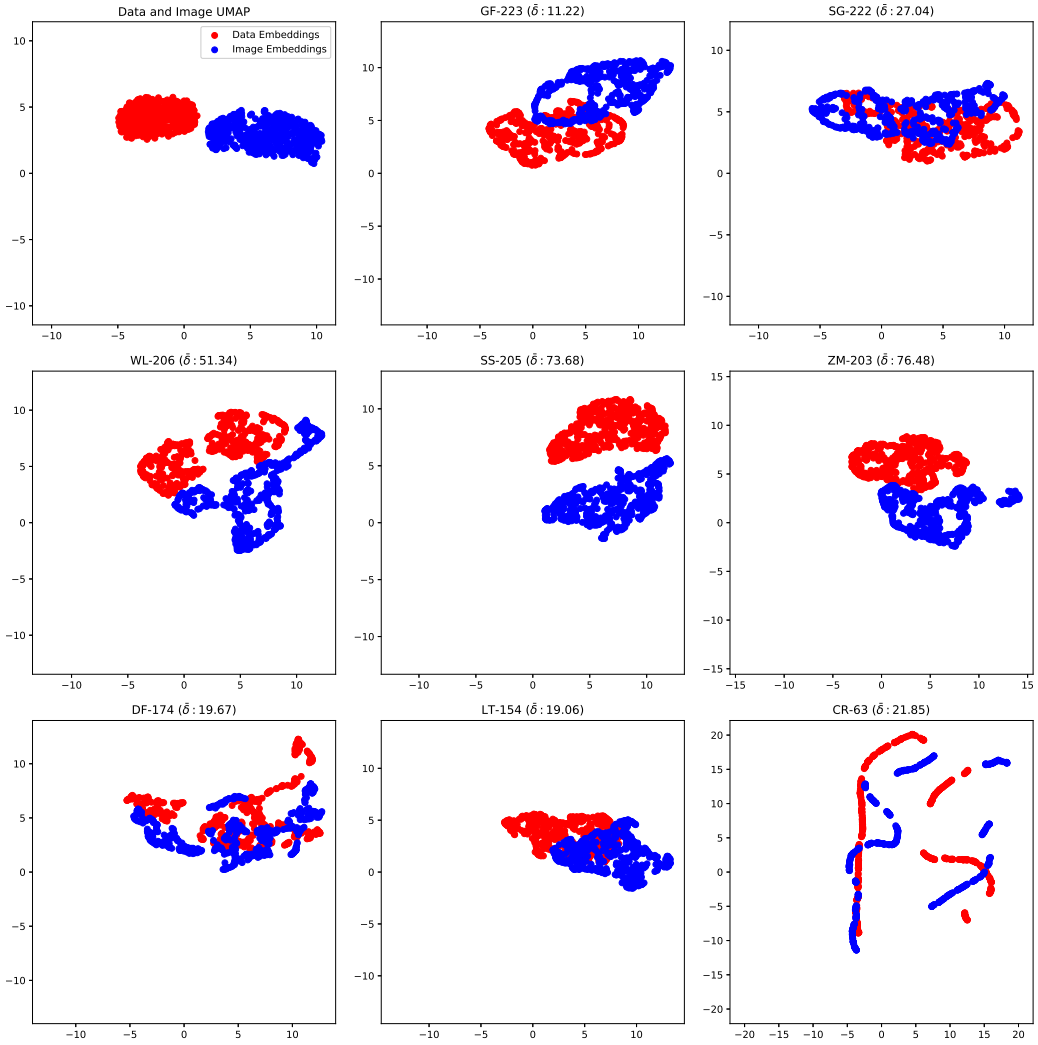


Figure C.2: Visualisation of the UMAP projections for data and image embeddings with a neighbourhood size of 15, Euclidean metric, and two components. The red dots represent data embeddings, while the blue dots represent image embeddings. The projections are plotted for the raw image and data (first plot) and for the embeddings from models GF-223, SG-222, WL-206, SS-205, ZM-203, DF-174, LT-154, and CR-63, with their respective average pair-wise Euclidean distances ($\bar{\delta}$) indicated in parentheses. Together with the PCA visualisation, the UMAP projections provide a comprehensive view of the alignment between the embeddings from different models.

Name	GF-223	SG-222	WL-206	SS-205	ZM-203	DF-174	LT-154	CR-63
GPU Count	4	4	4	2	4	8	2	2
Runtime (days)	1.23	1.23	1.06	0.33	1.39	0.18	0.33	0.38
batch_size	32	32	32	32	32	32	32	8
image_clip_outliers	outer	outer	outer	outer	outer	outer	outer	outer
data_clip_outliers	inner	inner	inner	inner	inner	outer	outer	outer
data_resize	1000	1000	1000	512	1000	512	512	256
vision_layers	3	3	2	2	2	2	2	4
vision_patch_size	3	3	3	3	3	3	3	3
embed_size	24	64	128	128	128	32	32	32
embed_dim	1	1	3	4	4	3	3	4
vision_width	64	64	64	64	64	64	64	256
learning_rate	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
optimiser	Adam	Adam	Adam	Adam	Adam	Adam	Adam	Adam
adam_betas	[0.9,0.98]	[0.9,0.98]	[0.9,0.98]	[0.9,0.98]	[0.9,0.98]	[0.9,0.98]	[0.9,0.98]	[0.9,0.98]
warm_restart_steps	2000	2000	2000	2000	2000	2000	2000	2000
epochs	500	500	500	650	650	650	450	450
training_loss	0.426	0.281	0.287	0.139	0.124	0.735	0.061	0.613

Table C.1: Hyperparameters used for contrastive embedding models.

GLOBAL INFORMATION RETRIEVAL IN DEEP LEARNING USING WIENER FILTERS

George Strong[†1], Deborah Pelacani Cruz[†1], Oscar Bates[2], Carlos Cueto[2], Javier Cudeiro[1], Jiashun Yao[1] and Lluís Guasch[1]

Imperial College London

INTRODUCTION & METHODS

Data comparison lies at the heart of machine learning: for many applications, simplistic loss functions - such as the L2 loss that rely on local element-wise differences between samples - have taken preference. Such metrics are notorious for producing low-quality results.

We address this issue by introducing a new convolutional approach to data comparison that uses Wiener filters to naturally incorporate global information. We demonstrate this concept by using it to define a new loss function, Wiener loss, and a generative model, Wiener diffusion.

A convolutional Wiener filter **v** that provides the best least squares match between two samples **x** and **d** is computed by the well known equation:

$$\mathbf{v} = (\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\mathbf{x}$$

where **D** represents convolution with **d**.

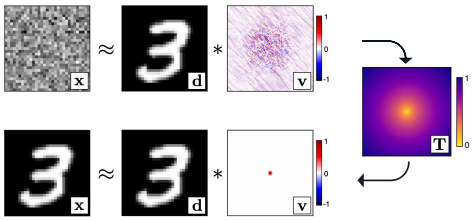


Figure 1. Data comparison using least squares convolutional Wiener filters **v**, where **x** is a predicted sample, **d** is a target sample and **T** is a monotonic penalty function.

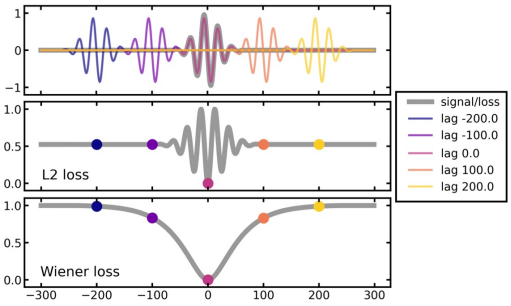


Figure 2. Top: a target signal (grey), and 5 duplicates shifted w.r.t the x-dimension. **Middle and bottom:** Loss evaluation between a continuously shifted duplicate signal and the target.

- The Wiener filter globally matches one entire sample to the other through convolution, without assuming any local element-wise relationship between samples.
- As illustrated in **Figure 1,** when samples **x** and **d** are equivalent to one another, the resulting Wiener filter is a zero-lag delta function.
- To determine the similarity of two samples, we apply an element-wise multiplication of the matching Wiener filter with a monotonically decreasing penalty function **T** centred at zero-lag.
- The more similar the samples are in a global sense, the closer the matching Wiener filter gets to a zero-lag delta function.
- As demonstrated in **Figure 2,** this gives rise to a loss function that is **sensitive to the global structure of the data**, rather than just the local element-wise differences.

WIENER LOSS

In this section we introduce the Wiener loss and demonstrate its potential for training neural networks on a high-dimensional imputation task. The Wiener loss is defined as follows:

$$L(\mathbf{x}, \mathbf{d}) = \frac{||\mathbf{Tv}||_2^2}{||\mathbf{v}||_2^2}$$

- The denominator of the Wiener loss is a normalization term required to prevent the loss function from driving all elements of the Wiener filter to zero.
- As shown in **Figure 3,** the U-Net trained with the Wiener loss outperforms traditional numerical and deep learning approaches, resulting in better **spatial continuity**, particularly within the skull, and **increased resolution** of inner-brain structures.
- The Wiener loss is primarily **driven by phase differences**. As such, it is naturally less sensitive to amplitude variations. In practice, using a hybrid loss that combines the Wiener loss with the MSE loss has been fruitful.
- Future work will examine variations on the Wiener loss formulation, its generalisation properties, and its suitability for different optimisation strategies.

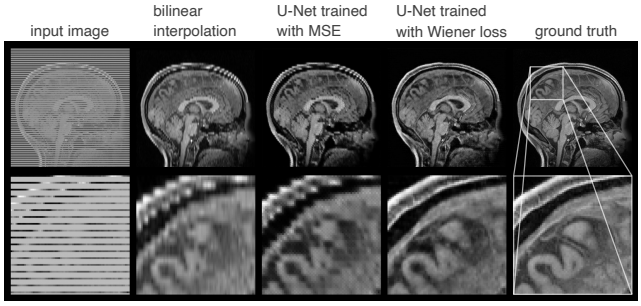


Figure 3. Comparison of traditional numerical and deep learning approaches at imputing a heavily under-sampled MRI dataset. **Top:** full-scale sagittal slice. **Bottom:** zoomed in region.

WIENER DIFFUSION

In this section we introduce the Wiener diffusion, a non-parametric generative model that constructs an energy function by a summation over *N* randomly selected data samples:

$$E(\mathbf{x}) = \sum_{i=1}^{N} \frac{||\mathbf{Tv}_i||_2^2}{||\mathbf{v}_i||_2^2}$$

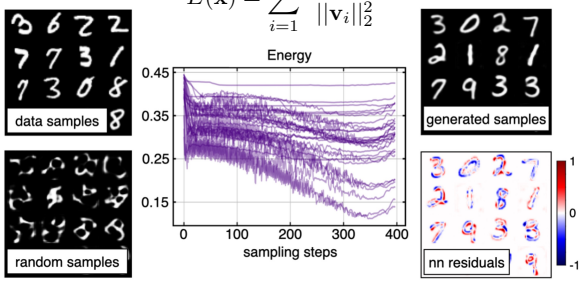


Figure 4. Wiener diffusion applied within the latent space of an auto-encoder using 600 latent samples from the MNIST dataset.

- The Wiener diffusion generates new samples by running a Langevin diffusion process on the energy function, as illustrated in **Figure 4**.
- In contrast to current popular generative modelling techniques, the generative process **does not require model training**.
- Initial results demonstrate that it is **effective on datasets of limited size**, as demonstrated in both **Figure 4** and **Figure 5**.

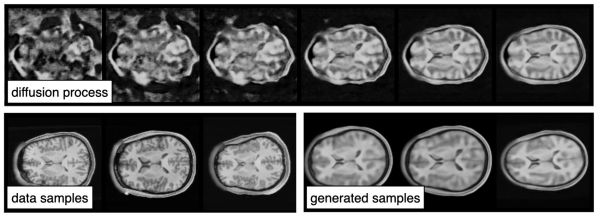


Figure 5. Wiener diffusion applied within the latent space of an auto-encoder using 500 latent samples from an MRI dataset of axial slices.

CONTRIBUTIONS

1. We have introduced a novel and revolutionary method for the global comparison of data samples that overcomes the key limitations of conventional loss functions.
2. We have demonstrated that this approach: **a) improves model training** compared to element-wise loss functions, **promoting spatial coherence** and resulting in **enhanced perceptual similarity; b) does not rely on large datasets** to produce high quality results, particularly for generative modelling and **c) a computationally efficient** and robust way to incorporate global information.
3. Despite showcasing our approach in two scenarios, we emphasize that the idea is general and has **immense potential for application throughout machine learning.**

[1] Department of Earth Science and Engineering, Imperial College London
[2] Department of Bioengineering, Imperial College London
[†] Equal contribution to this work

Convolve and Conquer: Data Comparison with Wiener Filters (Extended Abstract)

Deborah Pelacani Cruz*
Department of Earth Science and Engineering
Imperial College London
London, SW7 2BX
deborah.pelacani-cruz18@imperial.ac.uk

George Strong*
Department of Earth Science and Engineering
Imperial College London
London, SW7 2BX
george.stronge14@imperial.ac.uk

Oscar Bates
Department of Bioengineering
Imperial College London
London, SW7 2BX
o.bates18@imperial.ac.uk

Carlos Cueto
Department of Bioengineering
Imperial College London
London, SW7 2BX
c.cueto@imperial.ac.uk

Jiashun Yao
Department of Earth Science and Engineering
Imperial College London
London, SW7 2BX
j.yao14@imperial.ac.uk

Lluis Guasch
Department of Earth Science and Engineering
Imperial College London
London, SW7 2BX
l.guasch08@imperial.ac.uk

Abstract

Quantitative evaluations of differences and/or similarities between data samples define and shape optimisation problems associated with learning data distributions. Current methods to compare data often suffer from limitations in capturing such distributions or lack desirable mathematical properties for optimisation (e.g. smoothness, differentiability, or convexity). In this paper, we introduce a new method to measure (dis)similarities between paired samples inspired by Wiener-filter theory. The convolutional nature of Wiener filters allows us to comprehensively compare data samples in a globally correlated way. We validate our approach in two machine learning applications focused on medical imaging problems: Magnetic Resonance Imaging (MRI) data imputation and non-parametric generative modelling. Our results demonstrate increased resolution in reconstructed images with better perceptual quality and higher data fidelity compared to analogue conventional mean-squared-error implementations.

*These authors are equal contributors to this work

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

1 Introduction

Data comparison is essential to optimisation problems that aim to learn data distributions. Mean squared error (MSE) is a commonly used measure due to its simplicity, computational efficiency, and desirable mathematical properties like smoothness and differentiability (19). Despite these advantages, MSE-based reconstructions exhibit limited quantitative accuracy and perceptual fidelity (11; 19; 18; 22; 3) because the assumption that all data points are independent and of equal importance ignores contextual correlation. In the machine learning context, this behaviour is compounded by neural networks' biases towards low-frequency content (13; 23; 19), leading to high-frequency degradation and limited structural fidelity.

Exploration of better data-driven criteria has been comparatively overlooked in the machine learning community, which instead frequently adopts complex network architectures to address such low-fidelity issues. Previously proposed alternative metrics are often limited by training instabilities or high computational costs. For instance, the Structural Similarity Index Measure (SSIM) (20; 25) measures similarity based on luminance, contrast, and data structures, but produces unstable training results (10); perceptual losses (8) compare data representations within pre-trained convolutional neural networks' latent spaces, but are limited by costly evaluations and potential biases from its network architecture and training data; the discriminator in Generative Adversarial Networks (GANs) (4) offers another neural-network-based loss function but is susceptible to training instability issues; the Focal Frequency Loss (FFL) (7) proposes a Fourier-domain distance method aiming to limit high-frequency degradation, but further research is required to understand its range of applications; the Wasserstein distance from optimal transport theory (12; 16) can be used as a loss function for classification and generative modelling, but is computationally expensive and limited to non-negative, normalised signals.

In this work, we propose a new data-comparison strategy using Wiener filters that uses convolutional analysis to compare data samples without assuming local element-wise relationships. We are able to promote full-spectrum data recovery and improve data fidelity by leveraging its contextual awareness. The method is formulated in a least-squares framework and is therefore stable and cost-effective compared to the aforementioned metrics. We showcase the potential of our Wiener formulation as an optimisation metric in a machine-learning medical data imputation problem and as a new class of non-parametric generative models similar to the one proposed by (9). In these experiments, networks can better capture data distributions, boost perceptual quality compared to MSE, and generate realistic data samples from complex latent spaces.

2 Convolutional Data Comparison

We leverage Wiener filters as a measure of similarity, and use them to drive optimisation processes that match one data sample to another. This approach is inspired by the work of (21) on Adaptive Waveform Inversion (AWI), where the strategy is applied in the context of seismic imaging. A Wiener filter can be defined as the convolutional filter that provides the best least-squares match between two signals. Mathematically, for signals $\mathbf{x}$ and $\mathbf{y}$ represented as column vectors, this filter is given by:

$$\mathbf{v}(\mathbf{x}, \mathbf{y}) = (\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{x}, \tag{1}$$

where $\mathbf{v}$ is the Wiener filter, and $\mathbf{Y}$ is the Toeplitz matrix that represents convolution with signal $\mathbf{y}$. If signals $\mathbf{x}$ and $\mathbf{y}$ are vectors representing two-dimensional data, then $\mathbf{Y}$ is given by a doubly block Toeplitz matrix (5). In full, equation (1) can be interpreted as the cross-correlation of $\mathbf{y}$ with $\mathbf{x}$, deconvolved by the autocorrelation of $\mathbf{y}$. The size of the resulting Wiener filter determines the degree of contextual awareness within data samples. We choose to work with full-lag Wiener filters in order to capture correlation information between each data point against all others.

The Wiener filter that best matches two identical signals is given by the Dirac delta function - the identity function of the convolution operation. We thereby present a new loss function, the Wiener Loss, $\mathcal{L}_{\mathrm{W}}$, that maximises the likelihood of the Dirac delta function under a multivariate Gaussian distribution of mean defined as data-matching full-lag Wiener filters, thereby forcing the predicted samples towards the target samples in a global sense:

$$\mathcal{L}_{\mathrm{w}}(\mathbf{x}_\theta, \mathbf{y}) = \frac{1}{2}||\mathbf{W}\left\{\mathbf{v}(\mathbf{x}_\theta, \mathbf{y}) - \boldsymbol{\delta}\right\}||^2. \tag{2}$$

Due to its least-squares nature, the Wiener Loss retains desirable optimisation properties inherent to MSE, such as smoothness, convexity, and differentiability. The form of matrix $\mathbf{W}$, which can be interpreted as a whitening matrix, is considered a hyper-parameter that can be used to tailor the behaviour of the Wiener Loss to the required task.

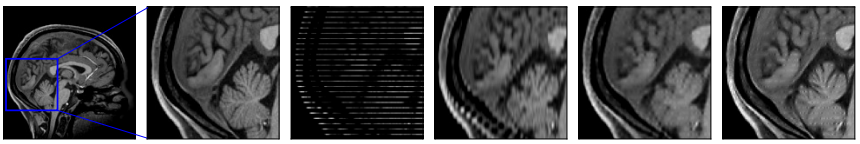
In a similar fashion, we introduce the Wiener diffusion: a new generative model that uses Wiener filters to define a non-parametric energy-based model, $E(\mathbf{x})$. Energy-based models enable sampling of complex data distributions whose probability densities are computationally intractable by representing the non-normalised negative log-likelihood of the data distribution (15). The energy function that defines the Wiener diffusion is given by the following:

$$E(\mathbf{x}) = \sum_{i=1} \frac{1}{2} \frac{||\mathbf{T}\mathbf{v}(\mathbf{x}_\theta, \mathbf{y})||^2}{||\mathbf{v}(\mathbf{x}_\theta, \mathbf{y})||^2} + \frac{\gamma}{2}||\boldsymbol{\delta} \otimes (\mathbf{v}(\mathbf{x}, \mathbf{y}_i) - \boldsymbol{\delta})||^2. \tag{3}$$

The first term of equation (3) drives the data-matching Wiener filters towards zero-lag spikes by minimising the Rayleigh quotient of a hyper-parameter penalty matrix $\mathbf{T}$ and Wiener filters $\mathbf{v}(\mathbf{x}, \mathbf{y}_i)$. The second term ensures the energy based model is sensitive to amplitude information by encouraging the zeroth-lag of the filters to have an amplitude of 1. Samples are then iteratively generated using a Lagenvin diffusion process initialised with random noise.

3 Experiments

Imputation (2) is a well-established technique across the medical imaging field that addresses sparse or limited resolution acquisitions through statistical estimation. In the first experiment we trained a supervised U-net (14) with MSE, our Wiener Loss, equation (2), and compared the results to a bicubic interpolation. We use T1-weighted MRI scans of healthy adult brains from the HCP Young Adult database (17) to train our network (see supplementary materials for specifications). Our results are shown in Figure 1, where qualitative results are superior under the Wiener Loss implementation, and quantitatively corroborated in Table 1.



(a) GT (b) GT (c) Degraded (d) Bicubic (e) MSE (f) Wiener Loss (Ours)

Figure 1: Ground truth (a-b) and zoomed outputs from simulated under-sampled scans (c) imputed using bicubic interpolation (d), U-net (14) with MSE (e) and U-net (14) with Wiener Loss (f). The Wiener Loss is able to recover finer structures of the scan that are aliased or blurred in other methods.

	MAE	MSE	Wiener Loss	SSIM	LPIPS	FID
Bicubic	$4.09 \cdot 10^{-4}$	$2.6 \cdot 10^{-4}$	0.21	0.8	**0.32**	**4.12**
MSE	$1.93 \cdot 10^{-4}$	$1.35 \cdot 10^{-4}$	0.14	0.91	**0.11**	**1.66**
Wiener Loss	$2.22 \cdot 10^{-4}$	$1.71 \cdot 10^{-4}$	$9.16 \cdot 10^{-2}$	0.9	$\mathbf{7.93 \cdot 10^{-2}}$	**0.23**

Table 1: MRI imputation and interpolation metrics, averaged per sample on the validation dataset. Better distribution restoration of our method evidenced by improved **LPIPS** (24) and **FID** (6) scores.

In our second experiment, we use the database to demonstrate the generative capabilities of our Wiener filter energy-based model (3), sampled in the latent space of a trained autoencoder (see supplementary materials for specifications). Figure 2 shows an example diffusion process of our

method towards realistic-looking samples, alongside the data-matching Wiener filters progressively focusing towards zero-lag delta functions. We stress that our Wiener diffusion is non-parametric and, although we use it on the latent space, the generative process does not require model training.

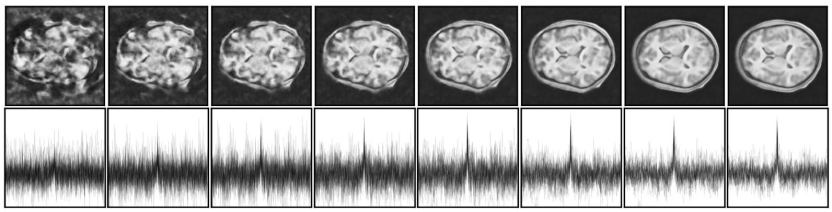


Figure 2: An illustration of the time-stepped diffusion process (top) alongside the 30 closest Wiener filters to the sample $\mathbf{x}_t$ (bottom), as defined by (3).

4 Contributions

We introduce a new data comparison method based on Wiener filters that is well suited to a wide range of medical imaging problems. We show that this strategy can be formulated as an optimisation metric capable of improving the fidelity of medical imputation outputs. This could have a knock-on effect on post-processing and image analysis of clinical datasets (1). Our metric retains desirable mathematical properties and is readily scalable with a computational complexity of $O(log\, n)$. We also demonstrate that our strategy can be formulated as a generative energy-based model that can truthfully sample complex latent spaces. This presents new possibilities for efficient expansion of current medical datasets, while also serving as a prior generator or as a regularisation term in imaging workflows. In addition, the non-parametric nature of our Wiener diffusion offers protection against training set biases, as only the most relevant data samples can be selected for the energy function.

Ethical Statement

Our research demonstrates that our new methodology for data comparison can statistically enhance images from limited-resolution scans and generate new samples from the same distribution. Despite these advantages, we highlight an ethical concern about potential biases stemming mainly from exclusive statistical representation of the training data and network architecture. These biases pose a significant risk in identifying misrepresented pathologies and can mislead clinical analyses. In this work, we used an MRI dataset of healthy adult brains to prove and validate our concept, but we emphasise the need for further research with diverse datasets to substantiate our findings.

References

[1] Mikael Brudfors, Yael Balbastre, Parashkev Nachev, and John Ashburner. A tool for super-resolving multimodal clinical mri. *arXiv preprint arXiv:1909.01140*, 2019.

[2] Adrian V Dalca, John Guttag, and Mert R Sabuncu. Unsupervised data imputation via variational inference of deep subspaces. *arXiv preprint arXiv:1903.03503*, 2019.

[3] Bernd Girod. What's wrong with mean-squared error? *Digital images and human vision*, pages 207–220, 1993.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[5] Per Christian Hansen. Deconvolution and regularization with toeplitz matrices. *Numerical algorithms*, 29(4):323–378, 2002.

[6] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[7] Liming Jiang, Bo Dai, Wayne Wu, and Chen Change Loy. Focal frequency loss for image reconstruction and synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13919–13929, 2021.

[8] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[9] Antoine Liutkus, Umut Simsekli, Szymon Majewski, Alain Durmus, and Fabian-Robert Stöter. Sliced-wasserstein flows: Nonparametric generative modeling via optimal transport and diffusions. In *International Conference on Machine Learning*, pages 4104–4113. PMLR, 2019.

[10] Jean-François Pambrun and Rita Noumeir. Limitations of the ssim quality metric in the context of diagnostic imaging. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 2960–2963. IEEE, 2015.

[11] Thrasyvoulos N Pappas, Robert J Safranek, and Junqing Chen. Perceptual criteria for image quality evaluation. *Handbook of image and video processing*, 110, 2000.

[12] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[13] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019.

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[15] Yang Song and Diederik P Kingma. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.

[16] Matthew Thorpe, Serim Park, Soheil Kolouri, Gustavo K Rohde, and Dejan Slepčev. A transportation L^p distance for signal analysis. *Journal of mathematical imaging and vision*, 59(2):187–210, 2017.

[17] David C Van Essen, Kamil Ugurbil, Edward Auerbach, Deanna Barch, Timothy EJ Behrens, Richard Bucholz, Acer Chang, Liyong Chen, Maurizio Corbetta, Sandra W Curtiss, et al. The human connectome project: a data acquisition perspective. *Neuroimage*, 62(4):2222–2231, 2012.

[18] Zhou Wang and Alan C Bovik. Modern image quality assessment. *Synthesis Lectures on Image, Video, and Multimedia Processing*, 2(1):1–156, 2006.

[19] Zhou Wang and Alan C Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE signal processing magazine*, 26(1):98–117, 2009.

[20] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[21] Michael Warner and Lluís Guasch. Adaptive waveform inversion: Theory. *Geophysics*, 81(6):R429–R445, 2016.

[22] Hong Ren Wu and Kamisetty Ramamohan Rao. *Digital video image quality and perceptual coding*. CRC press, 2017.

[23] Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *arXiv preprint arXiv:1901.06523*, 2019.

[24] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[25] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on computational imaging*, 3(1):47–57, 2016.

Convolve and Conquer: Data Comparison with Wiener Filters

Deborah Pelacani Cruz*, George Strong*, Carlos Cueto, Jiashun Yao, Oscar Bates, Lluis Guasch

Imperial College London

NEURAL INFORMATION PROCESSING SYSTEMS 2025

MOTIVATION & METHODS

Data comparison is essential to optimisation problems that aim to learn data distributions. Mean squared error (MSE) is a commonly used measure due to its simplicity, computational efficiency, and desirable mathematical properties, but the assumption that all data points are independent and of equal importance ignores contextual correlation and harms perceptual fidelity. Previously proposed alternative metrics are often limited by training instabilities or high computational costs.

$$\mathbf{v}(\mathbf{x},\mathbf{y}) = (\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{x}$$

We propose a new data comparison strategy using data-matching Wiener filters. Our strategy uses convolutional analysis to compare data samples **without assuming local element-wise relationships**. We leverage Wiener filters as a measure of similarity, exploiting the property that a Wiener filter, **v(x, y)** defined above, between two identical samples is given by the Direct delta function, i.e. the convolutional identity (Figure 1).

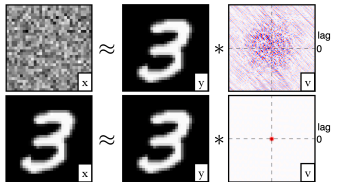


Figure 1: Top: visual illustration of two-dimensional Wiener filters **v** that match MNIST sample **y** to random noise **x**. Bottom: matching **y** to **x** where **x** = **y**, demonstrating that the Wiener filter when **x** = **y** is given by the Dirac delta function.

WIENER LOSS

We present a new loss function, **Wiener loss**, that uses Wiener filters to drive the optimisation processes that match one data sample to another in a global sense. We demonstrate its potential training a supervised U-net for MRI imputation. The loss is defined as follows:

$$\mathcal{L}_{\mathrm{w}}(\mathbf{x}_\theta,\mathbf{y}) = \frac{1}{2}||\mathbf{W}\left\{\mathbf{v}(\mathbf{x}_\theta,\mathbf{y}) - \boldsymbol{\delta}\right\}||^2$$

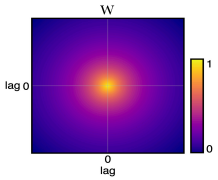


Figure 2: Whitening matrix used in the imputation problem that penalises error between the Wiener filter and a zero-lag Direc delta function.

» Whitening matrix **W** (**Figure 2**) is considered a hyper-parameter that can be used to tailor the behaviour of the Wiener Loss, seeing that it is monotonically increasing towards zero lag.

» The Wiener loss maximises the likelihood of the Dirac delta function under a multivariate Gaussian distribution of mean defined as data-matching full-lag Wiener filters, iteratively driving sample **x** to match sample **y**.

» It **retains desirable optimisation properties inherent to MSE** due to its least-squares nature.

» Superior qualitative (**Figue 3**) and quantitative (**Table 1**) results are achieved under the Wiener Loss implementation.

WIENER LOSS (cont'd)

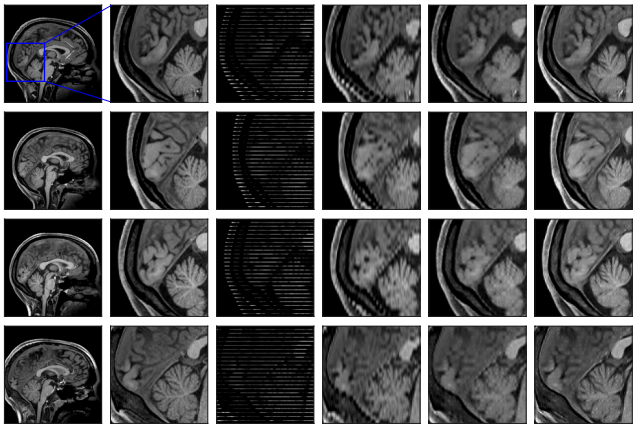


Figure 3: Ground truth (a-b) and zoomed outputs from simulated under-sampled scans (c) imputed using bicubic interpolation (d), U-net with MSE (e) and U-net with Wiener Loss (f). The Wiener Loss is able to recover finer structures of the scan that are aliased or blurred in other methods.

	MAE	MSE	Wiener Loss	SSIM	LPIPS	FID
Bicubic	$4.09 \cdot 10^{-4}$	$2.6 \cdot 10^{-4}$	0.21	0.8	**0.32**	**4.12**
MSE	$1.93 \cdot 10^{-4}$	$1.35 \cdot 10^{-4}$	0.14	0.91	**0.11**	**1.66**
Wiener Loss	$2.22 \cdot 10^{-4}$	$1.71 \cdot 10^{-4}$	$9.16 \cdot 10^{-2}$	0.9	$\mathbf{7.93 \cdot 10^{-2}}$	**0.23**

Table 1: MRI imputation and interpolation metrics, averaged per sample on the validation dataset. Better distribution restoration of our method evidenced by improved **LPIPS** and **FID** scores.

WIENER DIFFUSION

In a similar fashion, we introduce the **Wiener diffusion**: a new generative model that uses Wiener filters to define a non-parametric energy-based model, **E**, defined below.

$$E(\mathbf{x}) = \sum_{i=1} \frac{1}{2}\frac{||\mathbf{T}\mathbf{v}(\mathbf{x}_\theta,\mathbf{y})||^2}{||\mathbf{v}(\mathbf{x}_\theta,\mathbf{y})||^2} + \frac{\gamma}{2}||\boldsymbol{\delta} \otimes (\mathbf{v}(\mathbf{x},\mathbf{y}_i) - \boldsymbol{\delta})||^2$$

» The first equation term drives the data-matching Wiener filters towards zero-lag spikes by minimising the Rayleigh quotient of a hyper-parameter penalty matrix **T** (**Figure 5**) and Wiener filters **v**.

» The second term ensures the energy based model is sensitive to amplitude information by encouraging the zeroth-lag of the filters to have an amplitude of 1.

WIENER DIFFUSION (cont'd)

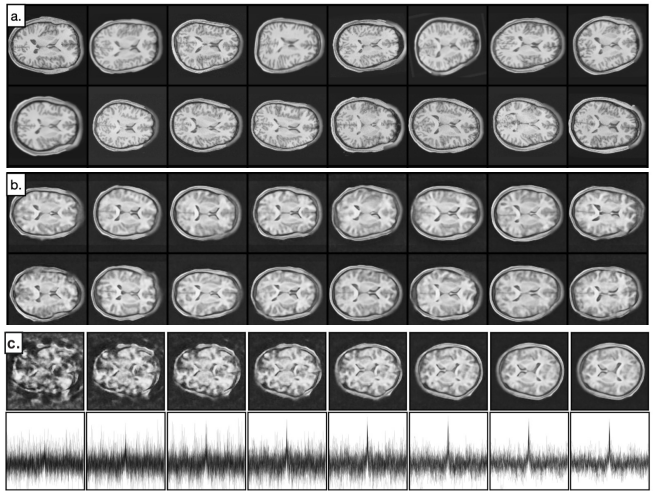


Figure 3: (a) MRI ground truth data samples, (b) Realistic samples generated with Wiener Diffusion, (c) illustration of the time-stepped diffusion process (top) alongside the 30 closest Wiener filters to the sample.

» Samples are iteratively generated using Lagenvin dynamics **and do not require network training.**

» The diffusion proccess generates realistic-looking MRI samples while progressively driving the Wiener filters toward zero-lag delta functions (**Figure 4**)

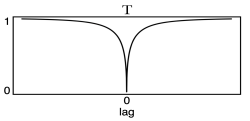


Figure 5: Function that penalises non-zero lag energy in the energy-based model Wiener filters.

CONTRIBUTIONS

1. We present a **new method to measure (dis)similarities between paired samples inspired by Wiener-filter theory.**We formulate our method as a **novel loss function and a non-parametric generative model** that iteratively match paired data samples by driving their **data-matching Wiener filter** towards the **convolutional identity**;
2. The **convolutional nature of our methods promote preservation of contextual information**, which is readily suited for a **wide a range of machine learning and inverse problems**;
3. We showcase our methods in 1) a machine-learning medical imputation problem, where **higher data fidelity and spatial coherence is achieved under our Wiener loss**; and 2) as a generative **energy-based model** that can **sample complex data distributions** without network training.

* Equal contribution to this work