Echocardiography Generation using Deep Generative Models

Emmanuel Oladokun

New College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Michaelmas 2025

Abstract

Echocardiography is a widely used, low-cost, and non-invasive imaging modality for the assessment and management of cardiac disease. It provides real-time visualisation of cardiac anatomy and function and supports the estimation of clinically important physiological parameters such as ejection fraction. Despite its clinical utility, echocardiography is highly operator-dependent, particularly for transesophageal echocardiography (TEE), and the availability of large, well-annotated datasets remains limited due to acquisition complexity, data protection constraints, and the cost of expert annotation. These challenges hinder the development and evaluation of data-driven methods for echocardiographic analysis.

This thesis investigates the use of deep generative models to address data scarcity and standardisation challenges in echocardiography. First, a pipeline termed `HeartToEcho` is proposed to generate anatomically consistent pseudo-images corresponding to 21 standard TEE views from three-dimensional heart models. These pseudo-images can be translated into realistic synthetic echocardiograms, as demonstrated for the mid-esophageal four-chamber view, and their utility is evaluated through qualitative assessment and improved performance on a downstream left-ventricle segmentation task.

Building on this, a data-efficient diffusion-based framework, EchoLoRA, is introduced for synthetic TEE image generation from semantic masks. By combining low-rank adaptation with pretrained diffusion models, the method enables effective cross-domain learning from transthoracic echocardiography (TTE) to TEE under limited data availability, while maintaining strong adherence to anatomical conditioning.

Finally, the thesis extends generative echocardiography from images to videos by proposing EchoLVFM, a one-step latent video flow matching model for temporally coherent echocardiogram generation. The approach supports variable-length sequences and masked video conditioning, enabling tasks such as video completion and animation from partial inputs. Together with the preceding chapters, this work demonstrates how anatomically conditioned generative models can help mitigate data scarcity and variability in echocardiography, while supporting more consistent generation of standard views and clinically relevant dynamics.

Echocardiography Generation using Deep Generative Models



Emmanuel Oladokun

New College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2025

Without counsel plans fail, but with many advisers they succeed

— Proverbs 15:22

Acknowledgements

Firstly, I would like to thank my supervisors Vicente and Jurica who have been with me throughout this journey! I could not have completed this DPhil without you. A big thank you especially for putting up with my last-minute nature and need for a deadline. Vicente, thank you for all the meetings and the reassuring wisdom that you provided to steer me back on track when I strayed. Jurica! Thank you for these last few years, providing direction, the GE resources, and for your brilliant sense of humour. Keep up the stylish Teams backgrounds. I am really glad we got to meet in person (you are a lot taller than you look on Teams :D)

To Sarina and Musa, thank you so much for your supervision, your advice, proof-reading, help brainstorming, and so much more.

Next, I would like to thank my family. I am so blessed to have been brought into this life with you all. Daddy, thank you for how you instilled the importance of education into me from such a young age. Not only did you lead by example, you also ensured that I had all I needed to continue succeeding. Mummy, for your consistent prayers, care, and love. I am certain those prayers have done so much heavy lifting in bringing me to where I am today.

To my siblings, Sis Seun and Bro Laide, Dare, and Bimbo. Thank you so much for putting up with me! For going through life, making mistakes, then sharing your hard earned wisdom with me so that I don't repeat your mistakes.

To all my aunties, and uncle. Auntie Adé! A special thank you for all the meals!

To the beautiful Dami, thank you for always being there for me.

To the wonderful O'Connells for their selflessness, constant love, and support. You opened your home to me multiple times and welcomed me as one of your own. I am very grateful.

To all the wonderful housemates I have had throughout this DPhil: Anthony, Hannah, Joel, and Jess! I hope you are having better luck with estate agents. To all my friends, thank you so much for being in my life! Pip, Paul, Charlie, Friends From Church, My SABS Cohort, IBME Friends: Alexis, Abdullah, Eoin, Marshia, and others! You all made this experience better. Everyone in the Grau group, past and current. It was a pleasure!

Finally, I would like to thank New College and the generous donors for all financial support they provided me throughout my DPhil. The travel grants alone were instrumental in me being able to go to Vancouver and Daejeon! Also, thank you to the catering staff for all the guest nights and brunches.

I have so much to be grateful for. Thank you God. To you be all the Glory. Amen.

Bí ẹmi bá wà ìrètí ò pin

Abstract

Echocardiography is a widely used, low-cost, and non-invasive imaging modality for the assessment and management of cardiac disease. It provides real-time visualisation of cardiac anatomy and function and supports the estimation of clinically important physiological parameters such as ejection fraction. Despite its clinical utility, echocardiography is highly operator-dependent, particularly for transesophageal echocardiography (TEE), and the availability of large, well-annotated datasets remains limited due to acquisition complexity, data protection constraints, and the cost of expert annotation. These challenges hinder the development and evaluation of data-driven methods for echocardiographic analysis.

This thesis investigates the use of deep generative models to address data scarcity and standardisation challenges in echocardiography. First, a pipeline termed `HeartToEcho` is proposed to generate anatomically consistent pseudo-images corresponding to 21 standard TEE views from three-dimensional heart models. These pseudo-images can be translated into realistic synthetic echocardiograms, as demonstrated for the mid-esophageal four-chamber view, and their utility is evaluated through qualitative assessment and improved performance on a downstream left-ventricle segmentation task.

Building on this, a data-efficient diffusion-based framework, EchoLoRA, is introduced for synthetic TEE image generation from semantic masks. By combining low-rank adaptation with pretrained diffusion models, the method enables effective cross-domain learning from transthoracic echocardiography (TTE) to TEE under limited data availability, while maintaining strong adherence to anatomical conditioning.

Finally, the thesis extends generative echocardiography from images to videos by proposing EchoLVFM, a one-step latent video flow matching model for temporally coherent echocardiogram generation. The approach supports variable-length sequences and masked video conditioning, enabling tasks such as video completion and animation from partial inputs. Together with the preceding chapters, this work demonstrates how anatomically conditioned generative models can help mitigate data scarcity and variability in echocardiography, while supporting more consistent generation of standard views and clinically relevant dynamics.

Contents

List of Figures **xi**

List of Tables **xix**

List of Abbreviations **xxiii**

1 Introduction **1**

1.1 Motivation 1
1.2 Thesis Structure 3
1.3 Thesis Contributions 4
1.4 Publications 5

2 Background **7**

2.1 Cardiovascular System 7
2.1.1 Cardiac Cycle 8
2.1.2 Ejection Fraction 10
2.2 Cardiac Imaging 11
2.2.1 Ultrasound 11
2.2.2 Echocardiography 13
2.3 Deep Learning in Medical Image Analysis 18
2.4 Generative Modelling 23
2.4.1 Contrastive Unpaired Translation 24
2.4.2 Diffusion Models 27
2.4.3 Flow Matching 31
2.4.4 Metrics 32

3 Datasets **37**

3.1 Internal TEE Dataset 37
3.2 Public Datasets 40
3.2.1 SSM Dataset 40
3.2.2 CAMUS 41

4 Unpaired Transesophageal Echocardiography Generation 43
4.1 Introduction . . . 44
4.1.1 Contributions . . . 46
4.2 Pseudo-image Generation . . . 46
4.2.1 Data . . . 47
4.2.2 Methods . . . 47
4.2.3 Results . . . 50
4.2.4 Discussion . . . 61
4.2.5 Conclusion . . . 63
4.3 Synthetic Image Generation . . . 64
4.3.1 Methods . . . 64
4.3.2 Experiments . . . 66
4.3.3 Results & Discussion . . . 70
4.4 Conclusion . . . 75

5 Cross-Modality Generation using LoRA Diffusion 77
5.1 Introduction . . . 78
5.1.1 Towards Diffusion-Based Approaches . . . 79
5.1.2 Cross-Domain Adaptation via Low-Rank Fine-Tuning . . . 81
5.1.3 Contributions . . . 82
5.2 Methods . . . 83
5.2.1 Elucidated Diffusion Model . . . 83
5.2.2 Low-Rank Adaptation . . . 85
5.2.3 MaskR^2 . . . 85
5.3 Experiments . . . 86
5.3.1 Data . . . 86
5.3.2 Image Generation . . . 87
5.4 Results & Discussion . . . 90
5.5 Conclusion . . . 94

6 One-step Echocardiogram Generation via Latent Video Flow Matching 97
6.1 Introduction . . . 98
6.1.1 Contributions . . . 100
6.2 Methods . . . 100
6.2.1 Linear Flow . . . 100
6.2.2 Mean Flow . . . 101
6.2.3 EchoLVFM . . . 102
6.3 Experiments . . . 107
6.3.1 Data . . . 107
6.3.2 Training . . . 108

6.3.3 Run Configurations 108
6.3.4 Evaluation 110
6.4 Results 112
6.4.1 VAE 112
6.4.2 Video Generation 113
6.5 Discussion 117
6.5.1 VAEs 117
6.5.2 Video Quality 119
6.6 Conclusion 120

7 Conclusion **123**
7.1 Contributions 123
7.1.1 TEE Generation using Anatomical Models - `HeartToEcho` 124
7.1.2 From Transthoracic to Transesophageal - `EchoLoRA` 124
7.1.3 One-step Echocardiogram Video Generation - `EchoLVFM` 124
7.2 Limitations and Future Work 125
7.2.1 TEE Generation using Anatomical Models 125
7.2.2 From Transthoracic to Transesophageal 125
7.2.3 One-step Video Generation 126
7.2.4 Practical Considerations and Open Directions 126

References **129**

List of Figures

2.1 **The Heart.** Image from [15]. 8

2.2 **Cardiac Pressure Evolution in a Cardiac Cycle.** a) Aortic Pressure versus Time, and b) Ventricular Pressure versus Volume. 9

2.3 **LVEF Calculation Using Two Orthogonal Views.** Left ventricular ejection fraction (LVEF) is estimated using the biplane method of disks, also known as Simpson's biplane method. Left ventricular widths measured in two orthogonal views are used to estimate the major and minor diameters of an elliptical cross-section. The length of the longest axis extending from the left ventricular apex to the mitral valve plane is computed and divided into N sections, where N denotes the number of disks. The volume of each disk is calculated by multiplying the area of the corresponding ellipse by the disk thickness, given by L_{max}/N. The total left ventricular volume is obtained by summing the volumes of all disks. This procedure is performed for both the end-diastolic and end-systolic frames, and the resulting volumes are used to compute the LVEF. Adapted from [16]. . . 11

2.4 **Wave Behaviour at a Boundary.** A wave incident on a boundary between media of different acoustic impedance is partially reflected and partially transmitted. Image from [19]. 12

2.5 **The Two Main Types of Echocardiography.** a) Transthoracic, and b) Transesophageal. 14

2.6 **Transesophageal Echocardiography Views 1 - 8.** A complete list of all the views in a comprehensive TEE Examination. *Imaging plane* shows an idealised representation of each view, *3D Model* shows typical probe position and plane of view, *2D TEE Image* shows an echo of each view, *Acquisition Protocol* details the typical set of actions required to acquire each view, and *Structures Images* contains a list of the structures that should be visible in each view. Image from Hahn et al. [25]. 15

2.6 **Transesophageal Echocardiography Views 9 - 15.** (continued). 16

2.6 **Transesophageal Echocardiography Views 16 - 24.** (continued). 17

2.6 **Transesophageal Echocardiography Views 25 - 28.** (continued). 18

2.7 **An Example CNN.** This CNN has 3 encoding layers and 2 decoding layers. In each encoding layer, many 3 × 3 kernels move around the feature maps extracting features. As the input travels deeper into the network, it is encoded into a latent space. Furthermore, the feature maps are downsampled, hence the local receptive field of each kernel increases. Moving from the 3rd to the 4th layer represents an entrance into the decoding section where the feature maps are upsampled. A CNN of this structure is well-suited for tasks like semantic segmentation. Through training of the kernels, the encoding section would figure out 'what' is present in the image, and the decoding section determines 'where'. Image from [57]. 20
2.8 **ResNet vs DenseNet.** High-level illustrations of how the ResNet and DenseNet tackle issues relating to the flow of gradients through a network. The ResNet uses addition to connect all preceding features, whilst DenseNet concatenates them. Image from [58]. 21
2.9 **An Example Transformer.** The Vision Transformer splits the input image into patches, flattens them, linearly embeds them, and adds positional embeddings so the model retains information about each patch's spatial location. The resulting vectors are passed through a series of encoder layers. Within each encoder, features are first normalised and then processed by a self-attention module, which learns relationships between different image patches as training progresses. A skip connection is then applied, encouraging stable gradient flow and preserving information across layers, before a feed-forward network further transforms the features. Image from [59]. 21
2.10 **Self- and Cross-Attention Mechanisms.** Self-attention learns relationships within a single input by deriving queries (Q), keys (K), and values (V) from the same feature representation, allowing contextual reasoning across elements of the input e.g. patches at different positions in an image or words at different parts of a sentence. In contrast, cross-attention allows information from an external source to influence the target representation by computing queries from the target features and keys and values from a separate input. This mechanism enables conditioning on other data, such as semantic masks, text, or features from other imaging modalities. 22
2.11 **CycleGAN Architecture.** a) The model contains two generators responsible for generating images in the other domain, and two discriminators for assessing the quality of the generated images in each domain, b) The original image in X should be retrievable from the projection in domain Y and generator F, c) The original image in Y should be retrievable from the projection in domain X and generator G. Image from [74]. 25

2.12 **CUT Process.** The original and synthetic image representations are compared at different layers within the encoding section of the generator. By ensuring that the representation of corresponding patches i.e., positive pairs, are encoded in a similar way, the content is preserved spatially but appearance is allowed to vary. The representations are first transformed by a learnable Multi-Layer Perceptron H_l before being passed to the loss term as this has been shown to improve performance. Image from [81]. 26

3.1 **DICOM to Data.** The preprocessing steps to go from the raw acquisitions to TEE images ready for labelling and use in training models. 38
3.2 **TEE-A Image & Mask.** Example TEE-A acquisition complete with semantic labels for the **LV**. 39
3.3 **TEE-B Image & Mask.** Example TEE-B acquisition complete with semantic labels for the **LA**, **LV**, **RA**, and **RV**. 40
3.4 **SSM Dataset Anatomical Heart Model Structures.** All structures represented in the heart models of the SSM Dataset, grouped by types. Left-atrium (LA), Left-ventricle (LV), Pulmonary Valve (PV), and Pulmonary Artery (PA). 42
3.5 **Anatomical Models and Structure Labels.** Structure labels for an example heart model from the SSM Dataset. *Vein Rings* show the rings where the veins are cropped, including the LAA. *Tissue* shows the myocardium of the four major chambers, namely, LV, RV, LA, RA, as well as the wall of the cropped aorta and pulmonary artery. *Valve Borders* shows the planes added at inlets and outlets of the four major chambers. 42

4.1 **The 20 Initial Heart Models.** Image from [113]. 47
4.2 **Mid-Esophageal 5-Chamber View Entry.** A view entry contains: an idealised image of the view, a 3D model, the 2D ultrasound image, the protocol for its acquisition, and the salient structures that should be visible in the image. Image from [25]. 48
4.3 **Generation of Mid-Esophageal 4-Chamber View and Off-Planes.** The slice definition uses the LV, mitral-, and tricuspid- valve as landmarks. The middle column shows the plane defined by the landmarks whilst the slices either side are generated by randomly rotating about the sampling axes. 51
4.4 **Generation of Mid-Esophageal Mitral Commissural View and Off-Planes.** The slice definition uses the LV, mitral valve, and LA as landmarks. The middle column shows the plane defined by the landmarks whilst the slices either side are generated by randomly rotating about the sampling axes. 52

4.5 **V1. Generated Mid-Esophageal Five-Chamber View.** Landmarks shown are the mitral valve, tricuspid valve, and left ventricle. The plane passes through the point $\frac{2.5\cdot\vec{p}_{av}+\vec{p}_{mv}+\vec{p}_{tv}}{4.5}$ with normal $(\vec{p}_{tv}-\vec{p}_{mv})\times(\vec{p}_{tv}-\vec{p}_{lv_{apex}})$. 53

4.6 **V2. Generated Mid-Esophageal Four-Chamber View.** Landmarks shown are the left ventricle, mitral valve, and tricuspid valve. The plane passes through the point $(\vec{p}_{lv}+\vec{p}_{mv}+\vec{p}_{tv})/3$ with normal $(\vec{p}_{tv}-\vec{p}_{mv})\times(\vec{p}_{tv}-\vec{p}_{lv})$. 53

4.7 **V3. Generated Mid-Esophageal Mitral Commissural View.** Landmarks shown are the left ventricle, left atrium, and mitral valve. The plane passes through the point $(\vec{p}_{mv}+\vec{p}_{la}+\vec{p}_{lv})/3$ with normal $(\vec{p}_{la}-\vec{p}_{lv})\times(\vec{p}_{la}-\vec{p}_{mv})$. 54

4.8 **V4. Generated Mid-Esophageal Two-chamber View.** Landmarks shown are the left ventricle, mitral valve, and left atrial appendage. The plane passes through the point $(\vec{p}_{lv}+\vec{p}_{mv}+\vec{p}_{laa})/3$ with normal $(\vec{p}_{mv}-\vec{p}_{lv})\times(\vec{p}_{mv}-\vec{p}_{laa})$. 54

4.9 **V5. Generated Mid-Esophageal Long-Axis View.** Landmarks shown are the left ventricle, mitral valve, and aortic valve. The plane passes through the point $(\vec{p}_{lv}+\vec{p}_{av}+\vec{p}_{mv})/3$ with normal $(\vec{p}_{mv}-\vec{p}_{lv})\times(\vec{p}_{mv}-\vec{p}_{av})$. 54

4.10 **V6. Generated Mid-Esophageal Aortic Valve Long-Axis View.** Landmarks shown are the left atrium, mitral valve, and aortic valve. The plane passes through the point $(\vec{p}_{la}+\vec{p}_{mv}+\vec{p}_{av})/3$ with normal $(\vec{p}_{mv}-\vec{p}_{la})\times(\vec{p}_{mv}-\vec{p}_{av})$. 55

4.11 **V10. Generated Mid-Esophageal Aortic Valve Short-Axis View.** Landmarks shown are the right atrium, aortic valve, and pulmonary valve. The plane passes through the point $(\vec{p}_{ra}+\vec{p}_{av}+\vec{p}_{pv})/3$ with normal $(\vec{p}_{av}-\vec{p}_{ra})\times(\vec{p}_{av}-\vec{p}_{pv})$. 55

4.12 **V11. Generated Mid-Esophageal Right Ventricular Inflow-Outflow View.** Landmarks shown are the tricuspid valve, aortic valve, and pulmonary valve. The plane passes through the point $(\vec{p}_{tv}+\vec{p}_{av}+\vec{p}_{pv})/3$ with normal $(\vec{p}_{av}-\vec{p}_{tv})\times(\vec{p}_{av}-\vec{p}_{pv})$. 55

4.13 **V12. Generated Mid-Esophageal Modified Bicaval Tricuspid Valve View.** Landmarks shown are the tricuspid valve, superior vena cava, left atrium, and right atrium. The interatrial septum point is approximated as $\vec{p}_{ias}=(\vec{p}_{la}+\vec{p}_{ra})/2$. The plane passes through the point $(\vec{p}_{tv}+\vec{p}_{ias}+\vec{p}_{svc})/3$ with normal $(\vec{p}_{ias}-\vec{p}_{tv})\times(\vec{p}_{svc}-\vec{p}_{tv})$. 56

4.14 **V13. Generated Mid-Esophageal Bicaval View.** Landmarks shown are the left atrium, right atrium, superior vena cava, and inferior vena cava. The interatrial septum point is approximated as $\vec{p}_{ias}=(\vec{p}_{la}+\vec{p}_{ra})/2$. The plane passes through the point $(\vec{p}_{ias}+\vec{p}_{ivc}+\vec{p}_{svc})/3$ with normal $(\vec{p}_{ivc}-\vec{p}_{ias})\times(\vec{p}_{ivc}-\vec{p}_{svc})$. 56

4.15 **V14. Generated Upper-Esophageal Right and Left Pulmonary Veins View.** Landmarks shown are the pulmonary artery, left superior pulmonary valve, and right inferior pulmonary valve. The plane passes through the point $(\vec{p}_{pa} + \vec{p}_{lspv} + \vec{p}_{ripv})/3$ with normal $(\vec{p}_{lspv} - \vec{p}_{pa}) \times (\vec{p}_{lspv} - \vec{p}_{ripv})$. 57

4.16 **V16. Generated Transgastric Basal Short-Axis View.** Landmarks shown are the left ventricle and tricuspid valve. The plane passes through the point $(\vec{p}_{mv} + \vec{p}_{tv})/2$ with normal $(\vec{p}_{mv} - \vec{p}_{lv_{apex}})$. 57

4.17 **V17. Generated Transgastric Mid-Papillary Short-Axis View.** The shown landmark is the left ventricle. The plane passes through the point $(\vec{p}_{rv} + \vec{p}_{lv})/2$ with normal $(\vec{p}_{mv} - \vec{p}_{lv_{apex}})$. 57

4.18 **V18. Transgastric Apical Short-Axis View.** Landmarks shown are the mitral valve and tricuspid valve. The plane normal is defined as $\vec{p}_{mv} - \vec{p}_{lv_{apex}}$. The plane passes through the point $\frac{(\vec{p}_{lv_{apex}} + \vec{p}_{rv_{apex}})}{2} + 10\frac{\vec{p}_{mv} - \vec{p}_{lv_{apex}}}{\|\vec{p}_{mv} - \vec{p}_{lv_{apex}}\|}$, corresponding to a forward-shifted midpoint between the ventricular apices. 58

4.19 **V19. Generated Transgastric Right Ventricular Basal View.** Landmarks shown are the tricuspid valve, pulmonary valve, and left ventricle. The plane passes through the point $(\vec{p}_{tv} + \vec{p}_{pv} + \vec{p}_{lv_{apex}})/3$ with normal $(\vec{p}_{pv} - \vec{p}_{tv}) \times (\vec{p}_{pv} - \vec{p}_{lv_{apex}})$. 58

4.20 **V20. Generated Transgastric Right Ventricular Inflow-Outflow.** Landmarks shown are the right atrium, tricuspid valve, and pulmonary valve. The plane passes through the point $(\vec{p}_{ra} + \vec{p}_{tv} + \vec{p}_{pv})/3$ with normal $(\vec{p}_{tv} - \vec{p}_{ra}) \times (\vec{p}_{tv} - \vec{p}_{pv})$. 59

4.21 **V21. Deep Transgastric Five-Chamber View.** Landmarks shown are the left ventricle, aorta, and aortic valve. The plane passes through the point $(\vec{p}_{lv} + \vec{p}_{aorta} + \vec{p}_{av})/3$ with normal $(\vec{p}_{aorta} - \vec{p}_{lv}) \times (\vec{p}_{aorta} - \vec{p}_{av})$. . . . 59

4.22 **V22. Generated Transgastric Two-Chamber View.** Landmarks shown are the left ventricle, left atrial appendage, and mitral valve. The plane passes through the point $(\vec{p}_{lv} + \vec{p}_{laa} + \vec{p}_{mv})/3$ with normal $(\vec{p}_{laa} - \vec{p}_{lv}) \times (\vec{p}_{laa} - \vec{p}_{mv})$. 59

4.23 **V23. Generated Transgastric Right Ventricle Inflow View.** Landmarks shown are the right ventricular, right atrium, and tricuspid valve. The plane passes through the point $(\vec{p}_{rv} + \vec{p}_{ra} + \vec{p}_{tv})/3$ with normal $(\vec{p}_{ra} - \vec{p}_{rv}) \times (\vec{p}_{ra} - \vec{p}_{tv})$. 60

4.24 **V24. Generated Transgastric Long-Axis View.** Landmarks shown are the left ventricle, tricuspid valve, and aortic valve. The plane passes through the point $(\vec{p}_{lv} + \vec{p}_{tv} + \vec{p}_{av})/3$ with normal $(\vec{p}_{tv} - \vec{p}_{lv}) \times (\vec{p}_{tv} - \vec{p}_{av})$. . . 60

4.25 **ME2CH and ME4CH Pseudo-images.** Pseudo-images generated using slices from different heart models by adding an ultrasound cone, gaussian blur, and speckle noise. 61

4.26 **Real Mid-Esophageal Mitral Commissural Views.** The Red circle shows the Coronary Sinus and the yellow arrows point to faint chordae tendinae. These real acquisitions share the same high-level, dominant anatomical features as the generated slice shown in Fig. 4.7. 62
4.27 **V25. Descending Aorta SAX View.** View 25 out of 28 views required for a comprehensive TEE. Image from [25]. 62
4.28 **Synthetic TEE Generation and Image Segmentation Pipeline.** Using landmarks, desired TEE planes are extracted from each heart model. These slices from the anatomical model serve as the anatomical guidance (conditioning masks) for the synthetic images. Next, pseudo-images are made by adding the acquisition cone and applying blurring, shadowing, and resizing transformations. The image synthesis phase concludes with pseudo-images being passed through a trained generator that transforms them into realistic synthetic images. These synthetic images and their masks can then be used to augment a real dataset for the chosen task. . 64
4.29 **Experimental Methods for Unpaired I2I Generation.** The real and pseudo-images are split into subsets dedicated to the image-to-image translation task and the segmentation task. Subsets used for image-to-image translation have the subscript 'I2I' and the segmentation subsets have the subscript 'Seg'. Once the generator is trained, the synthetic dataset $\hat{S}_{gen}$ is evaluated and also used to augment the real dataset. Finally, more nnUNet models are trained using the augmented dataset and their performance is also evaluated. 66
4.30 **Real vs Fake Quiz UI.** The experts and researchers used the shown tool for registering their answers (on a webpage, the question and response section does not overlap the cone). 68
4.31 **Segmentation Comparison of Difficult Cases.** Example segmentations generated by nnUNet models trained on: purely real data (Real), real data mixed with synthetic images generated by a CycleGAN {Real, $\hat{S}_{cyc}$}, and real data mixed with synthetic images generated by the CUT method {Real, $\hat{S}_{cut}$}. (Note that the 'Real' cases also segment the LV in the first two rows. Best visualised on a screen.). 73

5.1 **Desired Qualities of Generative Models.** An ideal generative model would exhibit all three qualities—diversity, high quality, and quick sampling—simultaneously. In practice, these properties exist in tension; for instance, diffusion models typically achieve diverse and high-quality outputs but at the cost of longer sampling times during inference. . . . 79
5.2 **TEE & TTE.** The first row shows TEE images of the ME4CH and ME2CH views. The second row shows acquisitions from the CAMUS dataset (more details in Sec. 3.2.2) and are of the A2CH and A4CH views. . . . 82

5.3 `EchoLoRA` **TTE → TEE Pipeline.** MaskR^2 is used to adapt the TEE Masks to the style expected by the TTE pretrained model. In this use case, the channel originally used for the left ventricular epicardium (LV$_{epi}$) conditioning is now responsible for generating the right-hand-side of the heart in TEE. Next, the base model is adapted in a targeted manner to generate synthetic TEE data. After training, inference is performed using real masks and masks sampled from SSMs to generate synthetic TEE datasets. Finally, the synthetic datasets are used to augment a real TEE dataset, which can be used for a downstream task. 84

5.4 **Synthetic TEE Images.** This figure shows how the adapted models perform at inference using real masks from the test set and SSM masks. In all cases, the generative model sees the right side of the heart as one structure due to MaskR^2. 'All Weights' represents a generative model trained on TEE images from scratch. The layer groups are Cross-attention (CA), Self-attention (SA), Convolution (C), and Linear (L). 93

6.1 **EchoLVFM Framework.** *Training:* Each echocardiogram is encoded frame-by-frame into a latent representation, temporally augmented to F frames, and noised. A masked latent counterpart x_m is formed by zeroing all but a small subset of observed frames; x_m is supplied as conditioning. A binary padding vector $p \in \{0,1\}^F$ marks valid frames from padded frames in the augmented input. A global conditioning scalar ϕ (e.g. ejection fraction) is also supplied. With timesteps $r < t$, the spatiotemporal U-Net $\hat{u}_\theta$ predicts the conditional average velocity over $[r,t]$ from the noised input together with the conditioning bundle $c = \{x_m, \phi, p\}$. *Inference:* a partial video containing as little as a single observed frame, together with a target ϕ and random noise, is passed to the trained model. One-step integration with `EchoLVFM-Mean` produces the generated video. The padding vector p additionally controls the output video length up to F. . 103

6.2 **Expectations of Sampling Schemes.** Frame sampling methods for generating x_m from x. 110

6.3 **VAE Performance in Preliminary Experiments.** The models are evaluated on both reconstruction and generation tasks using the FID metric at intervals of 50 epochs on the validation dataset. The black line indicates the lowest FID score achieved by a model trained in the 16f8 latent space. All lines above the black line correspond to 16f8, whilst all below correspond to 4f8 latent space results. 112

6.4 **Optimal Inference Steps Linear.** This figure shows how the best and worst performing run configurations progressed on Reconstruction and Generation tasks when only one frame is provided in x_m. Evaluations with differing guidance scales are also shown. The maximum number of inference steps is 250. 114

6.5 **`EchoLVFM-Linear` Runs.** Performance of `EchoLVFM-Linear` across all tasks and at different *nmf* levels. Configuration R07 is omitted from this plot, as its values are of a different order and significantly skew the visualisation. 115
6.6 **`EchoLVFM-Mean` Runs.** Performance of `EchoLVFM-Mean` across all tasks and at different *nmf* levels. Configuration R07 is omitted from this plot, as its values are of a different order and significantly skew the visualisation. 116

List of Tables

4.1 **List of TEE Views and Their Generation State.** *Probe Region:* the region of the transducer during image acquisition, *View No.:* The number of the view, *View Name:* Descriptor of the view, *Generation Status:* whether or not the view was successfully generated, *Reason:* If the view could not be generated, why was this the case? Reason A means the view could not be generated due to the absence of the full aorta structure in the models. Reason B means it was due to the absence of the full atrial appendage. Acronyms: Mid-esophageal (ME), Upper-esophageal (UE), Transgastric (TG) , short axis (SAX), long axis (LAX), Aortic Valve (AV), Right ventricular (RV). 50

4.2 **Training Schedule and Hyperparameters That Produced the Best Generators.** LR = Learning Rate; E[] = Epoch Range; TTUR = Two time-scale update rule. Discriminator LR = TTUR_Multiplier × Generator LR. 67

4.3 **Participant Results on the Quiz with Expert Performance and Researcher Performance Separated.** Each person was shown 60 real (R) and 60 synthetic (S) images. **Accuracy was rounded to 1 d.p whilst frequencies were rounded to the nearest integer. ** 95% confidence interval*. 70

4.4 **FID Between an Unseen Set of TEE Images and Synthetic Sets.** A lower score represents better fidelity between the generated and real images. 71

4.5 **Accuracy on the Synthetic Images Only.** A lower score shows more difficulty in identifying the generator's image as synthetic. 71

4.6 **Dice score achieved on the test set.** The column headers show real datasets with varying percentages of R_{train}, whilst the first column shows the source of the synthetic data and delta scores. Each non-grey element corresponds to the performance of an nnUNet trained on the union of the column and row datasets ($< column > \bigcup < row >$). **Blue** = Best Column Delta Score; **Bold =** Best Overall Dice Score. 73

5.1 **Image Generation Configuration for Pretraining and Finetuning.** The hyperparameters and configurations used to train the base diffusion model and fine-tune the LoRA parameters. Feature dimensions and attention layers are applied symmetrically across the downsampling and upsampling blocks. *Acronyms:* Self-attention (SA), Cross-attention (CA). 88

5.2 **Generation & Multiclass Segmentation Results.** This table summarises the performance of the LoRA-adapted models on both image-quality metrics and a downstream multiclass segmentation task. “All Weights” refers to the model trained purely on TEE images. Each adapter updates only the specified layer groups—Cross-attention (CA), Self-attention (SA), Convolution (Conv), and Linear—via LoRA. “Mask Source” indicates whether the synthetic echos were generated with real TEE masks or SSM masks. The “Per-class Dice” column reports the Dice score for each chamber, shown here as the delta relative to the baseline model. **Bold** highlights the overall best scores, and colored entries mark the top performer for each individual class.†Best FID Score in [134]. ‡Scores achieved from nnUNet trained on purely real data. 91

5.3 **Average Segmentation Dice.** For each chamber the Dice is computed per patient and averaged over the patients in which the chamber is present in the reference; the four per-chamber means are then averaged into a single **Average Dice**. Per-class entries are shown as the delta relative to the baseline. **Bold** marks the best Average Dice; colored entries mark the best delta per chamber. ‡Scores from nnUNet trained on purely real data. 92

6.1 **Video Generation Training Configuration.** *Model Configuration*: Feature dimensions and attention layers are applied symmetrically across the downsampling and upsampling blocks unless otherwise stated. *Training*: Learning Rate (LR) . *Flow Parameters*: $P(\mathcal{L}_{\text{MLF}})$ and $P(\mathcal{L}_{\text{RMMF}})$ represent the probability of selecting the masked linear flow, and the masked mean flow loss function at training time. p_{uncond} represents the probability of dropping conditioning during training. λ_{rec} is the weighting applied to the reconstruction loss. 109

6.2 **Run Configurations.** The configurations shown were explored to identify effective training settings for `EchoLVFM-Linear` and `EchoLVFM-Mean`. This range of configurations enables identification of the key factors influencing model performance. 111

6.3 **VAE CAMUS Image Reconstruction Metrics.** The performance of each VAE on a reconstruction task using the CAMUS dataset. MAE denotes the mean absolute error in pixel space. 112

6.4 **`EchoLVFM-Linear` Run Configuration Rankings.** This table shows the Rank-Sum and Min–max performance of each run across all metrics. Rank-Sum ranks the runs for each metric and sums their positions. Min–max rescales each metric by its range, subtracts the minimum, and then normalises the scores. This approach preserves magnitude information, allowing (near-)ties between runs to be distinguished. Gold, Silver, and Bronze shading highlight the top 3 configurations. 117

6.5 `EchoLVFM-Mean` **Run Configuration Rankings.** This table shows the Rank-Sum and Min–max performance of each run across all metrics. Rank-Sum ranks the runs for each metric and sums their positions. Min–max rescales each metric by its range, subtracts the minimum, and then normalises the scores. This approach preserves magnitude information, allowing (near-)ties between runs to be distinguished. R07 was not included in min-max normalisation as it strongly skews the values. Gold, Silver, and Bronze shading highlight the top 3 configurations. 117

6.6 **EchoLVFM Video Generation Performance.**Image (I), Text (T), Semantic Mask (M), Motion Mask (MM), Video (V), Partial Video (pV), and Scalar (S). †Although the authors do not share the number of inference steps used, the authors use DDPM [84] which is known to typically require several hundreds of steps. The authors of the original DDPM paper use 1000. ‡ The original CAMUS split contains videos with as little as 10 frames in its test set. However, the default FVD processes 16 frame batches. **Bold** shows the best overall score per metric. **Colour** shows the best generation scores. Baseline methods are reported for the reconstruction task only: they condition on spatial inputs rather than a target ejection fraction, and no prior work evaluates ejection-fraction-conditioned video generation on the CAMUS dataset. 118

6.7 **R07 Results.** $R07$ run performed the worst out of all datasets. This is thought to be because of the Pad And Observed mask scheme used. . . 119

List of Abbreviations

LA . . . . . Left Atrium

LV . . . . . Left Ventricle

LV$_{epi}$ . . . . . Left Ventricular Epicardium

RA . . . . . Right Atrium

RV . . . . . Right Ventricle

TTE . . . . . Transthoracic Echocardiography

TEE . . . . . Transesophageal Echocardiography

ME2CH . . . . . Mid-Esophageal 2-Chamber

ME4CH . . . . . Mid-Esophageal 4-Chamber

A2CH . . . . . Apical 2-Chamber

A4CH . . . . . Apical 4-Chamber

SAX . . . . . Short-Axis

LAX . . . . . Long-Axis

EDV . . . . . End-Diastolic Volume

ESV . . . . . End-Systolic Volume

EDA . . . . . End-Diastolic Area

ESA . . . . . End-Systolic Area

EF . . . . . Ejection Fraction

LVEF . . . . . Left-Ventricular Ejection Fraction

I2I . . . . . Image-to-Image

CNN . . . . . Convolutional Neural Network

GAN . . . . . Generative Adversarial Network

EDM . . . . . Elucidated Diffusion Model

LoRA . . . . . Low-Rank Adaptation

SSM . . . . . Statistical Shape Model

CA . . . . . Cross-Attention

SA . . . . . Self-Attention

VAE . . . . . Variational Autoencoder

DDPM . . . . . Denoising Diffusion Probabilistic Model

DDIM . . . . . Denoising Diffusion Implicit Model

FID . . . . . Fréchet Inception Distance

FVD . . . . . Fréchet Video Distance

LPIPS . . . . . Learned Perceptual Image Patch Similarity

SSIM . . . . . Structural Similarity Index Measure

PSNR . . . . . Peak Signal-to-Noise Ratio

MSE . . . . . Mean Squared Error

CAMUS . . . . . Cardiac Acquisitions for Multi-structure Ultrasound Segmentation

The greatest among you shall be your servant.
Whoever exalts himself will be humbled, and
whoever humbles himself will be exalted.

— Matthew 23:11-12

1 Introduction

Contents

1.1 Motivation **1**
1.2 Thesis Structure **3**
1.3 Thesis Contributions **4**
1.4 Publications **5**

1.1 Motivation

Today, medical imaging is an essential tool in healthcare. Its use can be found in screening, early diagnosis, treatment and follow-up. Multiple imaging techniques (termed modalities) are widely available in the clinic. These modalities, including Magnetic Resonance Imaging (MRI), Computed Tomography (CT) and Ultrasound have different strengths and limitations that tailor them for use in different clinical scenarios. Collectively, they allow the visualisation of many different parts of the human body, such as bones or soft tissue. Furthermore, they can provide measurements of several important physical and physiological parameters, making them indispensable in the effort to provide high-quality care.

Echocardiography (Echo) is a key contributor to the assessment and management of cardiac disease and is widely used as a first-line imaging modality for cardiac evaluation

[1]. Echocardiograms are produced using ultrasound waves to create real-time images of the heart. At the power levels used in clinics, these sound waves are painless, harmless, and the devices required to generate them are relatively low cost and portable [2]. These significant advantages of echo contribute to its widespread clinical use in both acute and outpatient settings. Despite echo images typically being 2D, clinically valuable approximations of important 3D variables such as chamber volumes can be calculated using information acquired from a combination of standard views. Moreover, the visualisation provided is key for diagnosing cardiac pathologies and assessing cardiac performance using physiological variables such as EF [3].

Two main forms of echocardiography are used in clinical practice: transthoracic echocardiography (TTE), and transesophageal echocardiography (TEE). TTE is non-invasive and typically used for routine cardiac assessment, whereas TEE provides superior visualisation of posterior cardiac structures and cardiac valves at the cost of being invasive [1]. As a result, TEE is often considered the reference standard for evaluating certain valvular and structural abnormalities, and is routinely used in interventional and perioperative settings [4].

Nevertheless, echocardiography has several significant limitations. One such difficulty is performing an echocardiographic examination without substantial training and experience [5], particularly in invasive forms such as TEE. It can be difficult to determine the probe position and to understand precisely which anatomical structures are being imaged. Differences in patient anatomy also make the standardisation of scanning with TEE more cumbersome. As a result, practitioners often rely heavily on experience and intuition to acquire appropriate views, leading to variability in image quality and diagnostic content.

Certain types of ultrasound have seen the implementation of tele-echocardiography, where images are acquired by a non-specialist with limited training and analysed remotely by an expert [2]. While this approach can improve access to expertise, it does not reduce the overall dependence on specialist knowledge. Machine learning methods, and more specifically deep learning-based approaches, have established themselves as a strong option for automating the analysis of echocardiograms. This includes detecting salient

anatomical structures [6], identifying cardiac pathologies [7, 8], and extracting clinically useful physiological variables such as EF [9].

However, these methods depend heavily on sizeable, high-quality datasets, which are difficult to acquire in general and particularly so for medical imaging. Due to its sensitive nature, medical data is protected, and its distribution and use are heavily regulated. Even when data is available, many deep learning methods require supervision, necessitating data annotation. Data annotation is costly and time-consuming, and is subject to both inter- and intra-operator variability [10, 11], which can impose an upper bound on achievable performance. Additionally, while some datasets [12] can be annotated by non-experts, medical imaging data often requires expert annotation.

Accordingly, echocardiography stands to benefit from the application of deep learning methods. The challenges outlined above motivate the development of approaches that can generate realistic and anatomically meaningful echocardiographic data, particularly for TEE, in order to support specialist training, algorithm development, and downstream clinical tasks. This thesis addresses them by exploring generative approaches for echocardiography, with an emphasis on standardised view generation, anatomical conditioning, cross-domain learning between TTE and TEE, and the controlled generation of echocardiogram sequences conditioned on clinically relevant parameters.

1.2 Thesis Structure

Chapters 1 - 3 motivate the research, provide relevant background to the methods used, establish where this work sits in literature, and present the datasets core to this thesis. Next, chapters 4 - 6 present the novel contributions of this thesis towards the generation of synthetic echocardiograms using deep generative models. Chapter 7 recaps this work, discusses the benefits and limitations of the contributions as well as their consequences in the field. Finally, potential avenues of future work are presented.

1.3 Thesis Contributions

The main contribution of this thesis is the development of a set of anatomically grounded and data-efficient deep generative methods for synthesising clinically usable echocardiographic data. These methods enable the generation of standardised TEE images and TTE videos under explicit anatomical control, addressing key challenges related to data scarcity, standardisation, and downstream usability.

Specifically, this thesis introduces a pipeline for generating 21 standard TEE views from anatomical heart models, proposes diffusion-based methods that leverage cross-domain learning to generate high-quality synthetic TEE images from semantic masks, and extends these ideas to temporally coherent echocardiogram video generation using latent flow matching conditioned on sparse image and physiological inputs. The utility of the generated data is demonstrated through extensive evaluation and downstream tasks, including segmentation.

Chapter 3: In this chapter, I describe the datasets used throughout this thesis. I detail their sources of origin, size, composition, and the pre-processing steps required to enable their use. Moreover, I explain how the real TEE datasets were annotated by myself and two expert clinicians.

Chapter 4: In this chapter, I present the `HeartToEcho` framework, which generates synthetic TEE images from slices extracted from anatomical models. I extend the methods developed by Gilbert et al. [13] to enable the generation of TEE pseudo-images. My methodology consists of two stages. The first one involves reproducing standard TEE views from heart models and converting them into pseudo-images. The second stage converts pseudo-images into synthetic TEE images, for which I evaluate multiple unpaired image-to-image translation methods. The generated images are rigorously evaluated using both qualitative and quantitative approaches. The qualitative evaluation involves recruiting clinicians and researchers to classify real and synthetic images via a custom user interface I developed. For the quantitative assessment, I use image quality metrics and assess the impact of the synthetic images on a downstream left-ventricle segmentation task.

Chapter 5: Generating synthetic data using `HeartToEcho` requires manual effort from the user to select image transformations for converting slices into pseudo-images. In this chapter, I propose `EchoLoRA`, which aims at addressing this limitation by generating images directly from semantic masks. Furthermore, while `HeartToEcho` relies exclusively on scarce TEE data, `EchoLoRA` is designed to learn from other imaging domains without requiring the data to share the same format as the TEE data. I perform extensive experiments to evaluate the generalisability of `EchoLoRA`, including tests using out-of-distribution masks as inputs. The generated images are assessed using a range of image quality metrics and further evaluated by augmenting a real dataset for a downstream multi-class segmentation task.

Chapter 6: The methods proposed in the previous chapters focus primarily on echocardiographic images rather than videos. In this chapter, I propose `EchoLVFM`, which explicitly accounts for the temporal nature of echocardiographic acquisitions by generating echocardiogram videos. Videos are generated by conditioning a latent video flow matching model on as little as a single echo frame and a global parameter; previous work typically fixes conditioning to a single frame [14]. `EchoLVFM` supports frame animation, video completion, and temporal upsampling. The generated videos are evaluated in terms of both image and temporal fidelity using multiple metrics.

1.4 Publications

Chapter 4: ***Oladokun, E.***, *Abdulkareem, M., Šprem, J., Grau, V.: Transesophageal Echocardiography Generation using Anatomical Models. In: Data Augmentation, Labelling, and Imperfections (DALI 2023), Lecture Notes in Computer Science, vol. 14379, pp. 43–52, Springer Nature Switzerland (2023)*

Chapter 5: ***Oladokun, E.***, *Ou, Y., Novikova, A., Kulikova, D., Thomas, S., Šprem, J., Grau, V.: From Transthoracic to Transesophageal: Cross-Modality Generation using LoRA Diffusion. In: International Workshop on Advances in Simplifying Medical*

Ultrasound (ASMUS 2025), *Lecture Notes in Computer Science*, *vol. 16165*, *pp. 120–129*, *Springer Nature Switzerland (2025)*

I lift up my eyes to the hills. From where does my help come?
My help comes from the Lord, who made heaven and earth.
He will not let your foot be moved; He who keeps you will not slumber.

— Psalm 121:1-3

2 Background

Contents

2.1 Cardiovascular System 7
2.1.1 Cardiac Cycle 8
2.1.2 Ejection Fraction 10
2.2 Cardiac Imaging 11
2.2.1 Ultrasound 11
2.2.2 Echocardiography 13
2.3 Deep Learning in Medical Image Analysis 18
2.4 Generative Modelling 23
2.4.1 Contrastive Unpaired Translation 24
2.4.2 Diffusion Models 27
2.4.3 Flow Matching 31
2.4.4 Metrics 32

This section explains in detail the concepts and methods utilised throughout this thesis as well as establishing the current state of the literature.

2.1 Cardiovascular System

The cardiovascular system consists of the heart, blood vessels, and blood. These components work together to:

- distribute oxygen and nutrients to all body tissues;

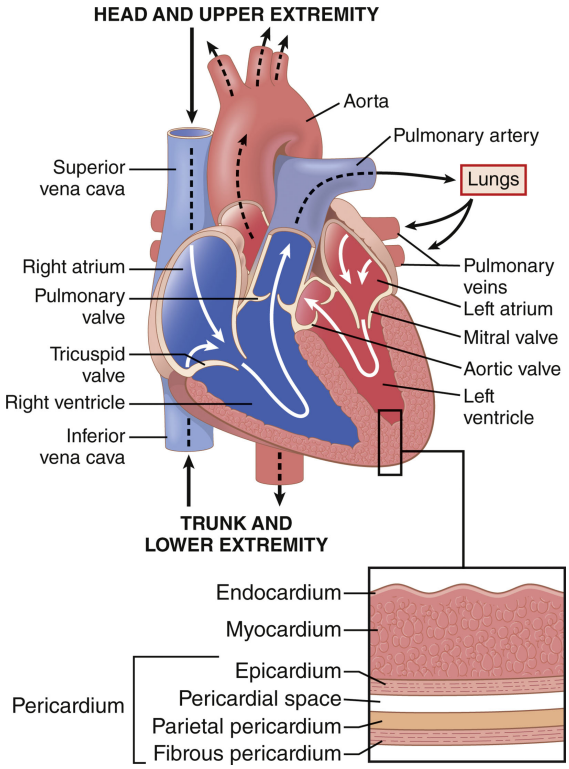


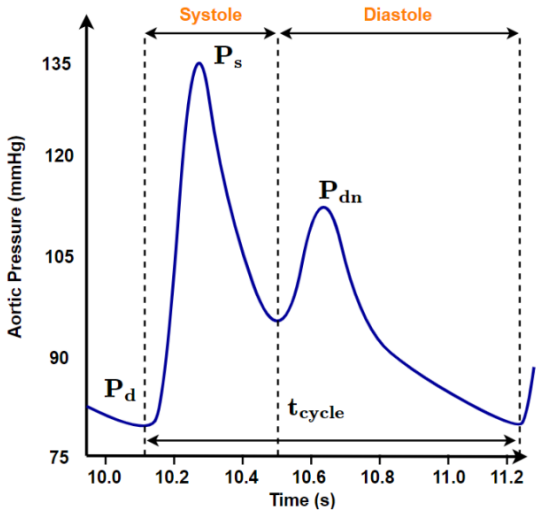
Figure 2.1: The Heart. Image from [15].

- remove carbon dioxide and metabolic waste from tissues and deposit them in the excretory organs;
- distribute water, electrolytes, and hormones throughout the body; and
- help regulate body temperature.

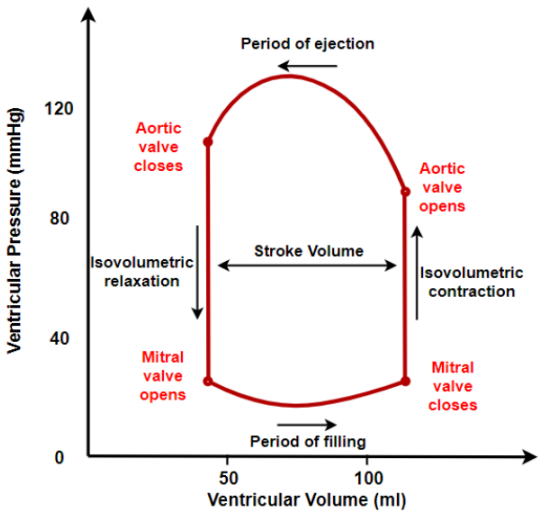
The cardiovascular system plays a key role in achieving homeostasis and is therefore indispensable. At the centre of the cardiovascular system lies the heart, shown in Fig. 2.1, a pulsatile pump responsible for driving blood around the body.

2.1.1 Cardiac Cycle

The sequence of events from the start of one heartbeat to the start of the next is termed the cardiac cycle. The cardiac cycle is separated into two major phases: diastole and systole. Diastole is known as the filling phase, while systole is the contraction phase. Variations in aortic and ventricular pressure with respect to time and volume are shown in Fig.2.2a and Fig.2.2b, respectively.



(a) Aortic pressure versus time during a cardiac cycle. Diastolic Pressure (P_d), Systolic Pressure (P_s), Dicrotic Notch Pressure (P_{dn}), and Cardiac Cycle Duration (t_{cycle}).



(b) Ventricular pressure versus ventricular volume during a cardiac cycle. The loop width is the stroke volume (SV) and the area is the stroke work done.

Figure 2.2: Cardiac Pressure Evolution in a Cardiac Cycle. a) Aortic Pressure versus Time, and b) Ventricular Pressure versus Volume.

Diastole commences with the closure of the aortic and pulmonary valves, followed by an isovolumetric relaxation of the ventricles. As ventricular pressure approaches a minimum, the atria fill with blood simultaneously. Eventually, atrial pressure exceeds ventricular pressure, causing blood to flow rapidly into the ventricles. As the ventricles fill, the rate of filling decreases in proportion to the reduction in pressure difference. Atrial contraction occurs at the end of diastole to increase this now depleted pressure difference and cause a final influx of blood into the ventricles. This final influx is responsible for approximately 20% of the total blood filling the ventricles per cardiac cycle [15].

A period of isovolumetric contraction follows ventricular filling, marking the start of the systolic phase. Ventricular pressure increases rapidly, causing the atrioventricular valves to close. All four valves are closed at this point, hence the isovolumetric nature of this phase. Once ventricular pressure exceeds the pressure in the aorta and pulmonary artery, the semilunar valves open, allowing the ejection of blood from the heart. This results in the significant increase in aortic pressure illustrated at approximately 10.1s in Fig. 2.2a. Systolic blood pressure is defined as the peak aortic pressure, P_s, observed during systole.

In a similar manner to the filling phase, blood ejection initially occurs at a high flow rate, which decreases over time. Eventually, ventricular pressure falls below that of the distended aorta and pulmonary artery. At this point, reversal of the pressure gradient causes a small amount of blood to flow back into the ventricles as the semilunar valves snap shut, marking the start of diastole. The point of reversal and the resulting momentary rise in aortic pressure are characterised by a notch in the waveform known as the dicrotic notch. Diastolic blood pressure is defined as the minimum aortic pressure, P_d, observed during diastole. Typical values of P_s and P_d for a healthy adult are <120 mmHg and <80 mmHg, respectively.

The volume of blood ejected from the ventricle during systole is referred to as the stroke volume (SV). Stroke volume is defined as the difference between the end-diastolic volume (EDV) and the end-systolic volume (ESV), as illustrated in Fig. 2.2b. It therefore represents the absolute change in ventricular volume over a single cardiac cycle.

2.1.2 Ejection Fraction

While stroke volume provides a direct measure of ventricular output, it does not account for differences in ventricular size. For this reason, systolic function is more commonly assessed using the ejection fraction (EF), which normalises stroke volume by the end-diastolic volume. Ejection fraction is defined as

$$\mathrm{EF} = \frac{\mathrm{SV}}{\mathrm{EDV}} = \frac{\mathrm{EDV} - \mathrm{ESV}}{\mathrm{EDV}}, \tag{2.1}$$

and is typically expressed as a percentage. As EF captures the proportion of blood ejected relative to ventricular filling, it is widely used in clinical practice and is a key functional metric assessed in echocardiography. When two orthogonal views are available, EF can be estimated using the biplane method of disks. An example illustrating this calculation is shown in Fig. 2.3.

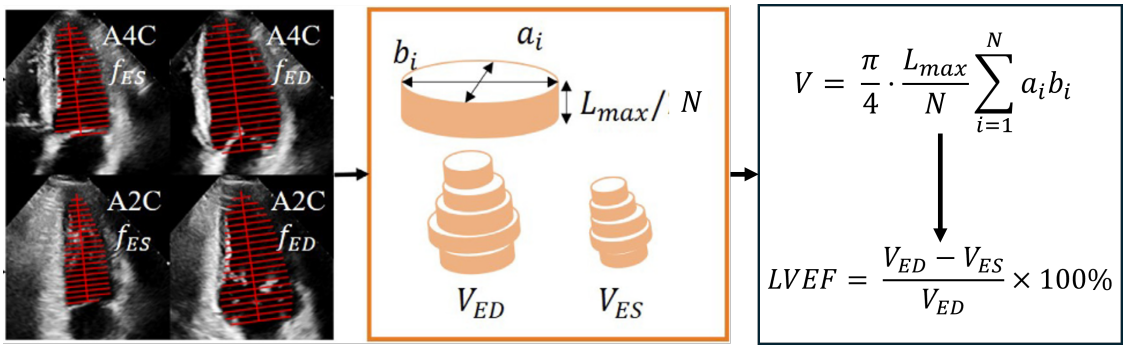


Figure 2.3: LVEF Calculation Using Two Orthogonal Views. Left ventricular ejection fraction (LVEF) is estimated using the biplane method of disks, also known as Simpson's biplane method. Left ventricular widths measured in two orthogonal views are used to estimate the major and minor diameters of an elliptical cross-section. The length of the longest axis extending from the left ventricular apex to the mitral valve plane is computed and divided into N sections, where N denotes the number of disks. The volume of each disk is calculated by multiplying the area of the corresponding ellipse by the disk thickness, given by L_{max}/N. The total left ventricular volume is obtained by summing the volumes of all disks. This procedure is performed for both the end-diastolic and end-systolic frames, and the resulting volumes are used to compute the LVEF. Adapted from [16].

2.2 Cardiac Imaging

2.2.1 Ultrasound

Ultrasound is a mechanical pressure wave and its frequency is measured in cycles per second (Hz). The human audible range is 20Hz - 20kHz, and so any sound waves with a frequency above this range is known as ultrasound. For medical purposes, the waves usually have a frequency larger than 2MHz.

The speed of sound in a medium is dependent on the medium's density and a physical property known as the acoustic impedance. The acoustic impedance describes how much resistance a sound wave faces as it passes through a medium. For example, through air, soft-tissue, and bone - sound travels at around 330m/s, 1540m/s, and 4080m/s, respectively [17, 18]. A medium's acoustic impedance (Z), density (ρ) and the speed of sound (c) through it are related by

$$Z = \rho \times c \tag{2.2}$$

This equation relates the sound wave with a given medium and is useful in generating medical images. Another phenomenon is a wave's response to boundaries. Figure 2.4

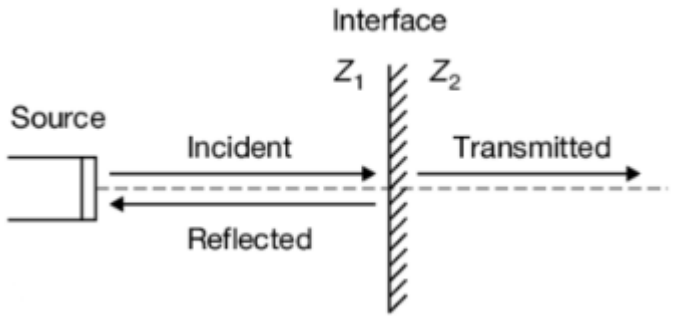


Figure 2.4: Wave Behaviour at a Boundary. A wave incident on a boundary between media of different acoustic impedance is partially reflected and partially transmitted. Image from [19].

shows that upon reaching a boundary between media of different acoustic impedance, sound waves are either transmitted or reflected. In practise, some absorption also occurs but at the intensities typically used, this is negligible compared to how much is reflected and only becomes an issue if the tissue is exposed to ultrasound waves for extended periods. The reflected fraction carries the signal/information needed to generate the ultrasound image. Its intensity can be quantified as follows

$$R = \frac{I_r}{I_i} = \left(\frac{Z_2 - Z_1}{Z_2 + Z_1}\right)^2 \tag{2.3}$$

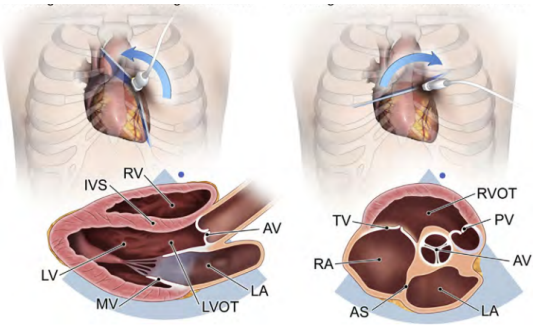
where I_i and I_r represent the intensity of the incident and reflected wave, and Z_1 and Z_2 represent the acoustic impedance of the former and latter medium, respectively. Equation (2.3) shows that a greater difference in acoustic impedance leads to stronger reflection, meaning boundaries between media with significantly differing Z would appear brighter than their surroundings on an ultrasound image.

Turning sound waves into an image is achieved through transducers. Transducers use the forward and reverse piezoelectric effect to turn an electrical current into a mechanical wave and vice-versa. Transducers contain an array of crystals (most transducers used in ultrasound applications are Lead Zirconate Titanium crystals [20]) that vibrate when an electric current is applied to them. Conversely, when stress is applied to the crystal, such as by sound waves that have been reflected, a current is generated. By estimating the speed of Sound in the material using Eq.(2.2), measuring the time taken for the signal to return, and its magnitude, useful illustrations can be developed. These signals can be visualised in different ways to highlight the regions of interest. As a result,

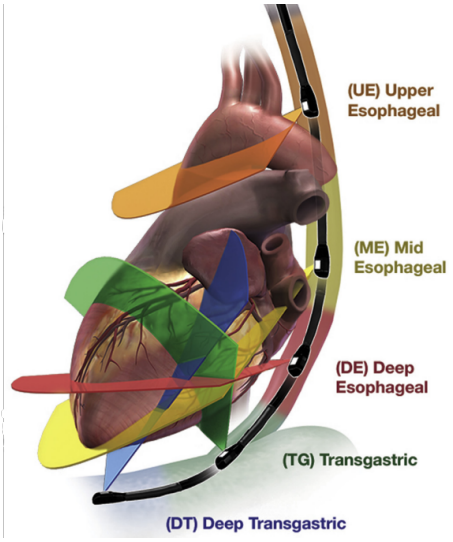
different forms of ultrasound exist, termed modes. A-mode is perhaps the most basic as it uses amplitude modulation to produce a 1D time series of the reflected signal amplitude. On such plots, the timing of the peaks provides information that is used to calculate where boundaries between two media with significantly different acoustic impedance lie in the body. B-mode can be thought of as many A-mode outputs placed side by side. It forms a 2D image and is the most commonly used form of ultrasound in clinics [21, 22]. Furthermore, 3D ultrasound images can be generated by taking several 2D B-mode images and stitching them together. Movement mode, more commonly known as M-mode, allows the observation of moving interfaces along a selected A-mode line over time (or equivalently, across consecutive B-mode images). Through M-mode, physiological variables like heart rate could be measured if one were to monitor, say, the motion of the mitral valve. Other ultrasound modes such as Colour-, Power- and Spectral- Doppler exist which use the Doppler effect to determine the speed, volume and direction of blood within vessels [22].

2.2.2 Echocardiography

When ultrasound is used to investigate the heart and its nearby vessels, the image produced is known as an echocardiogram. There are two main types of echocardiography: transthoracic and transesophageal, both of which are illustrated in Fig. 2.5a and Fig. 2.5b, respectively. Performing TTE is a relatively easy and non-invasive procedure. A technician spreads gel on the probe to better match the acoustic impedance of the skin, and hence prevent large amounts of reflection at the skin. Next, the probe is pressed firmly onto the patient's skin, within a few standard windows to avoid reflection by the ribs, and the ultrasound machine converts the reflected sound waves into images. TTE is the more commonly used form of echocardiography. Performing TEE is a more involved process. The probe is inserted into the patient's throat and guided down the esophagus until it reaches the desired position relative to the heart. The probe sends ultrasound waves and the reflected signals are processed by a computer and converted to images, just like in other forms of ultrasound. Although TEE is more inconvenient for the patient, its proximity to the heart and the absence of obstacles allow it to provide



(a) Probe positions and example visible regions during TTE. Here is shown views in the long- and short- axes of the heart. Image from [23].



(b) Probe positions and visible regions during TEE. Here is shown views that can be acquired at different esophageal depths. Image from [24].

Figure 2.5: The Two Main Types of Echocardiography. a) Transthoracic, and b) Transesophageal.

superior visualisation. The costs of this superior visualisation include: performing an invasive procedure, having to medicate the patient in order to numb their throat and help them relax, significant pressure to get the procedure right first time as performing a second procedure would be very inconvenient, and finally, less intuitive probe control as the transducer is far away from the manipulator.

Standard Transesophageal Views

According to Hahn et al. [25], there are 28 views, shown in Fig. 2.6, in a comprehensive TEE procedure.

Applying computer vision methods to echo images is made more difficult by the large number of unique standard views. Although these views are unique, several views are similar in appearance and many features appear in multiple views, making them difficult to classify. Additionally, an invasive procedure like TEE is avoided as much as possible, therefore there is less data available for training networks with, compared with TTE for example.

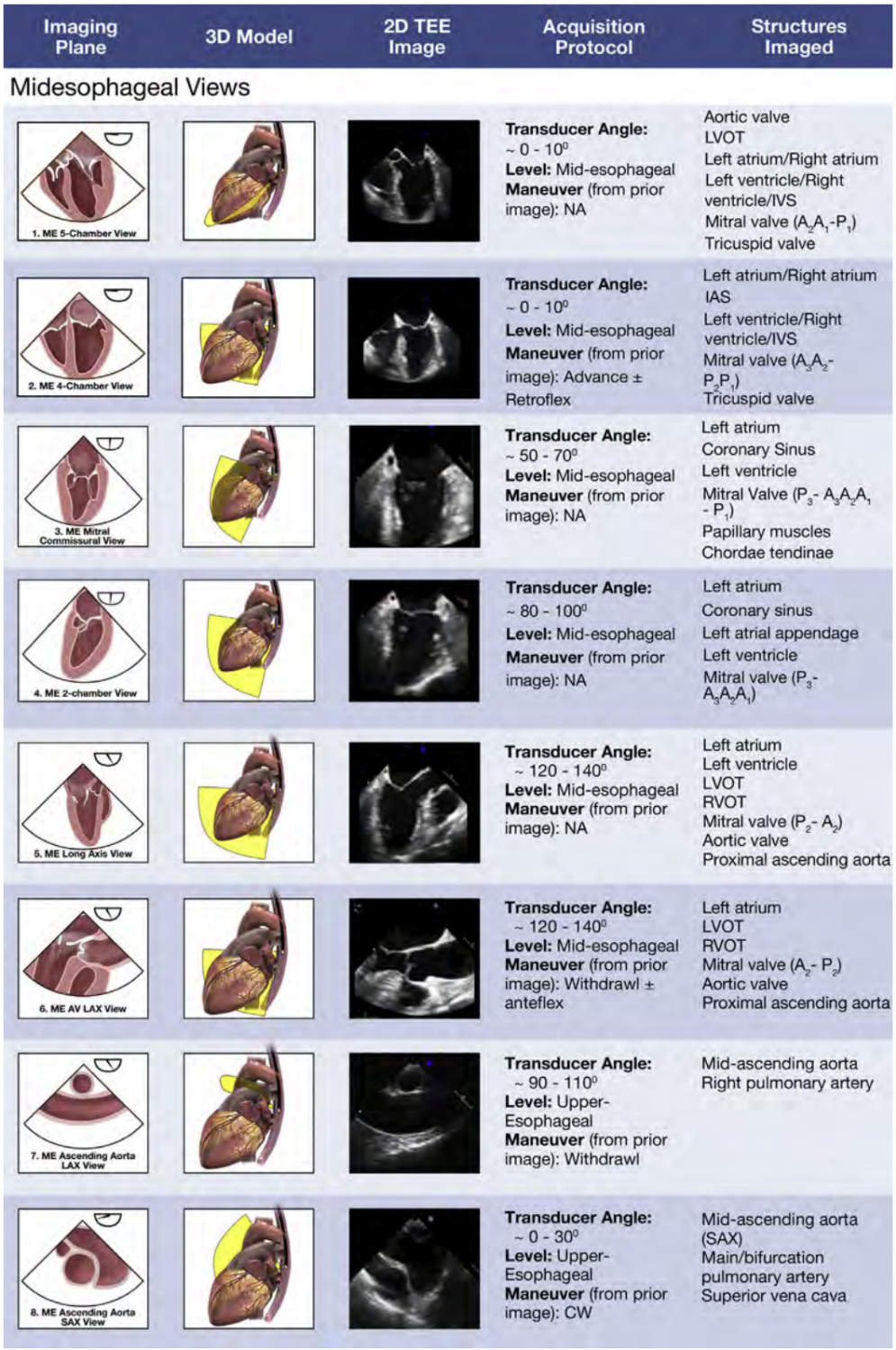


Figure 2.6: Transesophageal Echocardiography Views 1 - 8. A complete list of all the views in a comprehensive TEE Examination. *Imaging plane* shows an idealised representation of each view, *3D Model* shows typical probe position and plane of view, *2D TEE Image* shows an echo of each view, *Acquisition Protocol* details the typical set of actions required to acquire each view, and *Structures Images* contains a list of the structures that should be visible in each view. Image from Hahn et al. [25].

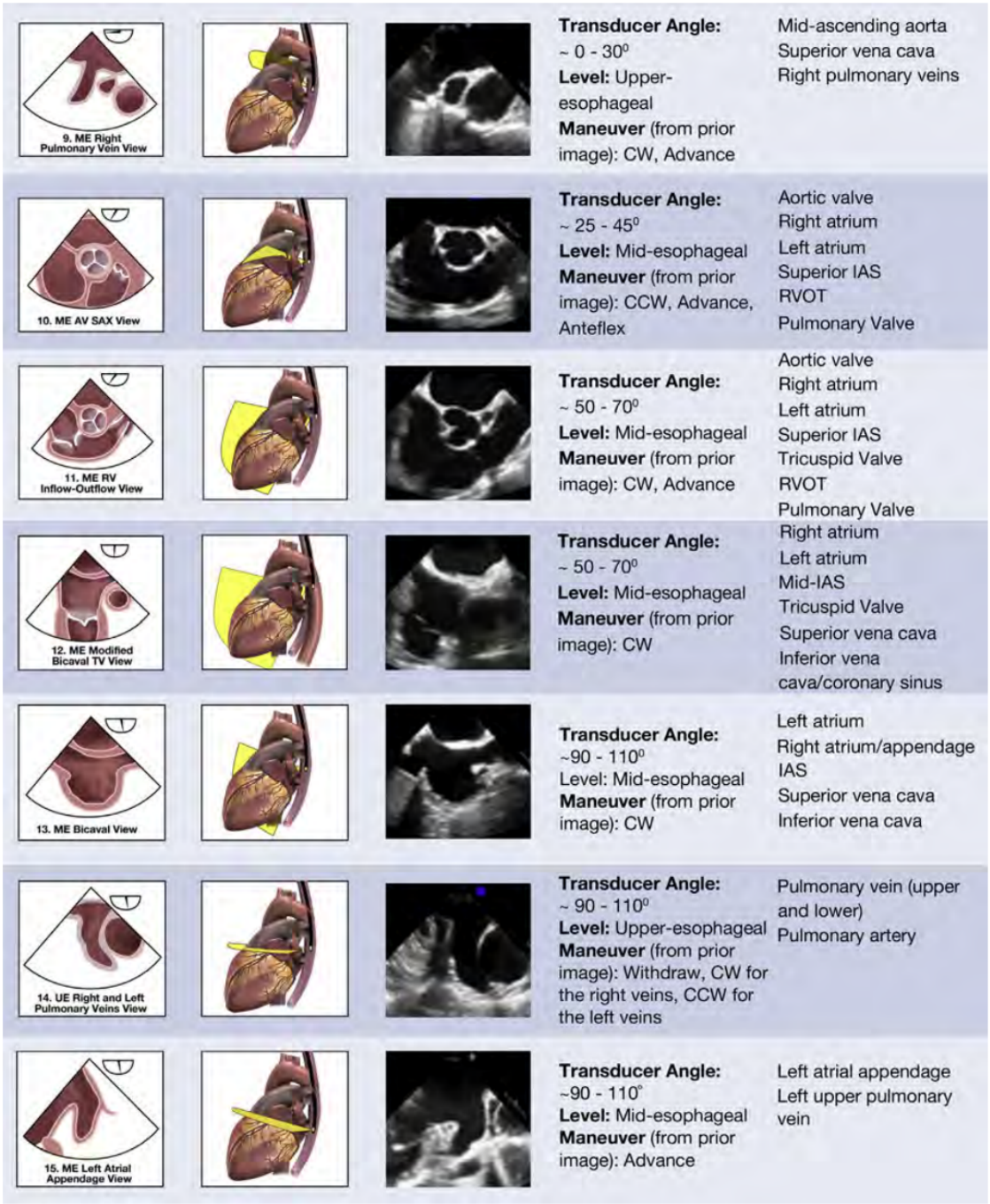


Figure 2.6: Transesophageal Echocardiography Views 9 - 15. (continued).

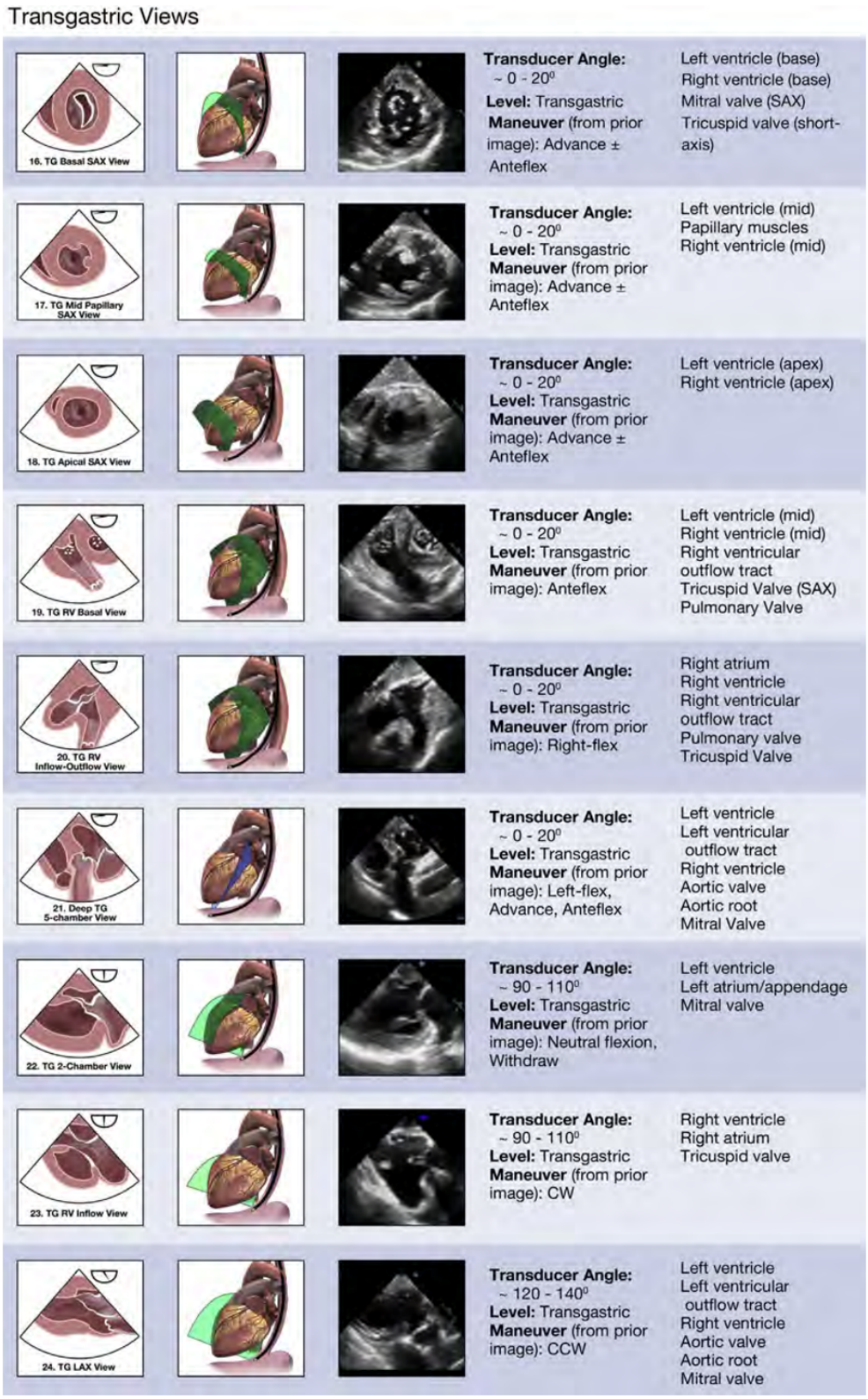


Figure 2.6: Transesophageal Echocardiography Views 16 - 24. (continued).

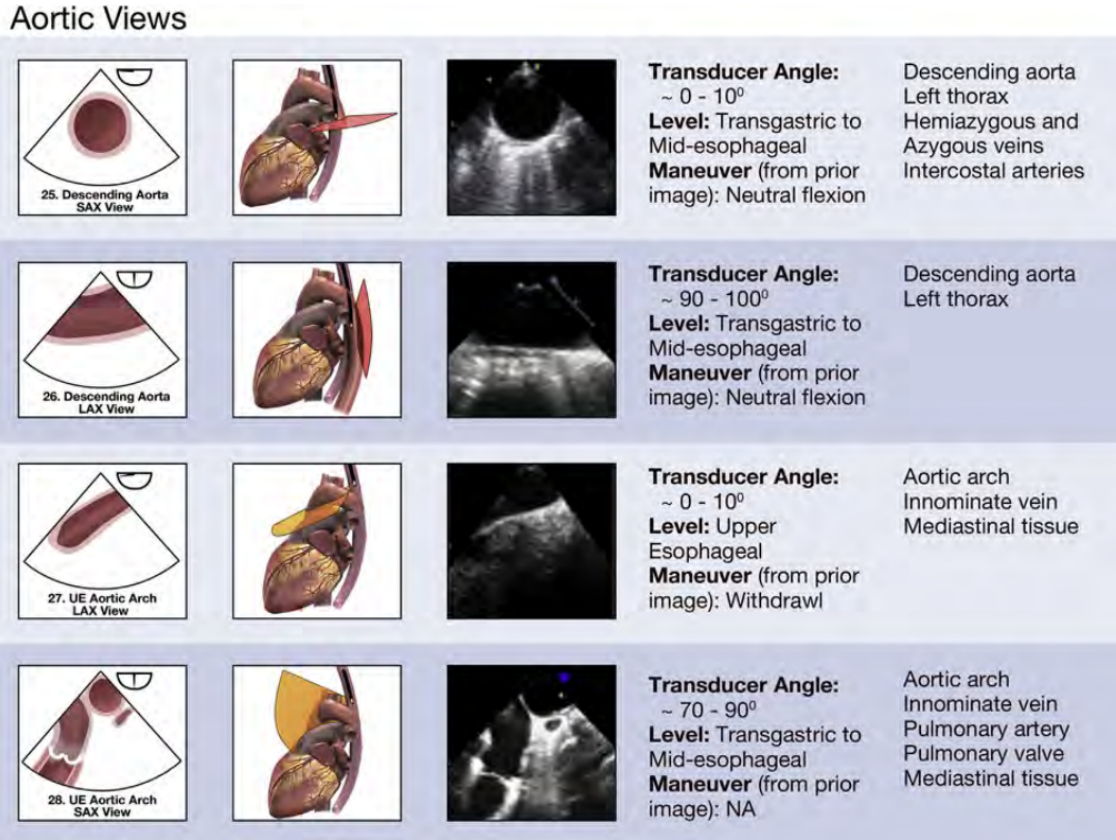


Figure 2.6: Transesophageal Echocardiography Views 25 - 28. (continued).

2.3 Deep Learning in Medical Image Analysis

Historically, medical images were primarily analysed by medical professionals e.g. radiologists. Recently, there has been an increase in the use of computational methods, motivated by improvements in technology, advancement in computer vision techniques (including deep learning methods), availability of powerful computers (such as CPU and GPU technologies), and accessibility to larger datasets. Many computer vision techniques are built upon machine learning methods. Machine learning is a subfield of artificial intelligence that aims to train machines to make decisions or predictions without being explicitly programmed to do so [26]. Example methods include artificial neural networks and random forests which are commonly used in classification and regression tasks. Deep learning is a subfield of machine learning where artificial neural networks with three or more layers are used i.e., the artificial neural network contains more than one intermediate layer between the input and output layer. This structure

is modelled after the brain's own network of neurons and is particularly suitable for analysing medical images because of its inherent ability to work with large datasets to identify important features.

Deep Learning

Conventionally, meaningful or task-related features were mostly designed by human experts, making it difficult for non-experts to apply machine learning methods in their own field. Deep learning has allowed the feature engineering step to become a part of the learning process and thus now, the identification of the most informative aspects of the data is performed by computers rather than humans, including in medical image analysis. Networks like Convolutional Neural Networks (CNNs), an example of which is shown in Fig. 2.7, naturally identify relevant features in the input data during training. CNNs are a unique type of neural networks that use a mathematical operation known as convolution for analysing matrices, giving them a distinctive ability to process pixel data that are common in image processing and image recognition problems. They are designed under the assumption that the input image possesses a structure, therefore neighbouring pixels are likely to contain related information. By repeatedly convolving and downsampling an image, hierarchical feature representations can be learned. The idea of stacking such local feature detectors into a hierarchy that is robust to shifts in position dates back to the Neocognitron [27], an early CNN-like architecture that predates modern convolutional networks. Moreover, the kernels which are responsible for feature extraction are a trained part of the network, therefore the network automatically learns which features are most useful through the examples provided. Its ability to do this solely from data has enabled great success in many applications [28–35]. Since its initial use, many flavours incorporating new innovations have been developed. From LeNet [36] which, building on earlier precursors such as the Neocognitron, is widely regarded as among the first CNNs trained end-to-end with backpropagation, through AlexNet [37], GoogLeNet [31], and VGG-Net [38] which rose to fame due to their then cutting-edge performance on the ImageNet dataset [12], to the latest versions like UNet [39], ResNet [40] and DenseNet [41] which significantly alleviated issues caused by poor

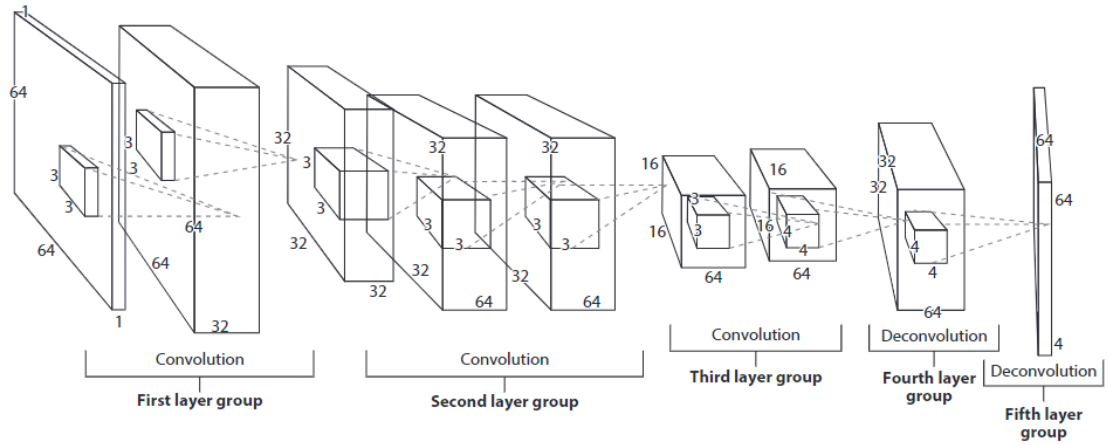


Figure 2.7: An Example CNN. This CNN has 3 encoding layers and 2 decoding layers. In each encoding layer, many 3×3 kernels move around the feature maps extracting features. As the input travels deeper into the network, it is encoded into a latent space. Furthermore, the feature maps are downsampled, hence the local receptive field of each kernel increases. Moving from the 3rd to the 4th layer represents an entrance into the decoding section where the feature maps are upsampled. A CNN of this structure is well-suited for tasks like semantic segmentation. Through training of the kernels, the encoding section would figure out 'what' is present in the image, and the decoding section determines 'where'. Image from [57].

.

gradient flow through existing architectures. Figure 2.8 shows high-level illustrations of the ResNet and DenseNet architectures. These significant improvements in the field of computer vision have provided useful tools for medical image analysis. Such tools have been successfully used for image segmentation [42, 43], image registration [44], image fusion [45], image annotation [46], video generation [47, 48], computer-aided-diagnosis and prognosis [49–51], lesion/landmark detection [52–54], and microscopic imaging analysis [55, 56], amongst others.

More recently, transformer architectures [60] have become a popular choice in computer vision despite the fact they were originally developed for natural language. Dosovitskiy et al. [59] created the Vision Transformer, shown in Fig. 2.9 by adapting the original transformer architecture to process image patches instead of text. The self-attention and cross-attention mechanisms used by these architectures are illustrated in Fig. 2.10.

Transformers are now used for the same tasks CNNs used to dominate and sometimes even outperform them in certain tasks e.g. image segmentation [61], landmark

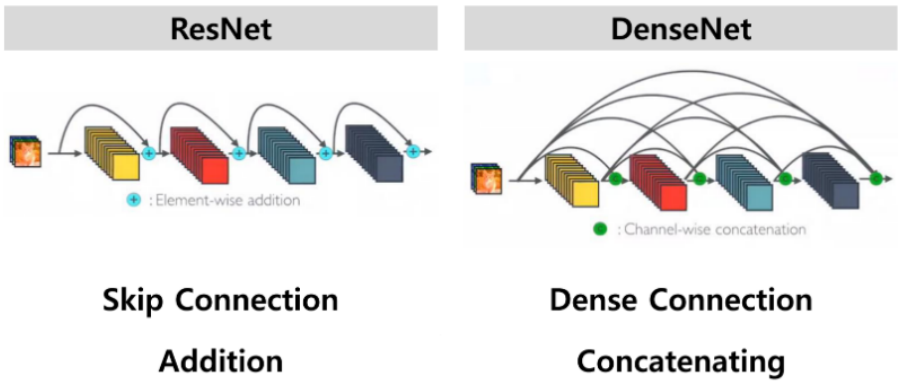


Figure 2.8: ResNet vs DenseNet. High-level illustrations of how the ResNet and DenseNet tackle issues relating to the flow of gradients through a network. The ResNet uses addition to connect all preceding features, whilst DenseNet concatenates them. Image from [58].

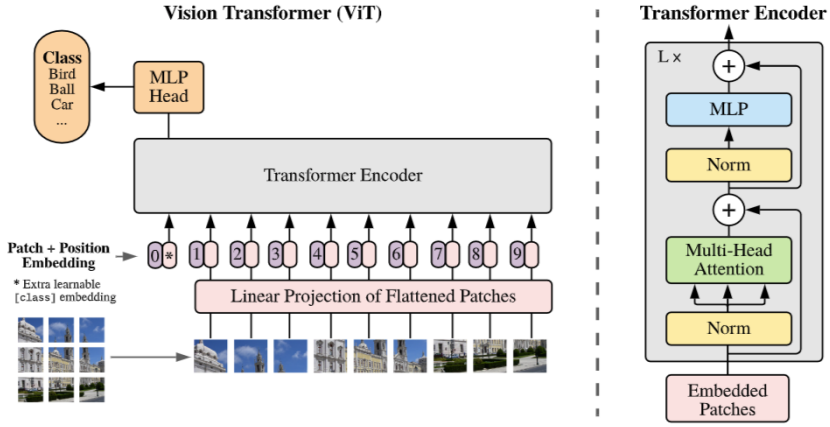
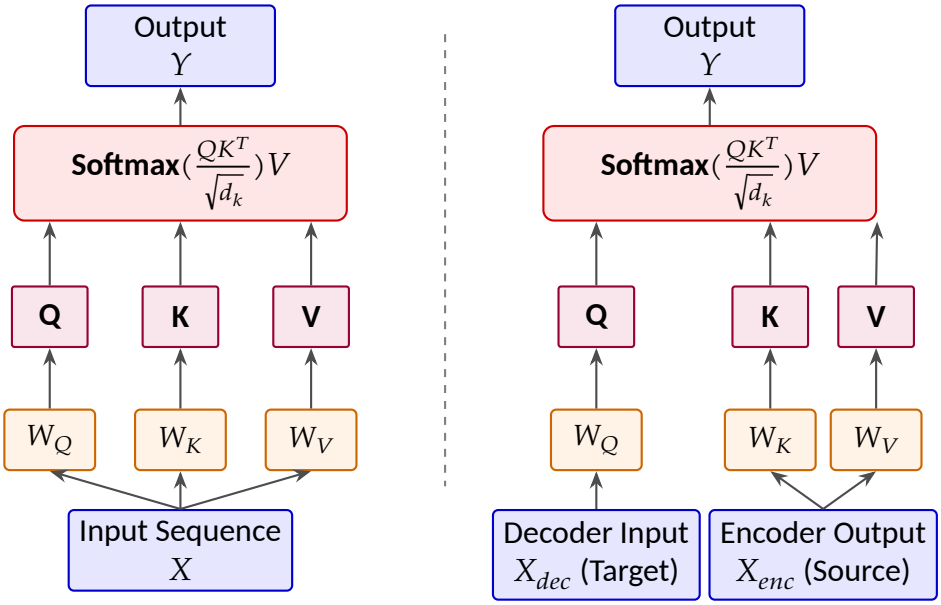


Figure 2.9: An Example Transformer. The Vision Transformer splits the input image into patches, flattens them, linearly embeds them, and adds positional embeddings so the model retains information about each patch's spatial location. The resulting vectors are passed through a series of encoder layers. Within each encoder, features are first normalised and then processed by a self-attention module, which learns relationships between different image patches as training progresses. A skip connection is then applied, encouraging stable gradient flow and preserving information across layers, before a feed-forward network further transforms the features. Image from [59].

detection [62, 63], and video generation [64, 65], amongst others. However, transformer architectures generally require substantially more training data to reach competitive performance, especially when trained from scratch. Unlike CNNs, transformers adapted to vision tasks lack strong inductive biases such as locality and translational invariance in the case of tasks like detection, and equivariance in the case of tasks such as segmentation. Not having these biases is especially harmful when training on small datasets, a common setting in medical imaging [66].



(a) Self-Attention: Q, K, V from same source. If the *query* is a patch representing a 'dog's ear', the *keys* identify related regions within the same image to establish context, and the *values* provide the visual details from those regions to ensure the ear is consistent with the surrounding anatomy.

(b) Cross-Attention: Q from target, K,V from source X_{enc} denotes the encoder output tokens that provide conditioning information, while X_{dec} denotes the decoder state at the current layer, which is updated via cross-attention

Figure 2.10: Self- and Cross-Attention Mechanisms. Self-attention learns relationships within a single input by deriving queries (Q), keys (K), and values (V) from the same feature representation, allowing contextual reasoning across elements of the input e.g. patches at different positions in an image or words at different parts of a sentence. In contrast, cross-attention allows information from an external source to influence the target representation by computing queries from the target features and keys and values from a separate input. This mechanism enables conditioning on other data, such as semantic masks, text, or features from other imaging modalities.

Deep learning methods excel particularly when large amounts of high-quality data is available. This is a luxury that is affordable in some cases [12] but is scarce in medical applications. Networks trained on small datasets are prone to overfitting and hence lack generalisability, a desirable quality of any network that is intended for real-world use. Consequently, successfully applying deep learning methods to medical applications where data is not plentiful is a significant challenge. Many strategies have been developed in the literature to tackle this with varying levels of success. Some researchers take an approach commonly known as transfer learning, where models that have been trained for an arbitrary computer vision task are used as a starting point [67, 68]. The network

would then be fine-tuned for the user's purpose by only training the final layer or just re-training the whole network but with a very small learning rate. This approach especially has gained much popularity as it makes the use of powerful and very deep networks more accessible to those who do not have the required computational resources or data. Next, other researchers used patches of an image as inputs because this allowed them to work with smaller networks [69, 70]. Smaller networks have fewer parameters and hence are less likely to overfit. Finally, some researchers generated 'new' instances of data by applying data augmentations to existing samples, thus expanding their dataset [70, 71].

Summary

Deep learning has been a revolutionary development and its contribution to the field of medical imaging has been profound. However, as has been shown, it is not without limitation. Aside from data requirements, its consistent and effective use are often very difficult to achieve as a result of lack of generalisability in many problems. Furthermore, there is the issue of sustainability and how the computationally intensive development and use of these methods affect our environment [72]. Nonetheless, deep learning has shown itself as an incredibly versatile and useful tool that will, arguably, only continue to improve.

2.4 Generative Modelling

Recently, augmenting pre-existing datasets with examples that have been synthesised through data generation methods has gained popularity. These methods generally fall into two categories namely, same- and cross- domain synthesis. The former involves creating data in the same domain as the source data, and the latter involves using data from a source domain to create data in a target domain. Many of these methods rely on generative adversarial networks (GANs) [73], especially the CycleGAN [74], shown in Fig. 2.11. For example, Tang et al. [75] used a conditional GAN to synthesise X-ray images and Chartsias et al. [76] synthesised pairs of MRI and CT images using a CycleGAN [74]. Echo images are considerably more difficult to synthesise due to the complex noise patterns. Attempts to synthesise echo images have traditionally taken a modelling

approach where the physics behind the observed noise patterns are simulated [77]. Unfortunately, this approach often lacks realism and has poor scalability due to the large computational power needed to run the simulations. Abdi et al. [78] successfully took a synthetic augmentation approach to synthesise echo images by training a patch-based cGAN, but did so using a paired, scarcely annotated dataset.

In its most basic form, a GAN is made up of two networks, a generator and a discriminator. The generator's task is to generate new, believable instances of the data, whereas the discriminator's task is to identify which data originated from the generator and which is part of the real dataset. The generator is usually a decoder or full encoder-decoder, depending on whether it is conditioned on an input or not. The discriminator is a classifier. By penalising the discriminator each time it is fooled by the generator i.e., labels a fake image as real, and penalising the generator each time its output is correctly labelled by the discriminator, the result is the discriminator and generator forcing each other to improve at their respective tasks. The loss function for the joint training of these networks is

$$\mathcal{L}_{adv}(G, D, X, Z) = \mathbb{E}_{\mathrm{x}\sim p_{data}(\mathrm{x})}[\log D(\mathrm{x})] + \mathbb{E}_{\mathrm{z}\sim p_z(\mathrm{z})}[\log(1 - D(G(\mathrm{z})))] \quad (2.4)$$

where x is a datum originating from the real dataset and z is the input to the GAN. Assuming successful training and hence convergence, the generator should be able to produce believable, new instances of the real data. The CycleGAN builds upon this idea. By combining multiple discriminators $\{D_x, D_y\}$, generators $\{G, F\}$, and the addition of a cycle-consistency loss term, it allows the rendering of images in a target domain (Y) from unpaired data in a source domain (X). The CycleGAN thus allows the creation of 'new' images that can be used to augment existing datasets.

2.4.1 Contrastive Unpaired Translation

Since the CycleGAN's introduction, other variants have been developed [79, 80]. Notably, there exists a variant known as the contrastive unpaired translation (CUT) method [81]. Whether CycleGAN or CUT is a better method depends on the applications but both have been significantly useful in solving the issue of a lack of high-quality labelled data.

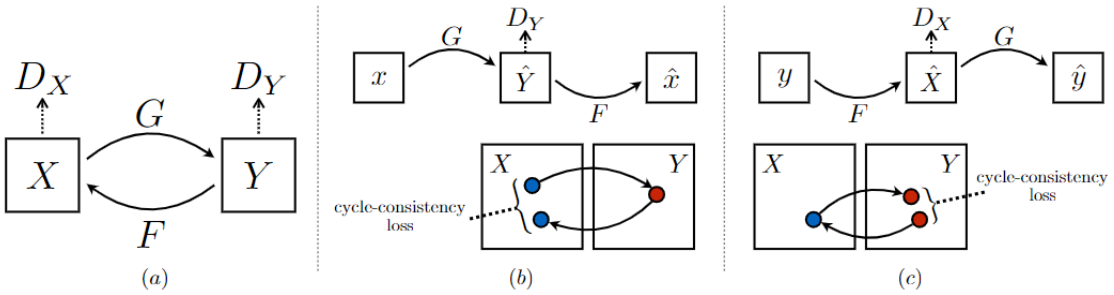


Figure 2.11: CycleGAN Architecture. a) The model contains two generators responsible for generating images in the other domain, and two discriminators for assessing the quality of the generated images in each domain, b) The original image in X should be retrievable from the projection in domain Y and generator F, c) The original image in Y should be retrievable from the projection in domain X and generator G. Image from [74].

CUT leverages contrastive learning [82] to encourage content preservation in unpaired image-to-image translation problems. In the CycleGAN, the content is preserved using the cycle-consistency loss; however, this is too restrictive in some cases. CUT, instead, samples patches from the original and generated images, then in latent spaces, maps patches in the same position together and distances patches from different positions. Patches in the same position are known as positive keys, and patches from differing positions are negative keys. The association and dissociation of positive and negative keys is achieved by taking inspiration from the contrastive cross-entropy loss

$$\ell(\mathrm{v}, \mathrm{v}^+, \mathrm{v}^-) = -\log\left[\frac{\exp(\mathrm{v}\cdot\mathrm{v}^+/\tau)}{\exp(\mathrm{v}\cdot\mathrm{v}^+/\tau) + \sum_{n=1}^{N}\exp(\mathrm{v}\cdot\mathrm{v}^-/\tau)}\right], \tag{2.5}$$

where v, v^+, and v^- are the selected patch, its positive(s) and its negative(s), respectively. A high level representation of the process is shown in Fig. 2.12.

Given a dataset of unpaired instances $X = \{x \in X\}$, $Y = \{y \in Y\}$, a generator G is tasked with creating images in target domain Y using images from source domain X. The generator is split into an encoder and a decoder, and hence, a synthetic image in Y is generated using $\hat{y} = G(\mathrm{x}) = G_{dec}(G_{enc}(\mathrm{x}))$. L layers of interest are then selected and the feature maps at each layer are passed through a two-layer, non-linear, multi-layer perceptron H_l. A feature stack can then be generated using $\{z_l\}_L = \{H_l(G^l_{enc}(\mathrm{x}))\}_L$ where G^l_{enc} is the generator output of the l-th chosen layer. Similarly, a feature stack is generated from $\hat{y}$ as follows $\{\hat{z}_l\}_L = \{H_l(G^l_{enc}(G(\mathrm{x})))\}_L$. From each layer $l \in \{1, 2, ..., L\}$, S_l patches are extracted. A single patch is represented as $z^s_l \in R^{C_l}$ where $s \in \{1, \dots, S_l\}$

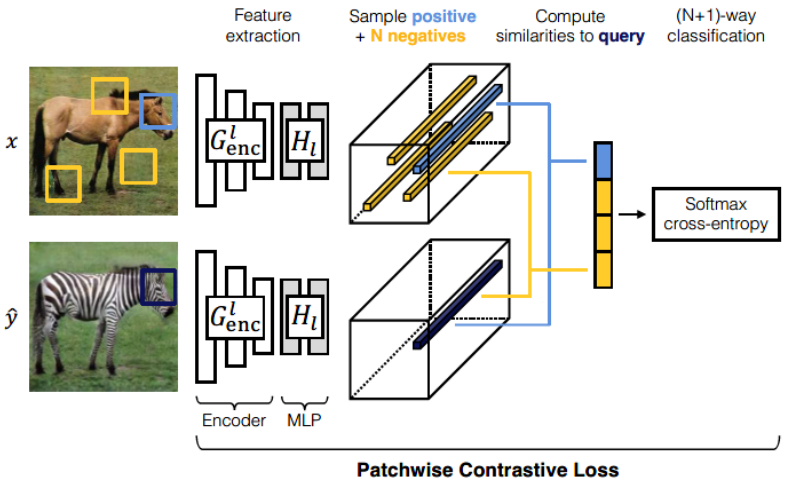


Figure 2.12: CUT Process. The original and synthetic image representations are compared at different layers within the encoding section of the generator. By ensuring that the representation of corresponding patches i.e., positive pairs, are encoded in a similar way, the content is preserved spatially but appearance is allowed to vary. The representations are first transformed by a learnable Multi-Layer Perceptron H_l before being passed to the loss term as this has been shown to improve performance. Image from [81].

and C_l is the number of channels at the l-th layer of interest. Furthermore, the other patches are written as $z_l^{S\setminus s} \in R^{(S_l-1)\times C_l}$. Patches in identical positions in both z_l^S and $\hat{z}_l^S$ are used as positive keys in the contrastive loss, and patches from different positions in the same image are used as negative keys. This results in a contrastive loss variant known as PatchNCE loss.

$$\mathcal{L}_{PatchNCE}(G,H,X) = \sum_{l=1}^{L}\sum_{s=1}^{s=S_l} \ell(\hat{z}_l^s, z_l^s, z_l^{S\setminus s}). \tag{2.6}$$

However, simply using this loss term is insufficient as the best mapping to minimise this loss alone is an identity mapping. Therefore, this loss is used in combination with the adversarial loss term shown in Eq. (2.4). The full loss function is thus

$$\mathcal{L}(G,H,D) = \mathcal{L}_{adv}(G,D,X,Y) + \lambda_x \mathcal{L}_{PatchNCE}(G,H,X) + \lambda_y \mathcal{L}_{PatchNCE}(G,H,Y), \tag{2.7}$$

where λ_x and λ_y are hyperparameters that weight their corresponding loss term relative to the others.

2.4.2 Diffusion Models

Diffusion models are a class of generative models that progressively corrupt data by adding noise in a *forward process*, and then learn to reverse this corruption in a *reverse process* [83, 84]. In the forward direction, a sample x_0 from the data distribution q_{data} is gradually transformed into increasingly noisy versions $x_1, x_2, \dots, x_T$ by a fixed Markov chain where each step adds Gaussian noise. The noise is added according to a fixed variance schedule $\{\beta_t \in (0,1)\}_{t=1}^{T}$, allowing the forward process to be written as follows

$$q(x_t \mid x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \mathrm{I}). \tag{2.8}$$

As this process follows a Markov chain, the sample at each timestep is only dependent on the sample at the previous timestep, therefore $q(x_{1:T} \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1})$. Letting $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ the noised data x_t can be sampled at any timestep using the equation

$$q(x_t \mid x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathrm{I}). \tag{2.9}$$

Using the reparameterisation trick, Eq. (2.9) can be written as

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \quad \epsilon \sim \mathcal{N}(0, \mathrm{I}), \tag{2.10}$$

In principle, the reverse process would involve sampling from $q(x_{t-1} \mid x_t)$, which can be written as

$$q(x_{t-1} \mid x_t) = \frac{q(x_t \mid x_{t-1})q(x_{t-1})}{q(x_t)}, \tag{2.11}$$

using Bayes Rule. $q(x_t \mid x_{t-1})$ is known from Eq. 2.8, and the marginals can be expressed as

$$q(x_{t-1}) = \int q(x_{t-1} \mid x_{t-2})q(x_{t-2})dx_{t-2}, \quad q(x_t) = \int q(x_t \mid x_{t-1})q(x_{t-1})dx_{t-1}. \tag{2.12}$$

Calculating these marginals involves integrating over all possible earlier states which is an intractable integral. However, the reverse process is tractable when conditioned on x_0. Using Bayes Rule, it can be shown that

$$q(x_{t-1} \mid x_t, x_0) = \frac{q(x_t \mid x_{t-1}, x_0)q(x_{t-1} \mid x_0)}{q(x_t \mid x_0)}. \tag{2.13}$$

Noting that a key property of a Markov chain is that the current state x_t is only dependent on the previous state x_{t-1}, we see that $q(x_t, \mid x_{t-1}, x_0) = q(x_t, \mid x_{t-1})$. Using the reparameterisation trick shown in (2.10) to substitute Eq.(2.9) into Eq.(2.13) and simplifying, $q(x_{t-1} \mid x_t, x_0)$ is revealed to be a Gaussian with mean and variance:

$$\tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\,\beta_t}{1-\bar{\alpha}_t}\,x_0 \;+\; \frac{\sqrt{\alpha_t}\,(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\,x_t, \quad \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. \tag{2.14}$$

The mean of this posterior is dependent on x_0 which is not available in practice during the reverse step. To tackle this, Ho et al. [84] approximate the posterior parameters $\tilde{\mu}_t$ and $\tilde{\beta}_t$ using a learnable function $p_\theta(x_{t-1} \mid x_t) \sim \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 \mathrm{I})$. To bridge the gap between the intractable true posterior and this parameterised form, they express the network's mean in terms of a noise-prediction network $\epsilon_\theta(x_t, t)$ and fix or learn simple variance schedules σ_t^2. Consequently,

$$\mu_\theta(x_t, t) = \tilde{\mu}\Big(x_t, \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta)\Big) = \frac{1}{\sqrt{\alpha_t}}\Big(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\Big), \quad \sigma_t^2 = \tilde{\beta}_t, \tag{2.15}$$

thus converting generative sampling into iteratively denoising x_t by predicting and removing the added noise. The network $\epsilon_\theta(x_t, t)$ is trained by optimising the variational bound on the negative log likelihood which corresponds to minimising the Kullback-Leibler (KL) divergence between the true two-point posterior $q(x_{t-1} \mid x_t, x_0)$ and the approximation $p_\theta(x_{t-1} \mid x_t)$.

$$\mathcal{L}_{\text{VLB}} = \mathsf{KL}(q(x_T \mid x_0) \| p(x_T)) + \sum_{t=2}^{T} \mathsf{KL}(q(x_{t-1} \mid x_t, x_0) \| p_\theta(x_{t-1} \mid x_t)) - \log p_\theta(x_0 \mid x_1). \tag{2.16}$$

Empirically, the dominant learning signal for the network arises from matching intermediate posteriors. In light of this Ho et al. [84] discard lower-impact terms and further simplify to achieve the following loss function

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{t, x_0, \epsilon}[\|\epsilon - \epsilon_\theta(x_t, t)\|^2]. \tag{2.17}$$

Despite its simplicity, this loss function has been shown to achieve performance on par with more complex variational bounds while greatly reducing implementation complexity and improving stability [84, 85]. This diffusion model variant is known as the Denoising Diffusion Probabilistic Model (DDPM).

Training Procedure

In practice, each training iteration consists of drawing a fresh data sample x_0, selecting a random timestep t, corrupting x_0 to obtain x_t, and updating θ via gradient descent on the MSE loss between the true noise and the network's prediction.

Once the model is trained, sampling proceeds by initializing a random tensor $x_T \sim \mathcal{N}(0, I)$ and iteratively applying the learned reverse transformation for $t = T, \dots, 1$:

$$x_{t-1} = \frac{1}{\sqrt{1-\beta_t}}\Big(x_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}\epsilon_\theta(x_t, t)\Big) + \sigma_t z, \quad z \sim \mathcal{N}(0, I), \tag{2.18}$$

with σ_t chosen to match the forward noise variance in the case of DDPM.

Conditioned Generation

A defining feature of diffusion frameworks is their flexibility in conditioning. By augmenting the denoiser network to accept auxiliary inputs c, such as class labels, text embeddings, or spatial masks, it is possible to learn conditional distributions $p_\theta(x_{t-1} \mid x_t, c)$ [84, 86–88]. The training loss naturally extends to

$$\mathbb{E}_{t, x_0, \epsilon, c}[\|\epsilon - \epsilon_\theta(x_t, t \mid c)\|^2], \tag{2.19}$$

allowing precise control over the generated content.

Classifier guidance [89] refines conditioned generation by incorporating gradients from a pretrained classifier $p_\phi(y \mid x_t)$, where y is a class label variable. This effectively biases the denoising path toward a desired class label:

$$\hat{\epsilon}_\theta = \epsilon_\theta(x_t, t) - s\,\nabla_{x_t} \log p_\phi(y \mid x_t), \tag{2.20}$$

where s weights the influence of the classifier gradient.

Alternatively, classifier-free guidance, developed by Ho et al. [90] achieves comparable results without an external classifier by training the model both with and without conditioning and mixing these two modes at sampling time.

$$\hat{\epsilon} = \epsilon_\theta(x_t, t \mid c) + s\Big(\epsilon_\theta(x_t, t \mid c) - \epsilon_\theta(x_t, t \mid \emptyset)\Big). \tag{2.21}$$

This allows flexible control over the generation process, supporting tasks such as class-conditional and text-guided synthesis [86]. However, classifier-based and classifier-free guidance often involve a trade-off: larger guidance weights improve fidelity but reduce diversity, while lower weights preserve variety at the cost of condition adherence.

While classical diffusion models typically rely on $T \approx 1000$ sampling steps, this comes at a steep computational cost. Several approaches have been proposed to address this, with Denoising Diffusion Implicit Models (DDIM) [91] being among the earliest to show that the reverse process can be made deterministic and significantly accelerated—often requiring only 10–50 steps while maintaining high sample fidelity. More recent methods like Elucidated Diffusion Models (EDM) [92] build on this by optimising noise schedules, loss weightings, and solver choices, demonstrating that diffusion models can be both fast and high quality with surprisingly few evaluations.

A more fundamental shift reframes diffusion as a continuous-time stochastic process. In score-based generative modelling [93], the forward process is modelled using a stochastic differential equation (SDE), and the network is trained to estimate the score function $\nabla_x \log p_t(x)$ at arbitrary noise levels. This continuous formulation allows numerical solvers, such as Heun's method or Runge–Kutta schemes, to be used with adaptive integration steps, enabling a trade-off between sampling speed and accuracy. As a result, sampling time can be reduced without a substantial loss in image quality. Related approaches, including consistency models [94] and progressive distillation [95], further move towards single-step sampling, although this typically comes at the cost of additional training complexity or reduced flexibility.

Video Diffusion Models

Video diffusion models [96–99] extend the standard diffusion framework to sequential data, aiming to generate frame sequences that are temporally coherent as well as spatially realistic. Early approaches adapted pretrained 2D diffusion backbones using "inflated" convolutions, thereby repurposing spatial kernels by inserting a temporal axis and appending temporal convolutions [98]. This (2+1)D strategy preserves spatial inductive biases while enabling basic motion modelling. More recent architectures use

temporal self-attention to model long-range dynamics, treating feature maps at each spatial location as sequences over time. These changes allow the model to capture smooth object motion and periodic activity, such as the contraction–relaxation cycles in cardiac imaging.

Latent Diffusion Models

Latent diffusion models (LDMs) [87, 96, 99] address the computational cost of high-resolution synthesis by learning to denoise in a compact latent space. Input data x_0 is first encoded using a pretrained autoencoder into a latent representation $z_0 = E(x_0)$, and the diffusion model is trained to predict noise in latent space, $\epsilon_\theta(z_t, t)$. Generated samples are subsequently decoded back to pixel space using a learned decoder, such that $x_0 \approx D(z_0)$. This formulation leads to substantial memory savings and faster sampling, particularly for high-dimensional data such as 512^2 images or full-resolution video.

However, operating in latent space introduces additional challenges for conditioning. While efficiency is improved, the separation between latent and pixel domains can weaken the enforcement of pixel-level constraints. Conditioning on high-resolution signals, including segmentation masks, spatial priors, or text prompts, therefore requires careful alignment between the latent representation and the original image space. Poor alignment can lead to degraded fine detail or the introduction of artefacts in the generated output.

Diffusion models have demonstrated strong performance across a wide range of tasks, including image super-resolution, video generation, and biomedical image synthesis [14, 100]. In the context of echocardiography, the ability to condition generation on anatomical structure and cardiac phase offers a powerful mechanism for producing realistic and physiologically consistent image sequences.

2.4.3 Flow Matching

Flow matching [101] is a continuous-time generative modelling method that learns a velocity field to transport samples from a simple base distribution to the data distribution

in a deterministic way. Given two distributions $x_0 \sim p_0$ and $x_1 \sim p_1$, where each data point x_0 corresponds to a matching point x_1, flow matching aims to learn a velocity field

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

that defines how a particle at x_t should move as t evolves such that, by integrating this field from $t = 0$ to $t = 1$, x_t is transported from x_0 to x_1. To train the model, a path p_t interpolating between p_0 and p_1 is defined, for example via linear interpolation

$$x_t = (1 - t)x_0 + t\,x_1,$$

and the neural network is trained by regressing its predicted vector field $v(x, t; \theta)$ toward a prescribed target field $u_t(x)$ along this path, where in this case $u_t(x)$ corresponds to the displacement $x_1 - x_0$. The basic flow matching objective is

$$\mathcal{L}_{\mathsf{FM}}(\theta) = \mathbb{E}_{t\sim ll(0,1),\, x\sim p_t(x)}[\| v(x, t; \theta) - u_t(x) \|_2^2].$$

In their original work, Lipman et al. [101] introduced this approach for a general family of Gaussian probability paths and showed that it can scale to high-dimensional generative tasks such as image generation. They further demonstrated that familiar diffusion paths arise as special cases within this framework, and that alternative paths, such as those based on optimal transport, can offer faster sampling and improved generalisation. Flow matching has seen many uses in video generation [102], biomedical image segmentation [103] and echocardiogram generation [14, 104].

2.4.4 Metrics

To evaluate the quality of generated images and videos, a range of metrics are required. These metrics provide objective measures for comparing generated data to reference data and for evaluating different aspects of visual fidelity and realism. Evaluating generated samples can be especially challenging as no single metric fully captures perceptual quality, structural consistency, and temporal coherence. As a result, multiple complementary metrics are commonly used in tandem, each targetting a specific aspect of the generated images. In many cases such as domain translation, there exists no reference data. In

these cases, distributional metrics are relied upon rather than pairwise metrics. In this section, pixel-level, perceptual, and distribution-based metrics used throughout this thesis are described

PSNR

Peak Signal-to-Noise Ratio (PSNR) is a pixel-level fidelity metric commonly used to compare a generated image or video frame against a ground-truth. It is derived from the mean squared error (MSE), which for an image I and its reconstruction $\hat{I}$ of size $H \times W$ is defined as

$$\text{MSE} = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} \left(I(i,j) - \hat{I}(i,j) \right)^2 . \tag{2.22}$$

PSNR is then computed as

$$\text{PSNR} = 10 \log_{10} \left(\frac{L^2}{\text{MSE}} \right), \tag{2.23}$$

where L is the maximum possible pixel value (L = 255 for 8-bit images, for example) PSNR is easy to compute and provides an interpretable measure of pixel-wise agreement, where higher values indicate a better reconstruction. However, its usefulness in image and video generation is limited. Being a pixel-wise metric, PSNR tends to favour overly smooth outputs and correlates poorly with perceived image quality. Furthermore, in generative settings, multiple plausible outputs often exist for a given input. Therefore, PSNR is more a metric to compare equality of images than whether a generated image looks realistic.

SSIM

The Structural Similarity Index Measure (SSIM) [105] is a metric designed to assess image similarity based on structural information rather than raw pixel differences. It decomposes similarity into three components: luminance, contrast, and structure. For two image patches x and y, SSIM is defined as

$$\text{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{2.24}$$

where μ_x and μ_y are the mean intensities, σ_x^2 and σ_y^2 are the variance, σ_{xy} the covariance between the two patches. The constants C_1 and C_2 are included to stabilise the computation. This work uses the Scikit-Image [106] implementation which uses $C_1 = 0.01L^2$ $C_2 = 0.03L^2$ where L is the data range ($L = (255 - 0)$ for 8-bit images, for example)

SSIM values lie in the range $[-1, 1]$ with values closer to 1 indicating greater structural similarity. In practice, SSIM is computed over local windows and averaged across an image. Compared to purely pixel-based measures, SSIM better captures changes in structure, edges, and contrast.

LPIPS

Learned Perceptual Image Patch Similarity (LPIPS) [107] measures the perceptual distance between two images by comparing deep feature representations extracted from a pretrained neural network. Instead of operating directly on pixel intensities, LPIPS computes distances between corresponding feature maps at multiple layers, capturing differences in texture, structure, and higher-level visual semantics. Consequently, it is better aligned with human judgement than pixel-wise metrics. LPIPS is in the range [0,1] and a lower LPIPS value means greater perceptual similarity between a generated image and a ground-truth. This work uses the VGG backbone implementation.

FID

Fréchet Inception Distance (FID) [108] evaluates the quality of a set of generated images by comparing their feature distribution to that of real images. Features are extracted using a pretrained InceptionNet [31] network, and the resulting activations for real and generated samples are each modelled as multivariate Gaussian distributions with means μ_r, μ_g and covariances Σ_r, Σ_g , respectively. The distance between these distributions is computed using the Fréchet distance,

$$\text{FID} = \|\mu_r - \mu_g\|_2^2 + \text{Tr}\Big(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{\frac{1}{2}}\Big). \tag{2.25}$$

Lower FID values indicate closer agreement between the real and generated feature distributions, reflecting both sample quality and diversity at the dataset level. Unlike

pairwise metrics, FID evaluates global distributional similarity rather than individual reconstructions. However, it relies on the choice of feature extractor and the assumption that feature activations are approximately Gaussian, and it is sensitive to the number of samples used to estimate the statistics.

FVD

Fréchet Video Distance (FVD) [109] extends the principles of FID to video generation by comparing distributions of spatiotemporal features extracted from short video clips. These features are obtained from a pretrained Inflated 3D ConvNet, allowing FVD to capture both spatial appearance and temporal dynamics. As with FID, lower FVD values indicate greater similarity between real and generated data distributions.

FVD is designed to assess video quality at the dataset level and is sensitive to temporal consistency, motion realism, and long-range dynamics. However, it shares many of the same limitations as FID, including dependence on the chosen feature extractor and a normality assumptions.

Human Evaluation

While the above metrics provide convenient and reproducible measures of image and video quality, they do not wholly capture human perception of realism, particularly in medical imaging as these metrics are often developed with a bias towards natural images. Consequently, human evaluation is often used as a complementary assessment, for example by asking participants to judge if a generated sample appears real or fake. This type of evaluation provide insight into perceptual realism that may not be shown by quantitative metrics alone, and are commonly used in generative modelling to evaluate visual plausibility. However, human evaluation is subjective and suffers from inter- and intra- observer variability

But He said to me, "My grace is sufficient for you, for my power is made perfect in weakness." Therefore I will boast all the more gladly of my weaknesses, so that the power of Christ may rest upon me.

— 2 Corinthians 12:9

3 Datasets

Contents

3.1 Internal TEE Dataset 37
3.2 Public Datasets 40
3.2.1 SSM Dataset 40
3.2.2 CAMUS 41

In this Chapter, the datasets, pre-processing, and annotation procedures used throughout this DPhil are described. The datasets comprise a combination of private TEE data supplied by GE HealthCare and public TTE datasets. Two expert cardiologists who use echocardiography routinely in clinical practice, with more than 23 and 15 years of experience respectively, were recruited to annotate the TEE data during the DPhil. All annotations were performed using the `V7darwin`[1] annotation software and each annotator reviewed the other annotator's work.

3.1 Internal TEE Dataset

During this DPhil, an internal TEE dataset supplied by GE HealthCare was used. The dataset consisted of DICOM files containing de-identified TEE sequences, acquired

[1] https://www.v7labs.com/darwin

predominantly using the GE VIVID E95 ultrasound system. The sequences depict mid-esophageal four-chamber and mid-esophageal two-chamber views. This dataset is particularly heterogeneous, as the scans were sourced from medical institutions across the Netherlands, Norway, Oceania, Germany, Italy, Spain, and Sweden.

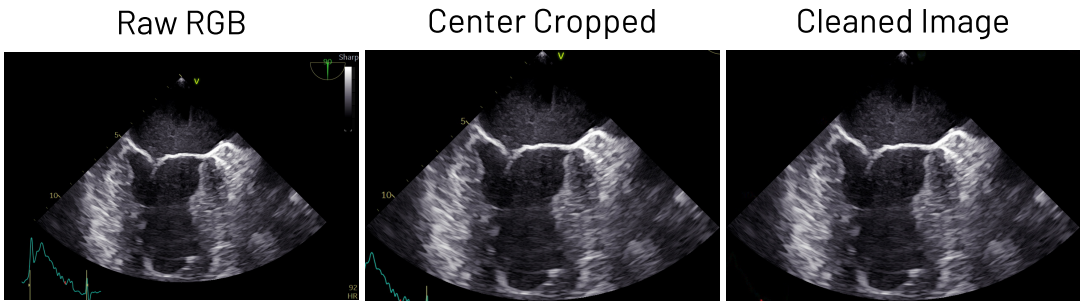


Figure 3.1: DICOM to Data. The preprocessing steps to go from the raw acquisitions to TEE images ready for labelling and use in training models.

In their raw format, the TEE videos were provided as DICOM files. To enable their use with AI-based methods, the DICOM files required pre-processing. All pre-processing was carried out in Python using Pydicom [110] and OpenCV [111]:

Colour space: Each video was converted from its original colour space to the RGB colour space. The source colour space was obtained from the DICOM tag (0028, 0004), *Photometric Interpretation*, with the majority of acquisitions originally encoded in the YBR_FULL_422 colour space.

Calibration region: Frames were cropped to a central region containing only the ultrasound cone and its contents. The coordinates defining the bounding box were retrieved from DICOM tag (0018, 6011), *Sequence of Ultrasound Regions*.

Annotation removal: Most videos contained overlaid annotations such as ECG traces, scale bars, and textual information. These overlays were typically rendered in colours distinct from the underlying ultrasound acquisition, allowing them to be removed via colour-based filtering. Each frame was converted to the HSV colour space, and multiple threshold ranges were evaluated to identify a range that removed the annotations while preserving the underlying image content. Through

iterative testing, the optimal filtering range was determined to be H: (0, 100), S: (20, 255), and V: (20, 255). Following filtering, any resulting gaps were filled using inpainting, specifically the `cv2.INPAINT_TELEA` method with an inpaint radius of 3.

An example of the full processing pipeline, from the original RGB frame to the final ready-for-use image, is shown in Fig. 3.1.

Over the course of the DPhil, the dataset expanded as additional videos were acquired and shared by GE HealthCare. This resulted in two curated subsets, referred to as TEE-A and TEE-B.

TEE-A

TEE-A was utilised in Chapter 4 and consists of 2,914 images of the mid-esophageal four-chamber view, sourced from 26 subjects. A subset of this dataset, comprising 182 images from 14 subjects, was annotated by clinicians with LV labels.

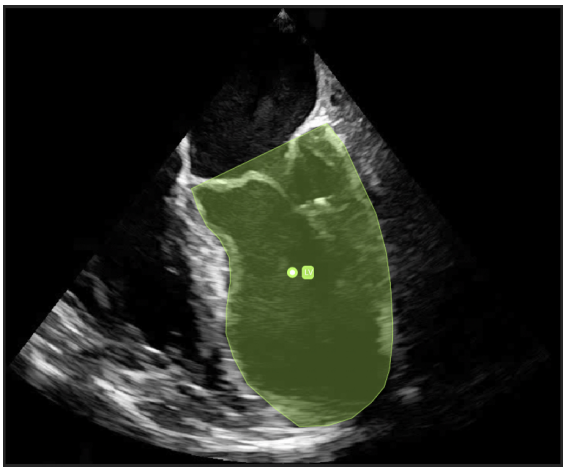


Figure 3.2: TEE-A Image & Mask. Example TEE-A acquisition complete with semantic labels for the **LV**.

TEE-B

TEE-B was utilised in Chapter 5 and comprises 288 image–mask pairs drawn from 71 mid-esophageal two- and four-chamber videos, each from a distinct subject. The recruited clinicians and I annotated the LA, LV, RA, and RV in four evenly spaced frames within a cardiac

cycle, starting from the end-diastolic frame, resulting in a total of 288 annotated frames. The clinicians labelled $\sim 70\%$ of the videos and reviewed the annotations I performed.

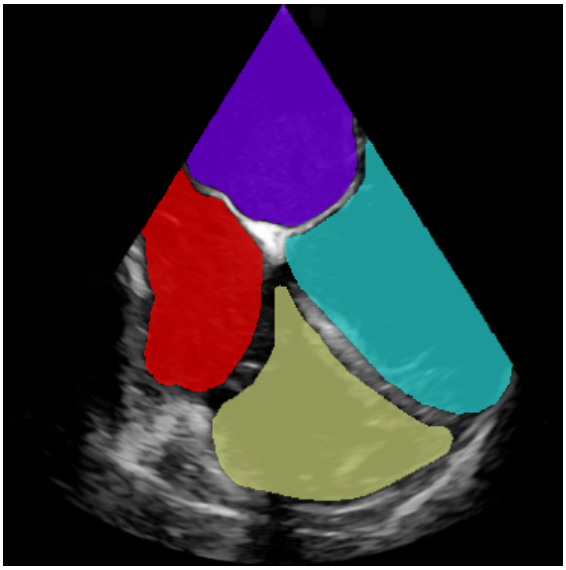


Figure 3.3: TEE-B Image & Mask. Example TEE-B acquisition complete with semantic labels for the LA, LV, RA, and RV.

3.2 Public Datasets

3.2.1 SSM Dataset

During this DPhil, anatomical heart models created by Strocchi et al. [112] and expanded into Statistical Shape Models (SSMs) by Gilbert et al. [13] and Cristobal et al. [113] were utilised. Strocchi et al. [112] created 20 four-chamber heart models using CT images of asymptomatic hearts acquired during the diastolic phase. This 20-patient cohort was 30% female with an average age of 51 ± 8 years and an average weight of 85 ± 19 kg. Firstly, parts of the heart were automatically segmented and post-processed, then tetrahedral meshes were constructed using Computational Geometry Algorithm Library [114]. More information on the mesh generation can be found in [112]. Using 19 heart models, Cristobal et al. [113] constructed a statistical shape model that captured population-level anatomical variability. Principal component analysis was then used to sample this learned shape space, generating 99 anatomically plausible cardiac models [115]. We call this dataset containing 99 heart models the *SSM Dataset* and was used in

Chapter 4. An example heart model is shown in Fig. 3.5 and a full list of the structures in the model are in Table 3.4.

3.2.2 CAMUS

The CAMUS (Cardiac Acquisitions for Multi-structure Ultrasound Segmentation) dataset, introduced by Leclerc et al. [9], comprises echocardiographic acquisitions from 500 patients, each providing apical two-chamber and four-chamber views captured between the end-diastolic and end-systolic cardiac phases. Thus the total number of videos is 1,000.

The echo scans were captured on GE Vivid E95 ultrasound systems using an M5S probe, and output via the proprietary EchoPAC software. The original acquisition was in polar coordinates; for consistency, all images were interpolated onto a Cartesian grid with a common spatial resolution.

The data is accompanied by manual segmentations of three cardiac structures: the LV, the LV_{epi}, and the LA. The authors intentionally avoided filtering or excluding "difficult" cases, such as images with missing wall boundaries, orientation deviations, or poor image quality, in order to reflect the heterogeneity of real-world echocardiography.

Figure 3.4: SSM Dataset Anatomical Heart Model Structures. All structures represented in the heart models of the SSM Dataset, grouped by types. Left-atrium (LA), Left-ventricle (LV), Pulmonary Valve (PV), and Pulmonary Artery (PA).

Heart Model Structures		
Vein Rings	Tissue	Valve Borders
LA Appendage Ring	LV Myocardium	Mitral Valve Border
Left Superior PV	RV Myocardium	Tricuspid Valve Border
Left Inferior PV	LA Myocardium	Aortic Valve Border
Right Inferior PV	RA Myocardium	Pulmonary Valve Border
Right Superior PV		LA Appendage Border
	Aortic Wall Thickness	Left Superior PV Border
Superior Vena Cava Ring		Left Inferior PV Border
Inferior Vena Cava Ring	PA Wall Thickness	Right Inferior PV Border
		Right Superior PV Border
		Superior Vena Cava Border
		Inferior Vena Cava Border

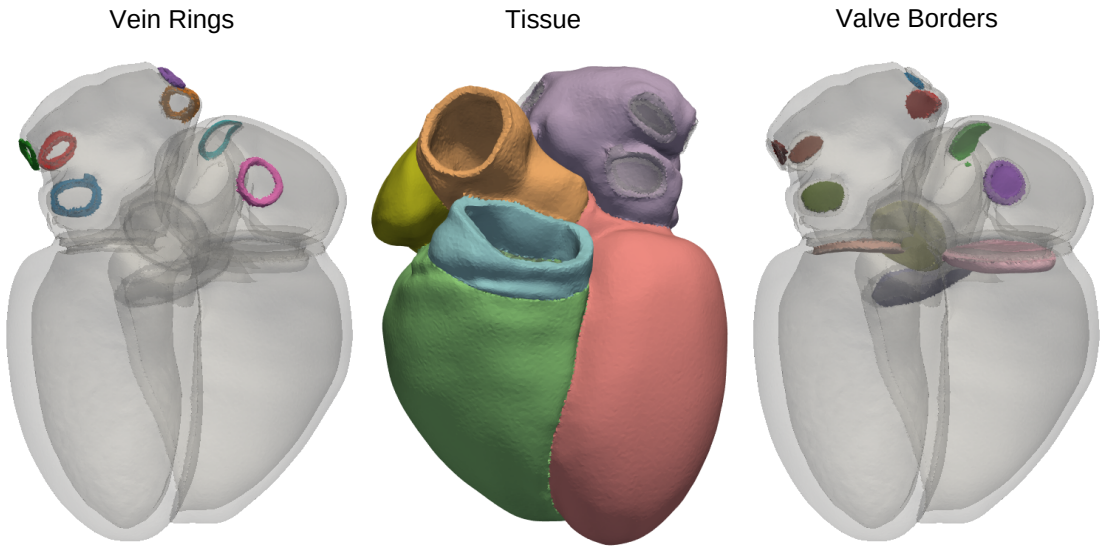


Figure 3.5: Anatomical Models and Structure Labels. Structure labels for an example heart model from the SSM Dataset. *Vein Rings* show the rings where the veins are cropped, including the LAA. *Tissue* shows the myocardium of the four major chambers, namely, LV, RV, LA, RA, as well as the wall of the cropped aorta and pulmonary artery. *Valve Borders* shows the planes added at inlets and outlets of the four major chambers.

Your word is a lamp to my feet and a light to my path.

— Psalm 119:105

4

Unpaired Transesophageal Echocardiography Generation

Contents

4.1 Introduction 44
4.1.1 Contributions 46
4.2 Pseudo-image Generation 46
4.2.1 Data 47
4.2.2 Methods 47
4.2.3 Results 50
4.2.4 Discussion 61
4.2.5 Conclusion 63
4.3 Synthetic Image Generation 64
4.3.1 Methods 64
4.3.2 Experiments 66
4.3.3 Results & Discussion 70
4.4 Conclusion 75

This first contribution chapter proposes `HeartToEcho`, a method for extracting idealised TEE views from 3D atlases of the heart, and converting them into realistic ultrasound images using deep generative models trained on unpaired data. By defining and extracting planes that correspond to standard TEE views, the proposed method allows the generation of synthetic datasets complete with semantic labels, readily usable for training other models for novel tasks. The work presented in this chapter was accepted

for a poster presentation at MICCAI DALI 2023:

Oladokun, E., *Abdulkareem, M., Šprem, J., Grau, V.: Transesophageal Echocardiography Generation using Anatomical Models. Lecture Notes in Computer Science, pp.43–52 (2024)*

4.1 Introduction

The ability to automatically acquire relevant physiological variables from echos can be greatly useful for diagnosis, observation and treatment planning. This automation could be accomplished using state-of-the-art computer vision methods, especially deep learning methods. However, the large amount of high-quality labelled data generally required for training is a challenge. Acquiring large amounts of TEE data is especially difficult due to its relatively infrequent use compared with TTE. Therefore, the first challenge to tackle when attempting to use such methods is often data availability.

Datasets of inadequate size and quality are a common problem in DL, motivating many attempts at finding a solution. A common approach is data augmentation, which aims to increase the size and diversity of a dataset. Rather than only using real datasets, some researchers have proposed generating synthetic datasets using synthetic augmentation methods [116]. These synthetic images are often combined with real data to form an extended dataset to be used on a downstream task, e.g. object classification or semantic segmentation. These synthetic augmentation methods generally fall into two categories, namely same-domain and cross-domain synthesis. The former involves creating data in the same domain, and the latter involves using data from a source domain to create data in a target domain. The data synthesis approach has largely been spurred on by advancement in generative methods; in particular generative adversarial networks (GANs). The advent of the CycleGAN, which allows the use of unpaired images (i.e., source-domain and target-domain images that need not have one-to-one correspondences) in the training data, has been a particularly important development [117]. For example, conditional GANs (cGANs) have been used to synthesise X-ray images [118], and pairs of MRI and CT images were synthesised using a CycleGAN [119]. Echo images are difficult to synthesise due to the complex speckle patterns and numerous

artefacts. Attempts to synthesise echo images have traditionally taken a modelling approach where the physics behind the observed noise patterns are simulated [120]. This approach often lacks realism and has poor scalability due to the large computational power needed to run the simulations. Abdi et al. [121] took a synthetic augmentation approach to synthesise echo images by training a patch-based cGAN, but did so using a paired, scarcely annotated dataset. Paired medical datasets, even in small sizes, are rare and difficult to acquire.

Using detailed 3D anatomical models and a CycleGAN, Gilbert et al. [13] developed a pipeline where synthetic images are generated from existing high-quality annotations. Using this approach, they generated a synthetic labelled dataset of echo images and trained a CNN for LV and LA segmentation. However, their work focuses only on TTE, and only presents two views, namely, apical A4CH and A2CH. The TEE modality, however, is more data-deprived due to its invasive nature and the large variety of acquired views, making dataset acquisition more challenging. Consequently, we extend their work [13] to TEE, and also greatly increase the number and variety of supported views.

Since the inception of [13], newer methods for unpaired I2I translation have been developed, such as contrastive unpaired translation (CUT) [122]. Unlike the CycleGAN, CUT is one-sided and does not require as many auxiliary networks and loss functions. Consequently, it has lower memory requirements and requires fewer computational resources to train. Due to the emergence of other promising unpaired I2I methods, whether CUT improves upon the CycleGAN for the task is explored. Furthermore, it is desirable to know whether the synthetic images from a generator are likely to improve downstream task segmentation performance without having to train a network on each augmented dataset to compare. Current metrics for predicting augmentation impact require training networks to perform the downstream task (segmentation in this case) using clean and augmented datasets [123], which requires at least as much work as determining the test accuracy with and without the addition of augmented data. This is computationally expensive and scales poorly as the number of synthetic datasets that need evaluating grows. Therefore, unlike [13], we also investigate whether the

metrics used to evaluate I2I translation methods can predict augmentation impact on a downstream semantic segmentation task.

4.1.1 Contributions

The contributions of this chapter are as follows:

PSEUDO-IMAGE GENERATION

1. A pipeline, termed `HeartToEcho`, is developed to automatically generate anatomically grounded, labelled synthetic images corresponding to 21 standard TEE views defined by the American Society of Echocardiography [4]. This represents the first automated framework capable of producing such comprehensive, anatomically consistent pseudo-images for TEE.

SYNTHETIC IMAGE GENERATION

1. A novel method for generating synthetic TEE images using unpaired TEE pseudo-images is developed.

2. It is shown that synthetic TEE images produced by the proposed framework can directly enhance downstream left-ventricle segmentation. This establishes, for the first time in this setting, that synthetically generated TEE data can serve as an effective augmentation resource that improves model performance and supports better generalisation.

3. A systematic study is carried out to quantify how well FID aligns with expert perception of realism for synthetic TEE images. This represents the first evaluation of FID and related perceptual metrics in the context of TEE image synthesis and their first assessment as potential predictors of augmentation benefit.

4.2 Pseudo-image Generation

This section details the process for standard TEE view plane definition and extraction from anatomical models. Using Hahn et al. [25] as a reference, all standard TEE views

are analysed and, where possible, reproduced from the anatomical models. A full list of all 28 standard transesophageal views is available in Fig. 2.6 in Sec. 2.2.2.

4.2.1 Data



Figure 4.1: The 20 Initial Heart Models. Image from [113].

This chapter uses the SSM Dataset introduced in Section 3.2.1, which contains 99 heart models that are derived from 19 of the anatomical models shown in Figure 4.1.

4.2.2 Methods

According to Hahn et al. [25], there are 28 unique views required to perform a comprehensive TEE. Figure 4.2 shows an example view. Using this information, slices from approximately corresponding locations can be taken from the 3D models. The *"Imaging Plane"* column of each view was used as a target image whilst the *"Structures Imaged"* provided information for defining the slice planes.

To define each view, the fact that a plane can be defined either by using three non-collinear points or a normal vector and a point is used. From the *'Structures Imaged'* column, three structures were selected and their positions used to define the slice

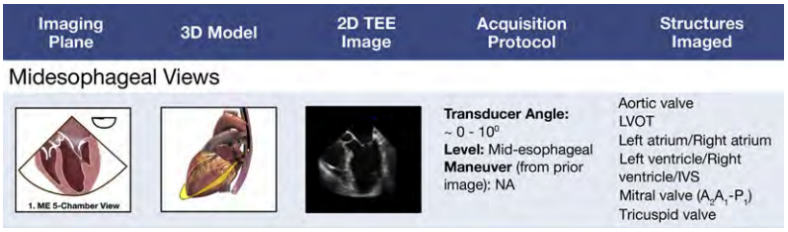


Figure 4.2: Mid-Esophageal 5-Chamber View Entry. A view entry contains: an idealised image of the view, a 3D model, the 2D ultrasound image, the protocol for its acquisition, and the salient structures that should be visible in the image. Image from [25].

plane. The position of each structure ($\vec{p}_{structure}$) is defined as the mean of all points in the model bearing its label

$$\vec{p}_{structure} = \frac{1}{N}\sum_{i=1}^{N} \vec{p}^{\,i}_{structure} \tag{4.1}$$

Structures used in the definition of the plane are called ”landmarks”. In order to mimic the natural variation in the acquisition of the ultrasounds, two mutually perpendicular axes within the plane were defined. The slice plane was then randomly rotated around both axes, thus producing a set of images where some of the landmarks are in-plane and others are off-plane. The rotation axes were defined using pairs of anatomical landmarks to generate lines approximating either the long or short axis of the heart. The long axis is loosely described as the line passing through the apex and the base of the heart. However, as the base of the heart is not represented by a single well-defined point in the anatomical models, the long axis was specifically defined as the line connecting the left ventricular apex ($\vec{p}_{lv_{apex}}$) and the mitral valve ($\vec{p}_{mv}$), which agrees with literature [124]. This is formally expressed as

$$\vec{p}_{lax} = \vec{p}_{lv_{apex}} - \vec{p}_{mv}. \tag{4.2}$$

This definition provides a close approximation to the conventional long-axis orientation and yields visually comparable views to long-axis echocardiographic views. Similarly, the short-axis was defined as the line connecting the mitral valve to the tricuspid valve.

$$\vec{p}_{sax} = \vec{p}_{tv} - \vec{p}_{mv} \tag{4.3}$$

Following the definition of a slice plane and sampling about the axes, the slice and samples were extracted from a heart model and qualitatively verified by visual assessment. This

assessment was performed by me through comparison of the resulting slice plane with the idealised view shown in the *'Imaging Plane'* section. If the slice did not sufficiently match the idealised image of the view, and alternative structures were available, then alternative landmarks were used, and the process was repeated until either the slice plane contained the expected structures shown in the idealised image or there were no more structures to use as landmarks. This process was also repeated with different heart models in the SSM Dataset to ensure the landmarks were able to generate a slice similar to the expected views, despite variations in the heart morphology.

Starting from the slice extraction stage, a pseudo-image is generated through a heuristic sequence of randomised photometric transformations modelled after common ultrasound artefacts. Each tissue type is first re-mapped to a fixed reference intensity (175 for tissue, 250 for pericardium, and 50 for intracardiac regions on a 0–255 scale). Multiplicative speckle noise is then applied by sampling a uniform random field $\mathcal{U}(0,1)$ at one-eighth of the working resolution and nearest-neighbour upsampling it to match the image dimensions before pixel-wise multiplication; for an $R \times R$ working image, an $(R/8) \times (R/8)$ noise field is sampled and upsampled to $R \times R$, producing speckle patches of approximately 8×8 pixels. A view-specific cone-origin shadow and up to three randomised Gaussian "spots" (each retained with probability 0.5) are then overlaid to approximate the larger-scale brightness inhomogeneities characteristic of ultrasound imaging. Additive noise is then applied in two passes (uniform $\mathcal{U}(0,100)$ at quarter- and full-resolution) and pixel intensities within the cardiac region are attenuated by 10% to mimic acoustic shadowing. Two successive Gaussian blurs (radii of 7 and 3 pixels) are applied to replicate the loss of high-frequency detail, after which the image is resized to 256×256 pixels and further additive noise drawn from $10 \cdot \mathcal{N}(0,1)$ at quarter- and half-resolution is added. Finally, a Gaussian blur of radius 5 pixels is applied, the global intensity range is rescaled to $[a,b]$ with $a \sim \mathcal{U}(5,30)$ and $b \sim \mathcal{U}(230,250)$ to introduce variability in contrast and brightness, and the ultrasound-cone mask is applied. Because every stage is independently parameterised, the pipeline can be retuned for a specific set of echos (e.g., a dataset acquired by a particular scanner) by re-fitting the noise, blur, and intensity-rescaling distributions to that dataset's measured

statistics. The reference implementation, including view-specific configurations and the exact hyperparameters used in this chapter, is released at `https://github.com/EngEmmanuel/pseudo-image-extraction`.

4.2.3 Results

Table 4.1: List of TEE Views and Their Generation State. *Probe Region:* the region of the transducer during image acquisition, *View No.:* The number of the view, *View Name:* Descriptor of the view, *Generation Status:* whether or not the view was successfully generated, *Reason:* If the view could not be generated, why was this the case? Reason A means the view could not be generated due to the absence of the full aorta structure in the models. Reason B means it was due to the absence of the full atrial appendage. Acronyms: Mid-esophageal (ME), Upper-esophageal (UE), Transgastric (TG) , short axis (SAX), long axis (LAX), Aortic Valve (AV), Right ventricular (RV).

Probe Region	View No.	View Name	Generation Status	Reason
Mid-Esophageal	1	ME 5-Chamber View	✓	-
	2	ME 4-Chamber View	✓	-
	3	ME Mitral Commissural View	✓	-
	4	ME 2-Chamber View	✓	-
	5	ME LAX View	✓	-
	6	ME AV LAX View	✓	-
	7	ME Ascending Aorta LAX View	✓	-
	8	ME Ascending Aorta SAX View	×	A
	9	ME Right Pulmonary Vein View	×	A
	10	ME AV SAX View	✓	-
	11	ME RV Inflow-Outflow	✓	-
	12	ME Modified Bicaval TV View	✓	-
	13	ME Bicaval View	✓	-
	14	UE Right and Left Pulmonary Veins View	✓	-
	15	ME Left Atrial Appendage View	×	B
Transgastric	16	TG Basal SAX View	✓	-
	17	TG Mid Papillary SAX View	✓	-
	18	ME TG Apical SAX View	✓	-
	19	TG RV Basal View	✓	-
	20	TG RV Inflow-Outflow View	✓	-
	21	Deep TG 5-chamber View	✓	-
	22	TG 2-Chamber View	✓	-
	23	TG RV Inflow View	✓	-
	24	TG LAX View	✓	-
Upper-Esophageal	25	Descending Aorta SAX View	×	A
	26	Descending Aorta LAX View	×	A
	27	UE Aortic Arch LAX View	×	A
	28	UE Aortic Arch SAX View	×	A

Table 4.1 shows that out of all 28 views, slices were generated for 21. Figure 4.3 and Figure 4.4 show two TEE views, the resultant slices generated, and the landmarks used in their definition. Furthermore, the slices acquired by sampling away from the defined planes can be seen in the first and last columns

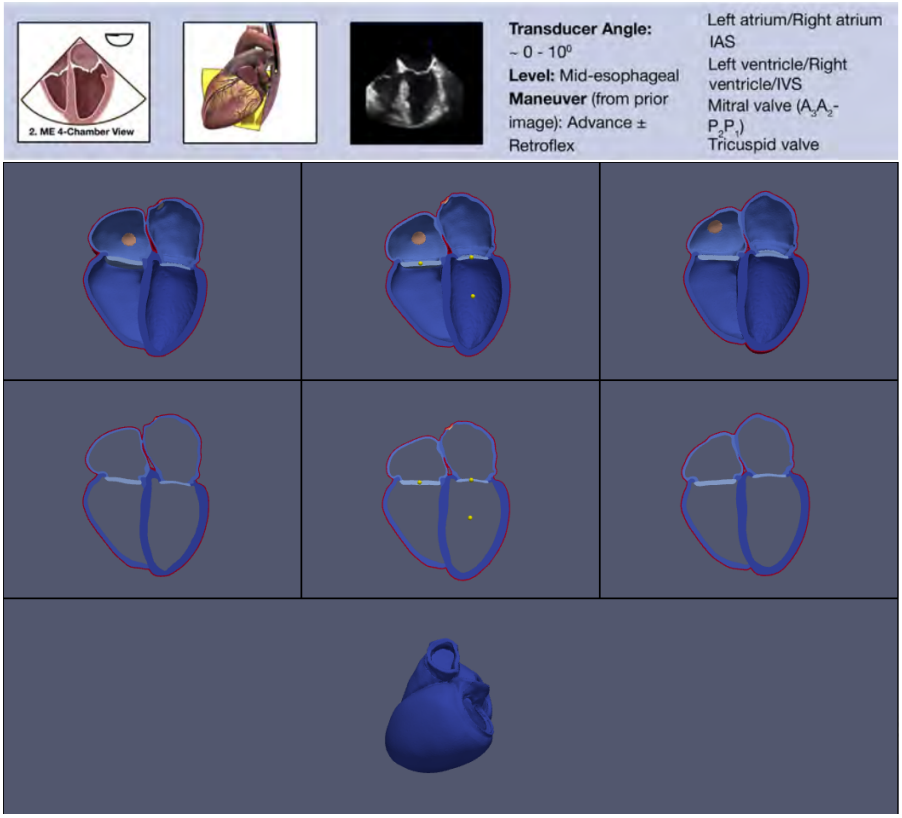


Figure 4.3: Generation of Mid-Esophageal 4-Chamber View and Off-Planes. The slice definition uses the LV, mitral-, and tricuspid- valve as landmarks. The middle column shows the plane defined by the landmarks whilst the slices either side are generated by randomly rotating about the sampling axes.

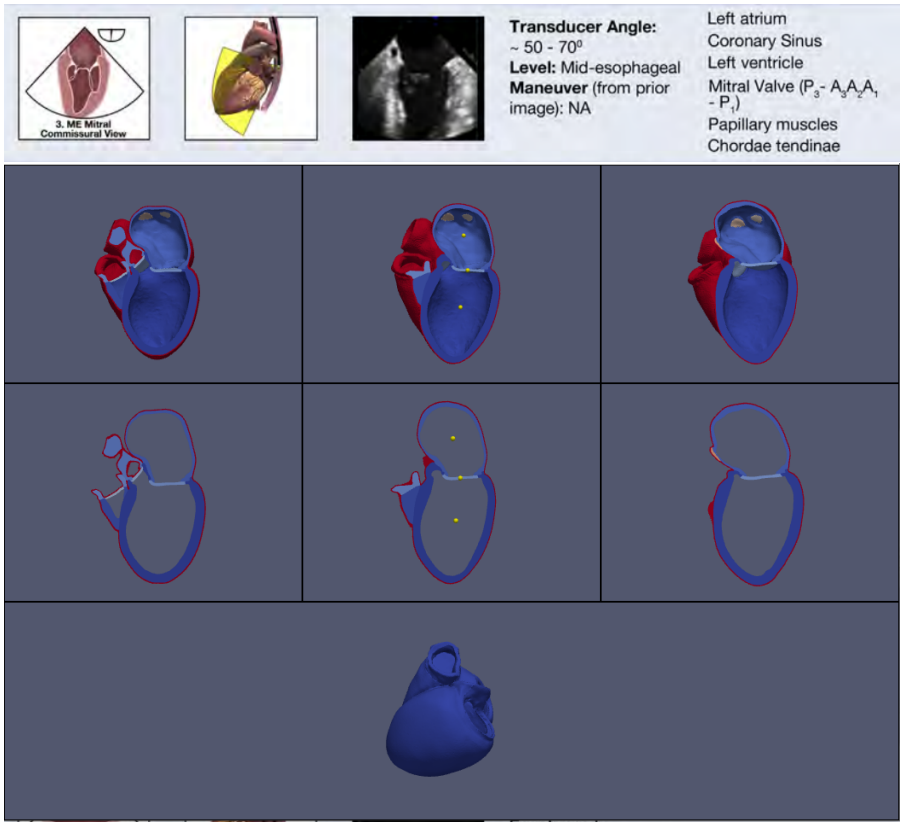


Figure 4.4: Generation of Mid-Esophageal Mitral Commissural View and Off-Planes. The slice definition uses the LV, mitral valve, and LA as landmarks. The middle column shows the plane defined by the landmarks whilst the slices either side are generated by randomly rotating about the sampling axes.

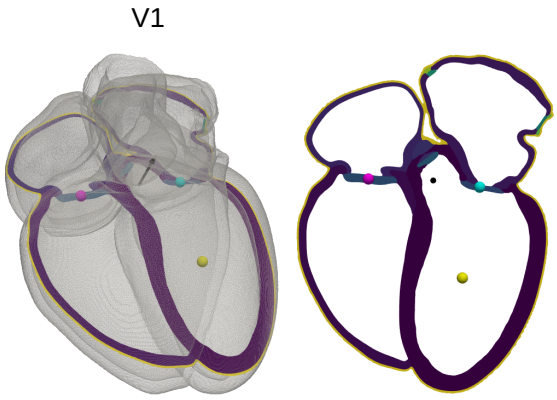


Figure 4.5: V1. Generated Mid-Esophageal Five-Chamber View. Landmarks shown are the mitral valve, tricuspid valve, and left ventricle. The plane passes through the point $\frac{2.5 \cdot \vec{p}_{av} + \vec{p}_{mv} + \vec{p}_{tv}}{4.5}$ with normal $(\vec{p}_{tv} - \vec{p}_{mv}) \times (\vec{p}_{tv} - \vec{p}_{lv_{apex}})$.

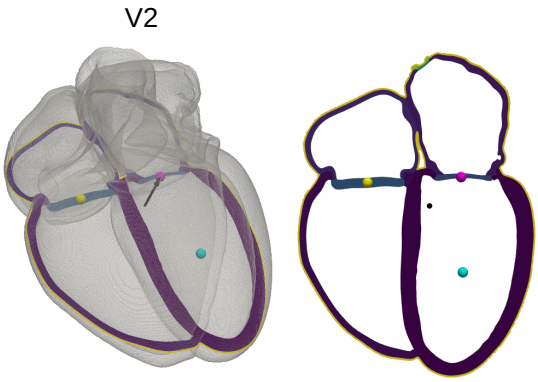


Figure 4.6: V2. Generated Mid-Esophageal Four-Chamber View. Landmarks shown are the left ventricle, mitral valve, and tricuspid valve. The plane passes through the point $(\vec{p}_{lv} + \vec{p}_{mv} + \vec{p}_{tv})/3$ with normal $(\vec{p}_{tv} - \vec{p}_{mv}) \times (\vec{p}_{tv} - \vec{p}_{lv})$.

Generated Views

As seen in Table 4.1, 21 out of the 28 standard TEE views were reproduced. The resultant slices and the vectors required for their definition are presented here.

For some views, structures that are not explicitly labelled in the anatomical models were approximated using available ones. For example, the inter-arterial septum's position was defined as $(\vec{p}_{la} + \vec{p}_{ra})/2$. Similarly, some views targeted the ventricular apices rather than the centres $(\vec{p}_{lv}, \vec{p}_{rv})$. Consequently, the left ventricular apex $(\vec{p}_{lv_{apex}})$ and right ventricular apex $(\vec{p}_{rv_{apex}})$ are defined as the points on the left ventricle and right ventricle that are furthest from the mitral valve and tricuspid valve, respectively.

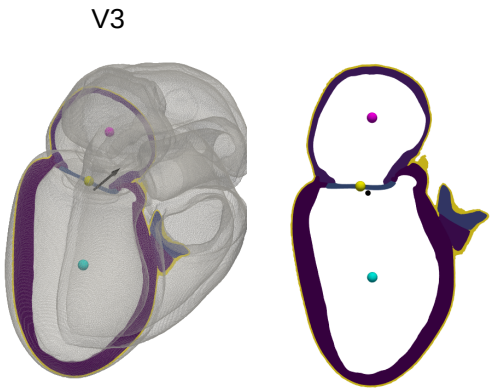


Figure 4.7: V3. Generated Mid-Esophageal Mitral Commissural View. Landmarks shown are the left ventricle, left atrium, and mitral valve. The plane passes through the point $(\vec{p}_{mv} + \vec{p}_{la} + \vec{p}_{lv})/3$ with normal $(\vec{p}_{la} - \vec{p}_{lv}) \times (\vec{p}_{la} - \vec{p}_{mv})$.

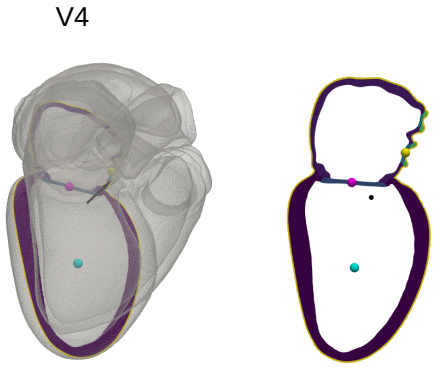


Figure 4.8: V4. Generated Mid-Esophageal Two-chamber View. Landmarks shown are the left ventricle, mitral valve, and left atrial appendage. The plane passes through the point $(\vec{p}_{lv} + \vec{p}_{mv} + \vec{p}_{laa})/3$ with normal $(\vec{p}_{mv} - \vec{p}_{lv}) \times (\vec{p}_{mv} - \vec{p}_{laa})$.

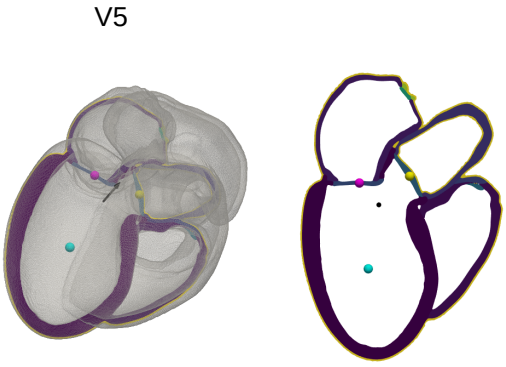


Figure 4.9: V5. Generated Mid-Esophageal Long-Axis View. Landmarks shown are the left ventricle, mitral valve, and aortic valve. The plane passes through the point $(\vec{p}_{lv} + \vec{p}_{av} + \vec{p}_{mv})/3$ with normal $(\vec{p}_{mv} - \vec{p}_{lv}) \times (\vec{p}_{mv} - \vec{p}_{av})$.

V6

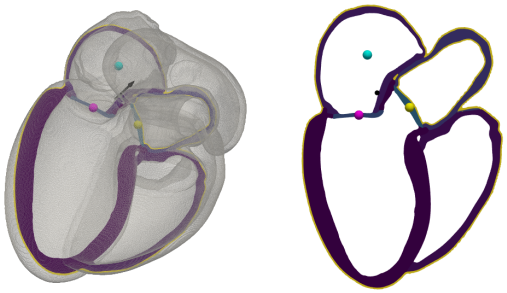


Figure 4.10: V6. Generated Mid-Esophageal Aortic Valve Long-Axis View. Landmarks shown are the left atrium, mitral valve, and aortic valve. The plane passes through the point $(\vec{p}_{la} + \vec{p}_{mv} + \vec{p}_{av})/3$ with normal $(\vec{p}_{mv} - \vec{p}_{la}) \times (\vec{p}_{mv} - \vec{p}_{av})$.

V10

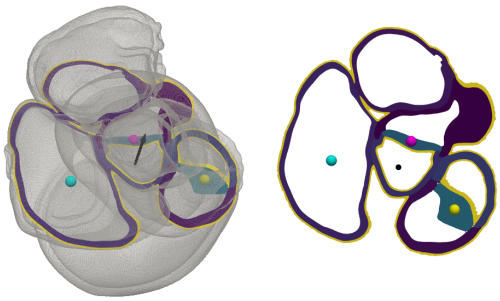


Figure 4.11: V10. Generated Mid-Esophageal Aortic Valve Short-Axis View. Landmarks shown are the right atrium, aortic valve, and pulmonary valve. The plane passes through the point $(\vec{p}_{ra} + \vec{p}_{av} + \vec{p}_{pv})/3$ with normal $(\vec{p}_{av} - \vec{p}_{ra}) \times (\vec{p}_{av} - \vec{p}_{pv})$.

V11

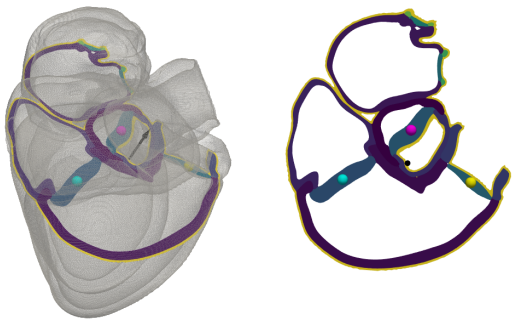


Figure 4.12: V11. Generated Mid-Esophageal Right Ventricular Inflow-Outflow View. Landmarks shown are the tricuspid valve, aortic valve, and pulmonary valve. The plane passes through the point $(\vec{p}_{tv} + \vec{p}_{av} + \vec{p}_{pv})/3$ with normal $(\vec{p}_{av} - \vec{p}_{tv}) \times (\vec{p}_{av} - \vec{p}_{pv})$.

V12

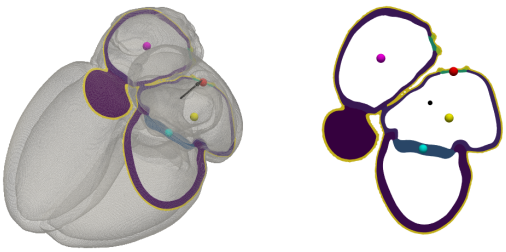


Figure 4.13: V12. Generated Mid-Esophageal Modified Bicaval Tricuspid Valve View. Landmarks shown are the tricuspid valve, superior vena cava, left atrium, and right atrium. The interatrial septum point is approximated as $\vec{p}_{ias} = (\vec{p}_{la} + \vec{p}_{ra})/2$. The plane passes through the point $(\vec{p}_{tv} + \vec{p}_{ias} + \vec{p}_{svc})/3$ with normal $(\vec{p}_{ias} - \vec{p}_{tv}) \times (\vec{p}_{svc} - \vec{p}_{tv})$.

V13

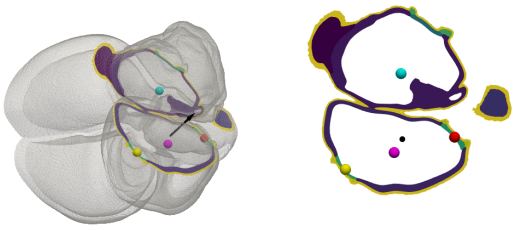


Figure 4.14: V13. Generated Mid-Esophageal Bicaval View. Landmarks shown are the left atrium, right atrium, superior vena cava, and inferior vena cava. The interatrial septum point is approximated as $\vec{p}_{ias} = (\vec{p}_{la} + \vec{p}_{ra})/2$. The plane passes through the point $(\vec{p}_{ias} + \vec{p}_{ivc} + \vec{p}_{svc})/3$ with normal $(\vec{p}_{ivc} - \vec{p}_{ias}) \times (\vec{p}_{ivc} - \vec{p}_{svc})$.

V14

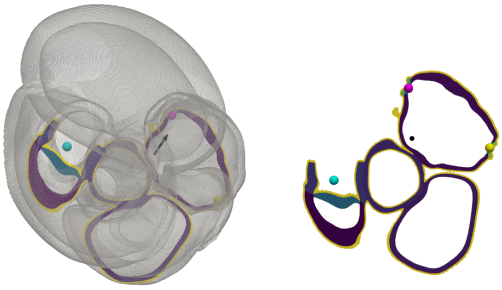


Figure 4.15: V14. Generated Upper-Esophageal Right and Left Pulmonary Veins View. Landmarks shown are the pulmonary artery, left superior pulmonary valve, and right inferior pulmonary valve. The plane passes through the point $(\vec{p}_{pa} + \vec{p}_{lspv} + \vec{p}_{ripv})/3$ with normal $(\vec{p}_{lspv} - \vec{p}_{pa}) \times (\vec{p}_{lspv} - \vec{p}_{ripv})$.

V16

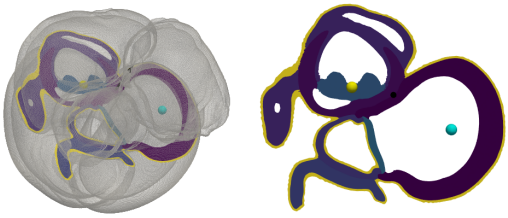


Figure 4.16: V16. Generated Transgastric Basal Short-Axis View. Landmarks shown are the left ventricle and tricuspid valve. The plane passes through the point $(\vec{p}_{mv} + \vec{p}_{tv})/2$ with normal $(\vec{p}_{mv} - \vec{p}_{lv_{apex}})$.

V17

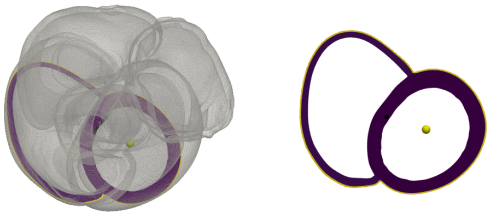


Figure 4.17: V17. Generated Transgastric Mid-Papillary Short-Axis View. The shown landmark is the left ventricle. The plane passes through the point $(\vec{p}_{rv} + \vec{p}_{lv})/2$ with normal $(\vec{p}_{mv} - \vec{p}_{lv_{apex}})$.

V18

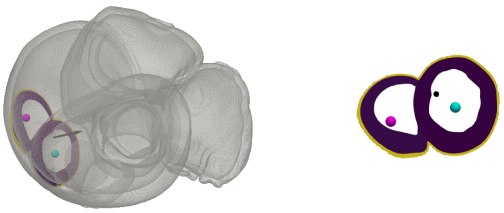


Figure 4.18: V18. Transgastric Apical Short-Axis View. Landmarks shown are the mitral valve and tricuspid valve. The plane normal is defined as $\vec{p}_{mv} - \vec{p}_{lv_{apex}}$. The plane passes through the point $\frac{(\vec{p}_{lv_{apex}} + \vec{p}_{rv_{apex}})}{2} + 10 \frac{\vec{p}_{mv} - \vec{p}_{lv_{apex}}}{\|\vec{p}_{mv} - \vec{p}_{lv_{apex}}\|}$, corresponding to a forward-shifted midpoint between the ventricular apices.

V19

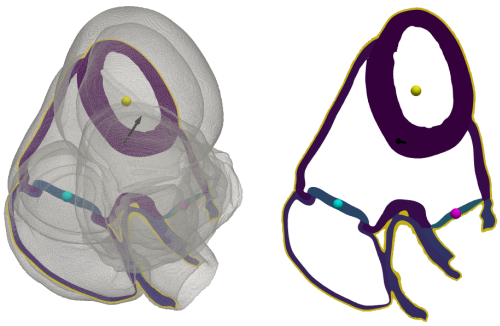


Figure 4.19: V19. Generated Transgastric Right Ventricular Basal View. Landmarks shown are the tricuspid valve, pulmonary valve, and left ventricle. The plane passes through the point $(\vec{p}_{tv} + \vec{p}_{pv} + \vec{p}_{lv_{apex}})/3$ with normal $(\vec{p}_{pv} - \vec{p}_{tv}) \times (\vec{p}_{pv} - \vec{p}_{lv_{apex}})$.

V20

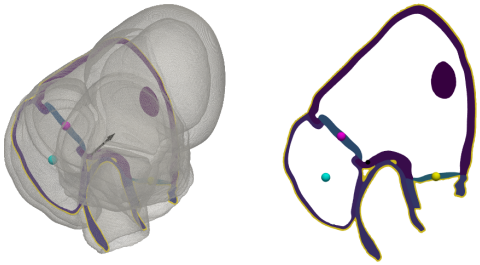


Figure 4.20: V20. Generated Transgastric Right Ventricular Inflow-Outflow. Landmarks shown are the right atrium, tricuspid valve, and pulmonary valve. The plane passes through the point $(\vec{p}_{ra} + \vec{p}_{tv} + \vec{p}_{pv})/3$ with normal $(\vec{p}_{tv} - \vec{p}_{ra}) \times (\vec{p}_{tv} - \vec{p}_{pv})$.

V21

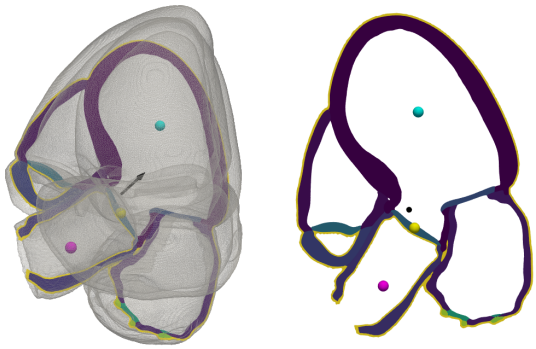


Figure 4.21: V21. Deep Transgastric Five-Chamber View. Landmarks shown are the left ventricle, aorta, and aortic valve. The plane passes through the point $(\vec{p}_{lv} + \vec{p}_{aorta} + \vec{p}_{av})/3$ with normal $(\vec{p}_{aorta} - \vec{p}_{lv}) \times (\vec{p}_{aorta} - \vec{p}_{av})$.

V22

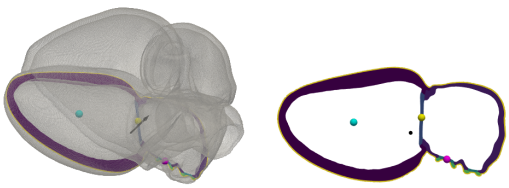


Figure 4.22: V22. Generated Transgastric Two-Chamber View. Landmarks shown are the left ventricle, left atrial appendage, and mitral valve. The plane passes through the point $(\vec{p}_{lv} + \vec{p}_{laa} + \vec{p}_{mv})/3$ with normal $(\vec{p}_{laa} - \vec{p}_{lv}) \times (\vec{p}_{laa} - \vec{p}_{mv})$.

V23

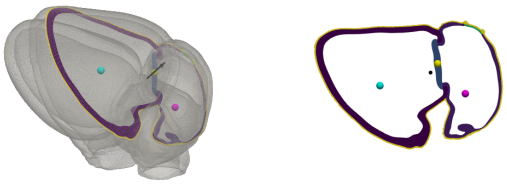


Figure 4.23: V23. Generated Transgastric Right Ventricle Inflow View. Landmarks shown are the right ventricular, right atrium, and tricuspid valve. The plane passes through the point $(\vec{p}_{rv} + \vec{p}_{ra} + \vec{p}_{tv})/3$ with normal $(\vec{p}_{ra} - \vec{p}_{rv}) \times (\vec{p}_{ra} - \vec{p}_{tv})$.

V24

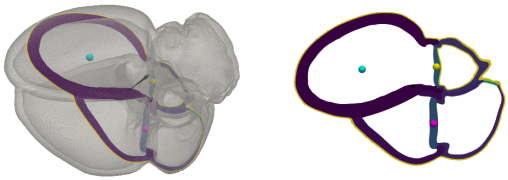


Figure 4.24: V24. Generated Transgastric Long-Axis View. Landmarks shown are the left ventricle, tricuspid valve, and aortic valve. The plane passes through the point $(\vec{p}_{lv} + \vec{p}_{tv} + \vec{p}_{av})/3$ with normal $(\vec{p}_{tv} - \vec{p}_{lv}) \times (\vec{p}_{tv} - \vec{p}_{av})$.

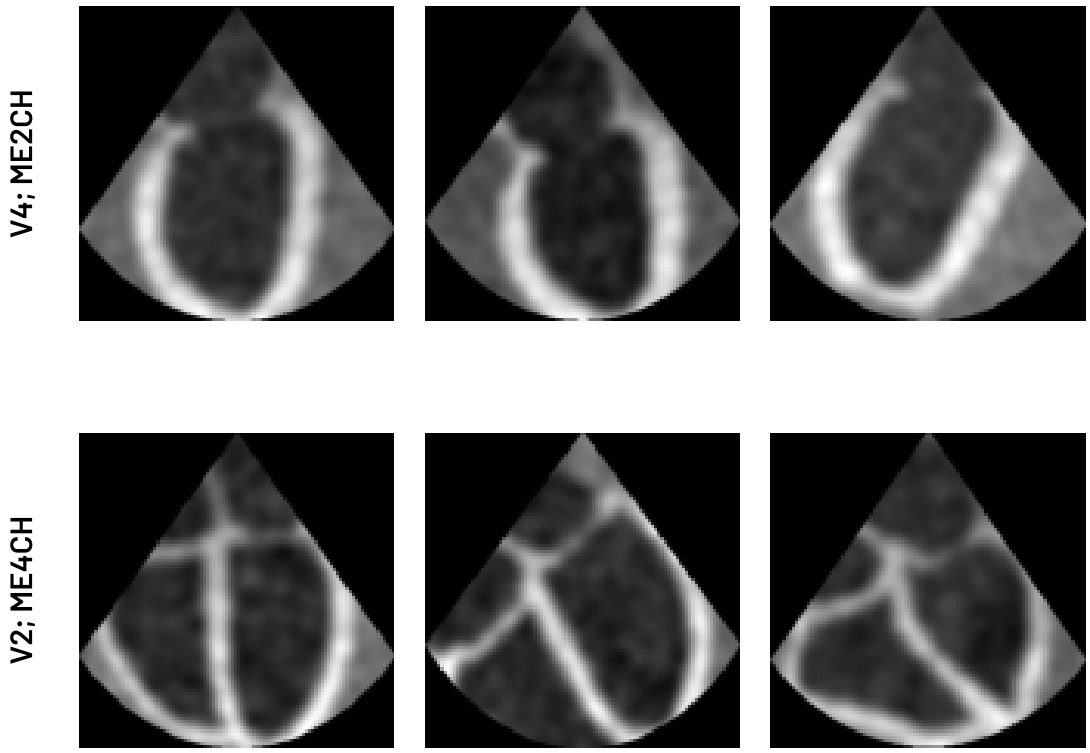


Figure 4.25: ME2CH and ME4CH Pseudo-images. Pseudo-images generated using slices from different heart models by adding an ultrasound cone, gaussian blur, and speckle noise.

Example pseudo-images produced by the pipeline of Sec. 4.2.2 are shown in Figure 4.25.

4.2.4 Discussion

The pipeline developed by [125] has been extended to support the generation of 21 novel TEE views, as shown in Table 4.1. Figure 4.3 demonstrates the quality of the slices obtained by the extended pipeline. All desired structures appear in the generated slices, despite only a subset of landmarks (LV, mitral valve, and tricuspid valve) being used to define the slice plane. The resulting slices capture the dominant anatomical structures expected in the corresponding standard TEE views and exhibit appearances sufficient to serve as the basis for the subsequent I2I translation stage, and also mimic variations that occur during acquisition.

There were, however, scenarios in which the major anatomical features of a slice were present but finer details were absent because they do not exist in the anatomical

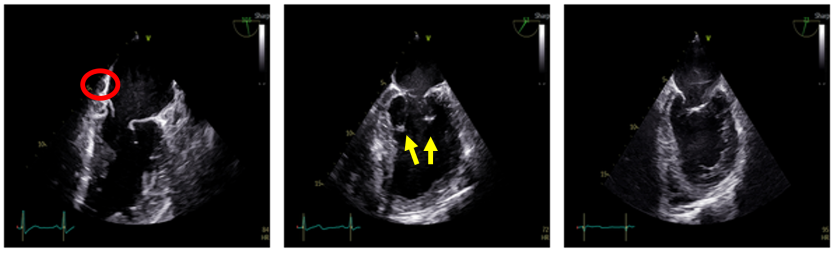


Figure 4.26: Real Mid-Esophageal Mitral Commissural Views. The Red circle shows the Coronary Sinus and the yellow arrows point to faint chordae tendinae. These real acquisitions share the same high-level, dominant anatomical features as the generated slice shown in Fig. 4.7.



Figure 4.27: V25. Descending Aorta SAX View. View 25 out of 28 views required for a comprehensive TEE. Image from [25].

models. For example, the slices shown in Fig. 4.4 do not contain structures such as the chordae tendinae or coronary sinus, because the anatomical models do not. Despite this limitation, the resulting slices preserve the dominant anatomical layout that characterise the corresponding idealised views.

Visual comparison with real TEE acquisitions (Fig. 4.26) shows that these major features are sufficient to represent the intended view, particularly given that fine structures such as the chordae tendinae and coronary sinus are not consistently visible in clinical ultrasound images. With the dominant anatomical features present, the pseudo-images provide sufficient structural information, and the addition of finer details can be handled by subsequent I2I translation methods. Taken together, these observations indicate that the extracted slices capture the essential characteristics of the target views and are therefore suitable for subsequent ultrasound image generation.

Not all views contained at least three labelled anatomical structures in the heart models; consequently, only a subset of the 28 standard TEE views could be generated. Figure 4.27 shows an example of a view that could not be reproduced. This view is irreproducible because the heart models contain only the base of the aorta, rather than the ascending or descending segments required for this view. In such cases, the aorta

represents a dominant feature, and its absence leads to slices that lack critical anatomical information, which would result in inadequate ultrasound images.

Table 4.1 shows that six (V7, V8, V25, V26, V27, V28) of the seven views that could not be reproduced relied on portions of the aorta that are not present in the heart models. The remaining view, V15, could not be reproduced because the models include only the left atrial appendage border and ring, rather than the complete structure.

The views that could not be reproduced were therefore limited to those requiring anatomical structures absent from the heart models, most notably the full aorta and left atrial appendage. While this represents a limitation of the models, the majority of standard views were successfully generated and most clinically relevant structures were adequately represented. As a result, downstream image generation tasks would be largely unaffected, as the available views provide sufficient anatomical context to produce realistic pseudo-images and, ultimately, high-quality synthetic ultrasound data.

A key strength of `HeartToEcho` is that it makes no assumptions about the absolute positions of anatomical structures, as all landmarks are computed directly from each individual heart model. Consequently, the method is applicable to hearts with unusual morphology and shows potential for use in downstream tasks such as chamber and valve detection and segmentation.

4.2.5 Conclusion

In this study, the pipeline developed by Gilbert et al. [13] was extended to support the generation of standard TEE views. A total of 21 out of 28 standard TEE views were successfully implemented, substantially expanding the original framework beyond TTE. The generated slice planes were shown to closely resemble idealised TEE views, despite being defined using a limited set of anatomical landmarks, and were sufficient to support realistic pseudo-image generation. Limitations in view coverage were as a result of incomplete anatomical representation in the available heart models, rather than deficiencies in the proposed methodology. Overall, this work demonstrates the feasibility of automated TEE view generation from 3D heart models and establishes a

foundation for future synthetic ultrasound generation and downstream learning-based cardiac imaging tasks.

4.3 Synthetic Image Generation

This section presents the next step for synthetic image generation which involves the translation from pseudo-images to realistic synthetic images. Two deep learning based generative methods are investigated and the quality of the synthetic images are rigorously evaluated using qualitative and quantitative metrics.

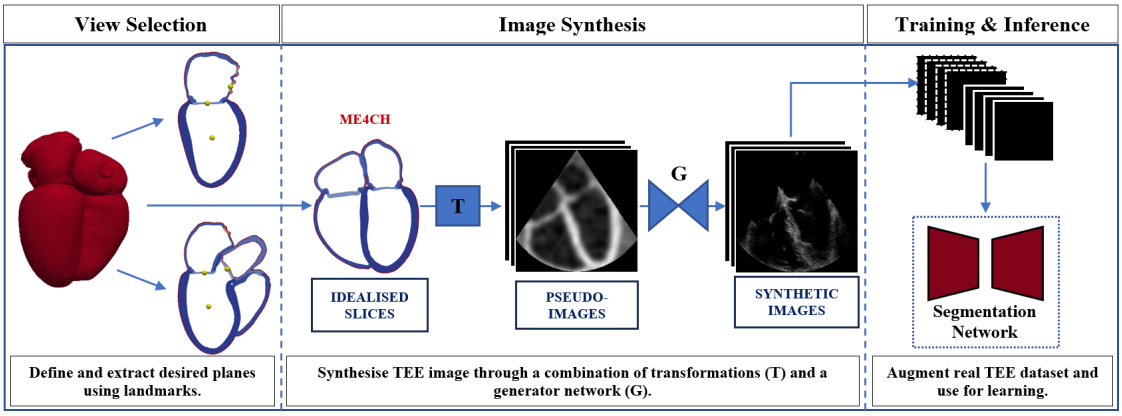


Figure 4.28: Synthetic TEE Generation and Image Segmentation Pipeline. Using landmarks, desired TEE planes are extracted from each heart model. These slices from the anatomical model serve as the anatomical guidance (conditioning masks) for the synthetic images. Next, pseudo-images are made by adding the acquisition cone and applying blurring, shadowing, and resizing transformations. The image synthesis phase concludes with pseudo-images being passed through a trained generator that transforms them into realistic synthetic images. These synthetic images and their masks can then be used to augment a real dataset for the chosen task.

4.3.1 Methods

Pseudo-images

Figure. 4.28 shows how the idealised slices defined in Sec. 4.2.2 are progressively transformed into realistic synthetic echos. The full pseudo-image pipeline, including all numerical parameters, is given in Sec. 4.2.2; the description below summarises the qualitative role of each stage in the context of the ME4CH experiment. Following slice extraction, a sequence of image-level transformations is applied to generate the final pseudo-images. These transformations are designed to approximate common ultrasound

artefacts, including speckle noise, intensity inhomogeneity, and limited spatial resolution. Multiplicative and additive noise are applied at multiple spatial scales to simulate both fine-grained speckle and lower-frequency intensity variations. Pixel intensities within the ultrasound cone are attenuated to mimic acoustic shadowing. Gaussian blurring with varying kernel sizes is then used to replicate the loss of high-frequency detail characteristic of ultrasound imaging. Finally, global intensity limits are adjusted to introduce variability in contrast and brightness across generated samples. After using these traditional transformations, the last stage utilises unpaired I2I translation approaches.

Unpaired I2I Models

Given unpaired datasets X (source domain), and Y (target domain), both the CUT [122] and CycleGAN [117] methods attempt to find an optimal generator $\hat{G}$ that best maps from $X \rightarrow Y$. The CycleGAN's two generators $\{G, F\}$ try and fool their respective discriminators $\{D_X, D_Y\}$. High-quality mappings can be attained by combining these networks and adding a cyclic-consistency loss. The CycleGAN's loss function is

$$\begin{aligned}\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{adv}(G, D_Y, X, Y) \; + \mathcal{L}_{adv}(F, D_X, Y, X) + \\ \lambda_{cyc}\mathcal{L}_{cyc}(G, F) \; + \lambda_{idt}\mathcal{L}_{idt}(G, F),\end{aligned} \tag{4.4}$$

where $\mathcal{L}_{\{\text{adv, cyc, idt}\}}$ corresponds to the adversarial, cyclic and identity losses, and $\lambda_{\{cyc,idt\}}$ allow the weighting of individual loss terms.

CUT takes a different approach and leverages contrastive learning to encourage content preservation while allowing changes in appearance. CUT samples patches from the original and generated images then via a projection head H, maps patches in the same position together in latent spaces and distances patches from different positions. Its loss function is

$$\begin{aligned}\mathcal{L}(G, H, D) = \mathcal{L}_{adv}(G, D, X, Y) + \lambda_x\mathcal{L}_{PatchNCE}(G, H, X) + \\ \lambda_y\mathcal{L}_{PatchNCE}(G, H, Y),\end{aligned} \tag{4.5}$$

where $\mathcal{L}_{PatchNCE}$ is the contrastive loss used for the patches, and $\lambda_{\{x,y\}}$ are weighting hyperparameters.

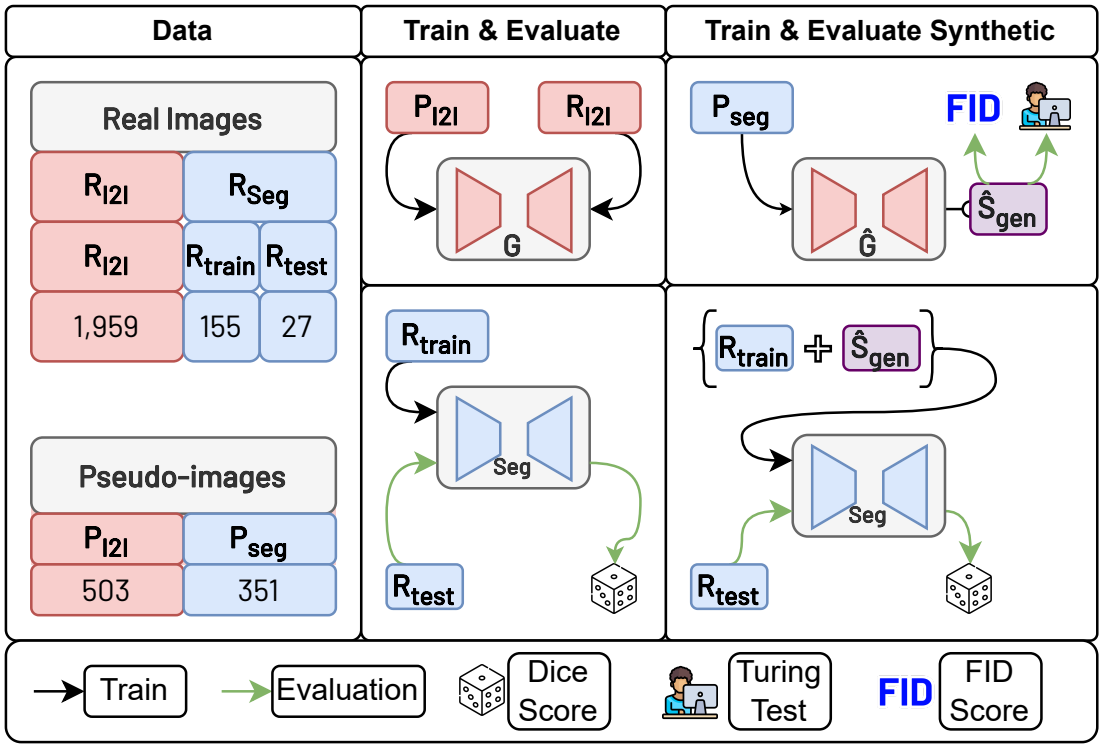


Figure 4.29: Experimental Methods for Unpaired I2I Generation. The real and pseudo-images are split into subsets dedicated to the image-to-image translation task and the segmentation task. Subsets used for image-to-image translation have the subscript 'I2I' and the segmentation subsets have the subscript 'Seg'. Once the generator is trained, the synthetic dataset $\hat{S}_{gen}$ is evaluated and also used to augment the real dataset. Finally, more nnUNet models are trained using the augmented dataset and their performance is also evaluated.

4.3.2 Experiments

Data

This study utilises the TEE-A dataset, explained in detail in Section 3.1, which contains ME4CH echos. Additionally, a dataset of pseudo-images generated using `HeartToEcho` is also used. The set of real ME4CH echos consists of 2,914 images sourced from 26 subjects. The images are sampled freely from all parts of the cardiac cycle and were sourced from several institutions in different countries by GE HealthCare. In all cases, splitting the images into subsets e.g., train and test, was performed on a per subject basis to ensure that no subject contributed images into multiple sets. Images from 12 subjects were sampled to generate R_{I2I}, which contains 1,959 images, leaving 955 images. From the remaining 955 images from 14 subjects, 182 images were labelled, and then split

Table 4.2: Training Schedule and Hyperparameters That Produced the Best Generators. LR = Learning Rate; E[] = Epoch Range; TTUR = Two time-scale update rule. Discriminator LR = TTUR_Multiplier × Generator LR.

Category	Parameter	$\hat{G}_{cut}$	$\hat{G}_{cyc}$
Architecture	Generator (G)	`resnet_9blocks`	`unet_256`
	Discriminator (D)	70×70 PatchGAN	70×70 PatchGAN
Training	Epochs (E)	290	400
	Generator LR	E[1,150]: 0.0001	E[1,200]: 0.0001
		E[151,290]: linear decay	E[201,400] linear decay
	TTUR Multiplier	2	1
Loss Weights (λ)		$\lambda_x = 1, \lambda_y = 1$	$\lambda_{cyc} = 1, \lambda_{idt} = 0.5$
Optimisation	Optimiser	Adam ($\beta_1 = 0.5, \beta_2 = 0.999$)	Adam ($\beta_1 = 0.5, \beta_2 = 0.999$)
	Batch Size	4	4
	Normalisation	Instance	Instance

into train (R_{train}) and test (R_{test}) sets of sizes 155 and 27, respectively. Each image's LV was labelled by a team of two expert cardiologists who use echo daily. The labels were generated using the polygon tool on the annotation software `v7darwin`

Using `HeartToEcho` and the SSM dataset, 854 ME4CH pseudo-images and their corresponding labels were generated. To obtain these pseudo-images, nine slices were sampled from each of the 99 heart models. These comprised a central plane with normal vector $\hat{n}$ defined using anatomical landmarks, four planes randomly sampled in either direction along the long axis $(\vec{p}_{lax})$, and a further four planes sampled about the orthogonal direction $(\hat{n} \times \vec{p}_{lax})$. Any outlier images produced during plane sampling were discarded, resulting in a total of 854 pseudo-images. The dataset was then divided into two subsets, P_{I2I} and P_{seg}, containing 503 and 351 pseudo-images, respectively.

Generative Training

To train the generators, P_{I2I} and R_{I2I} were used as the source domain and target domain data. At train time, generated images were sampled every 5 epochs. I reviewed the images and selected the best-performing generators in terms of realism, giving generators $\hat{G}_{cyc}$ and $\hat{G}_{cut}$. Table 4.2 shows the training configurations for the generators that created the best images according to visual review.

Once trained, P_{seg} which has not been seen during train time by the generators or segmenters is then transformed from pseudo-images into a synthetic dataset by each



Figure 4.30: Real vs Fake Quiz UI. The experts and researchers used the shown tool for registering their answers (on a webpage, the question and response section does not overlap the cone).

generator. Consequently, two synthetic datasets $\hat{S}_{cyc}$ and $\hat{S}_{cut}$ are generated, each containing 351 images.

Evaluation

The synthetic images were evaluated in three separate ways. To assess realism, a Turing-like test was performed in which two expert cardiologists (Expert-1 and Expert-2, with over 15 and over 23 years of clinical echocardiography experience, respectively) and six researchers were asked to classify echocardiographic images as either real or synthetic. A custom user interface was implemented using VGG's Image Annotator [126] to present individual images to participants and record their responses, as shown in Fig. 4.30.

Before the test began, participants were shown several real ME4CH echocardiograms. During the evaluation, images could be toggled back and forth before responses were provided. The quiz dataset comprised 120 images, including 60 real images, 30 generated using CUT, and 30 generated using CycleGAN.

Secondly, each dataset was evaluated using the FID score, a metric that quantifies the similarity between two datasets and is commonly used in the image-to-image translation literature [127].

Finally, the impact of synthetic data augmentation was evaluated using the nnUNet framework [128] to train UNets for LV segmentation. Baselines were generated by training nnUNet models on datasets containing randomly sampled fractions of the original training set. The synthetic datasets $\hat{S}_{cyc}$ and $\hat{S}_{cut}$ were then independently added to the real image sets, producing mixed datasets with varying fractions of real images, and datasets containing all real images and all synthetic images from a given generator. The performance difference between the baseline trained on real images only and the augmented sets is referred to as the delta (Δ) metric. In all cases, 3-fold cross-validation was used, and only real images were included in the validation sets, ensuring that performance was evaluated only on real data. The DA5 augmentation pipeline of nnU-Net was used as the baseline data augmentation strategy for all runs. For 2D inputs, the following stochastic transformations were applied to each training patch in sequence:

- **Affine transformations:** random rotation ($p = 0.4$; $\pm 180°$ for approximately isotropic patches, $\pm 15°$ otherwise) and isotropic scaling ($p = 0.2$, factor $\in [0.7, 1.43]$).
- $90°$ **rotations and axis transposition** ($p = 0.5$ each), applied only along axes of equal extent.
- **Blurring** ($p = 0.2$): one of median filtering or Gaussian blurring (sampled mutually exclusively).
- **Gaussian noise** addition ($p = 0.1$, variance $\in [0, 0.1]$).
- **Brightness** shift ($p = 0.1$, additive, $\sigma = 0.5$).
- **Contrast** adjustment ($p = 0.2$, factor $\in [0.5, 2]$), with or without range preservation.
- **Low-resolution simulation** ($p = 0.15$) via downsampling and upsampling (scale $\in [0.25, 1]$).
- **Gamma correction** on the inverted image ($p = 0.1$, $\gamma \in [0.7, 1.5]$), applied twice independently.

- **Mirroring** along both spatial axes.
- **Random rectangle erasing** ($p = 0.4$): 1–5 rectangles per patch filled with the channel mean.
- **Local intensity perturbations** ($p = 0.3$ each): additive brightness gradients and local gamma variations.
- **Sharpening** ($p = 0.2$, strength $\in [0.1, 1]$).

We refer the reader to Isensee et al. [128] for further details. The addition of synthetic images was evaluated alongside these transformations.

4.3.3 Results & Discussion

Generative Results

Table 4.3: Participant Results on the Quiz with Expert Performance and Researcher Performance Separated. Each person was shown 60 real (R) and 60 synthetic (S) images. **Accuracy was rounded to 1 d.p whilst frequencies were rounded to the nearest integer. ** 95% confidence interval.*

Human Perception Quiz						
Participant	Accuracy [%]	R as R	R as S	S as R	S as S	F1-Score
Expert 1	95.0	55	5	1	59	94.8
Expert 2	79.2	60	0	25	35	82.8
Researchers*	69.7 $[60.2, 79.2]$**	39	21	15	45	68.4

Table 4.3 shows the confusion matrices for each expert and the researcher cohort. The experts were especially adept at identifying real images compared to the researchers. Moreover, the experts had significantly fewer false negatives (Real as Synthetic), showing strong knowledge of what TEE images should look like. Observing the F1-Score, a score which helps measure precision and recall simultaneously, we see that the experts were noticeably better classifiers than the average researcher. Interestingly, Expert-1 performed extremely well in identifying the synthetic images as well. Consequently, we further investigated how they made their decisions. They commented that the real

Table 4.4: FID Between an Unseen Set of TEE Images and Synthetic Sets. A lower score represents better fidelity between the generated and real images.

FID ↓	
$\hat{G}_{cut}$	$\hat{G}_{cyc}$
188	230

Table 4.5: Accuracy on the Synthetic Images Only. A lower score shows more difficulty in identifying the generator's image as synthetic.

Accuracy (%) ↓			
$\hat{G}_{cut}$		$\hat{G}_{cyc}$	
Experts	Researchers	Experts	Researchers
96.7	91.1	60.0	58.3

image's grey areas and dark cavities were more homogeneous and that there were differences in the myocardium. This observation seemed to be the one difference that made them perform better than Expert-2.

Table 4.4 and 4.5 show FID for each generator and how successful they were at fooling the quiz participants, respectively. Interestingly, the two metrics do not agree, which suggests that FID may not correlate well with human-judged realism, perhaps for medical images in general, but at least for ME4CH TEE images. In Table 4.5, we see that $\hat{G}_{cyc}$ creates more realistic images compared to $\hat{G}_{cut}$. Furthermore, the expert and researcher cohorts produced mixed results, with Expert-1 markedly outperforming the average researcher and Expert-2 performing closer to the researcher cohort. This suggests that expert opinion can offer an advantage in some cases, although a researcher cohort can already provide informative judgments on the relative quality of a generator's synthetic images.

LV Segmentation

To reference subsets of the real dataset the notation R^{x}_{train} is used to refer to a subset containing $x\%$ of the training data. Furthermore, to represent augmented datasets with differing proportions of real and synthetic data, the union symbol is used to represent a

mixture of two datasets. For example, $[\hat{S}_{cut} \cup R^{20}_{train}]$ defines an augmented dataset containing all the synthetic images generated by $\hat{G}_{cut}$ and only 20% of the real training data.

Table 4.6 shows the dice score achieved on the test set (R_{test}) by the models trained on datasets with differing mixtures of real and synthetic data. In all columns except R^{20}_{train}, the networks trained on mixed sets performed best, showing improvements of up to 10%. The maximum dice score achieved (72.9) was by a network trained with mixed data and shows a 5.9% improvement on the baseline. This is especially significant when considering the dataset's realistic size and its heterogeneity in time and source institutions. Moreover, this improvement occurred due to leveraging data from `HeartToEcho` which did not require manual labelling, and despite the synthetic images only depicting the heart at end-diastole, just like the anatomical models they were sourced from.

Interestingly, the models trained on $[\emptyset \cup R^{20}_{train}]$ and $[\hat{S}_{cut} \cup R^{20}_{train}]$ performed better than expected compared to some sets with more real data. One explanation for this could be the randomness of the data added from 20% to 40%. Adding training data only improves performance when the added data is relevant to the task at hand. Given the heterogeneity of the data, it is likely that the domain gap between datasets in the R^{40}_{train} column and the test set is larger than the domain gap between the R^{20}_{train} column and test set. This would likely lead to a drop in performance, as seen here. Note that each cell in Table 4.6 corresponds to a single nnUNet training run with one random seed (3-fold cross-validation is used within each run, but seeds are not repeated across runs) owing to the high computational cost of training a fresh nnUNet for every dataset mixture. The reported numbers are therefore subject to some seed-level variance.

Figure 4.31 shows three examples of images the nnUNet trained purely on real images struggled to obtain an accurate segmentation, often segmenting both the right- and left-ventricle as the left-ventricle. Other columns show the predictions of nnUNet models trained on $[R_{train} \cup \hat{S}_{cyc}]$ and $[R_{train} \cup \hat{S}_{cut}]$. The model trained on data augmented with $\hat{S}_{cut}$ also struggled with the examples in the first two rows, by mistaking the RV for the LV and overestimating the chamber area. Notably, the model trained on data mixed with $\hat{S}_{cyc}$ does not share the same difficulties and achieves a strong prediction. This improvement in segmenting the LV in difficult examples like these is thought to be the reason why $\hat{S}_{cyc}$

Table 4.6: Dice score achieved on the test set. The column headers show real datasets with varying percentages of R_{train}, whilst the first column shows the source of the synthetic data and delta scores. Each non-grey element corresponds to the performance of an nnUNet trained on the union of the column and row datasets ($< column > \bigcup < row >$). **Blue** = Best Column Delta Score; **Bold =** Best Overall Dice Score.

	Dice Score (x100) ↑					
	$\emptyset$	R^{20}_{train}	R^{40}_{train}	R^{60}_{train}	R^{80}_{train}	R_{train}
$\emptyset$	-	54.7	47.9	51.1	53.5	67.0
$\hat{S}_{cut}$	34.9	50.2	48.2	49.5	63.6	69.5
Δ_{cut}	-	**-4.5**	+0.3	-1.6	**+10.1**	+2.5
$\hat{S}_{cyc}$	20.9	44.2	50.4	53.7	61.8	**72.9**
Δ_{cyc}	-	-10.5	**+2.5**	**+2.6**	+8.3	**+5.9**

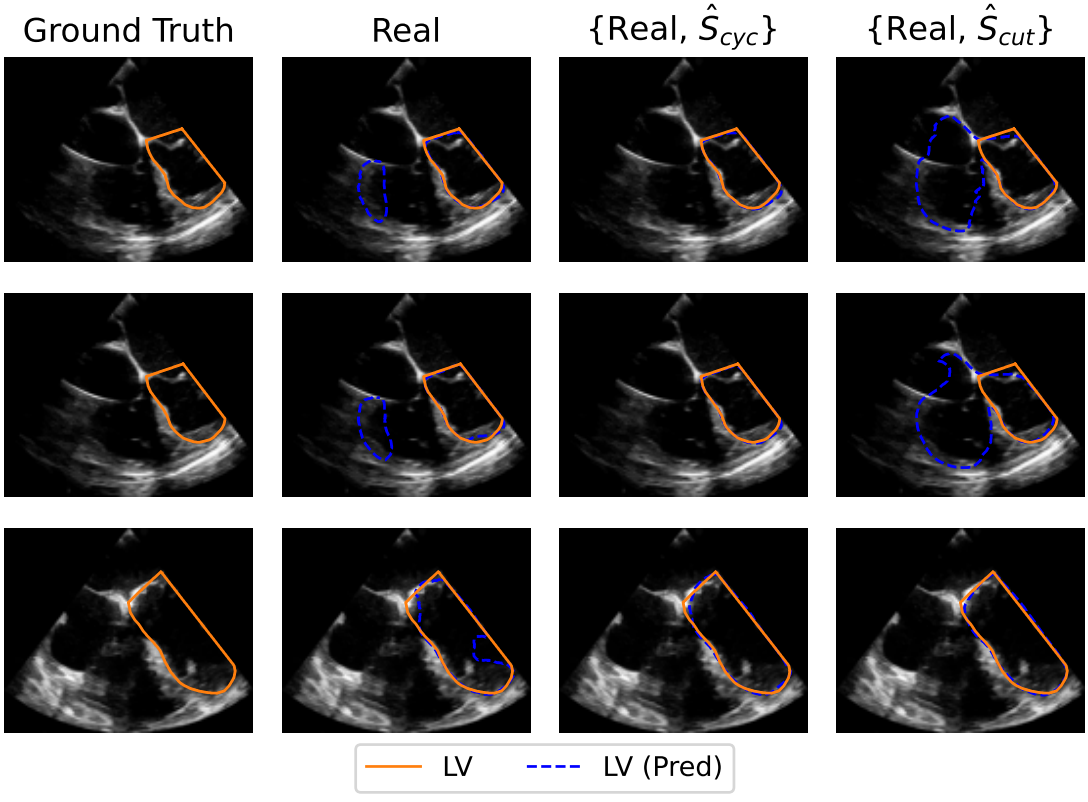


Figure 4.31: Segmentation Comparison of Difficult Cases. Example segmentations generated by nnUNet models trained on: purely real data (Real), real data mixed with synthetic images generated by a CycleGAN {Real, $\hat{S}_{cyc}$}, and real data mixed with synthetic images generated by the CUT method {Real, $\hat{S}_{cut}$}. (Note that the 'Real' cases also segment the LV in the first two rows. Best visualised on a screen.).

improves the dice score more often than $\hat{S}_{cut}$ in different mixes, and especially for the mixture that uses all available data $(+2.5$ vs $+5.9)$ which the figures are from.

Training on synthetic data alone resulted in poor segmentation performance, with Dice scores of 34.9 for $\hat{S}_{cut}$ and 20.9 for $\hat{S}_{cyc}$, highlighting a domain gap between the synthetic and real data. However, when synthetic data was combined with real samples during training, performance improved significantly, achieving Dice scores of 69.5 for $[R_{train} \cup \hat{S}_{cut}]$ and 72.9 for $[R_{train} \cup \hat{S}_{cyc}]$. Notably, despite $\hat{S}_{cyc}$ performing worse than $\hat{S}_{cut}$ in the synthetic-only setting, it provided a larger benefit when mixed with real data. This indicates that $\hat{S}_{cyc}$, while achieving a worse FID score than $\hat{S}_{cut}$, may introduce greater variability that complements the real data and supports improved generalisation when training is anchored to the real data distribution. In contrast, $\hat{S}_{cut}$ appears closer to the real distribution but may offer less additional variability, resulting in smaller gains when combined with real data.

One possible explanation for the poor standalone performance of the synthetic datasets relates to the use of unpaired I2I translation, which optimises visual similarity to the target domain rather than the preservation of anatomical structure specifically. When using I2I methods, translated images may contain subtle distortions or inconsistencies that, although not immediately apparent, could be detrimental to segmentation performance when no real data is present. This effect may be exacerbated by the limited size of the real dataset used to train the generative models, which constrains the range of anatomical and imaging variation that can be learned. These results suggest that the greater diversity introduced by $\hat{S}_{cyc}$ and $\hat{S}_{cut}$ may help the model handle more challenging cases.

When the generative model metrics are considered alongside the segmentation results, human-judged realism is shown to be a better predictor of augmentation impact than FID. $\hat{S}_{cyc}$ produced a higher delta score across more dataset mixes and resulted in the largest overall Dice score. This finding aligns more closely with the quiz results. In particular, the accuracy results shown in Table 4.5 indicate that an average researcher is able to make this assessment at a level comparable to that of an expert.

This disagreement between evaluation metrics is not unexpected. Although FID is often reported to correlate well with human perception, there is doubt as to whether this holds for medical imaging data [129]. The Inception network used to compute FID is pre-trained on natural scene images. Unlike natural scenes, echocardiographic images may contain diagnostically relevant information within noise and speckle patterns. As a result, FID may be less meaningful when applied to TEE images.

4.4 Conclusion

In this study, `HeartToEcho` was developed to extract standard TEE views and generate corresponding pseudo-images. Image translation methods for transforming pseudo-images into synthetic TEE images were compared and evaluated using the FID score and a human perception quiz. It was then shown that these synthetic images improved performance on an LV segmentation task by up to 10%. Finally, disagreement was observed between human perception and the FID score when evaluating the realism of synthetic TEE images, with human perception providing a better predictor of augmentation impact than the FID score.

Future work could involve exploring the use of these synthetic images for other tasks, such as augmenting ultrasound datasets beyond TEE. Another promising direction is the development of a quantitative image-to-image translation metric that is better suited to medical imaging data.

If any of you lacks wisdom, let him ask God, who gives generously to all without reproach, and it will given him

— James 1:5

5 Cross-Modality Generation using LoRA Diffusion

Contents

5.1 Introduction 78
5.1.1 Towards Diffusion-Based Approaches 79
5.1.2 Cross-Domain Adaptation via Low-Rank Fine-Tuning 81
5.1.3 Contributions 82
5.2 Methods 83
5.2.1 Elucidated Diffusion Model 83
5.2.2 Low-Rank Adaptation 85
5.2.3 MaskR^2 85
5.3 Experiments 86
5.3.1 Data 86
5.3.2 Image Generation 87
5.4 Results & Discussion 90
5.5 Conclusion 94

The previous chapter presented `HeartToEcho`, a method to extract standard TEE views from 3D atlases and convert them into realistic ultrasound images using unpaired deep generative modelling. This chapter extends `HeartToEcho` with a novel approach called `EchoLoRA`. `EchoLoRA` is a LoRA-based framework designed to efficiently adapt diffusion models for echocardiography image generation under limited data availability. The method builds upon prior work on synthetic TEE generation and introduces several advancements that address the scarcity of high-quality TEE datasets and the challenges

of domain adaptation. Throughout this chapter, the term “cross-domain” is used in the body text to refer to the transfer between the TTE and TEE acquisition domains; this is the same notion captured by the chapter’s published title “Cross-Modality Generation using LoRA Diffusion”.

The work presented in this chapter has been published and was accepted for a poster presentation at MICCAI ASMUS 2025:

Oladokun, E., *Ou, Y., Novikova, A., Kulikova, D., Thomas, S., Šprem, J., Grau, V.: From Transthoracic to Transesophageal: Cross-Modality Generation using LoRA Diffusion. In: *International Workshop on Advances in Simplifying Medical Ultrasound* (ASMUS 2025), Lecture Notes in Computer Science, vol. 16165, pp. 120–129, Springer Nature Switzerland (2025)*

5.1 Introduction

In Chapter 4, `HeartToEcho` was presented as a method for generating labelled synthetic datasets. These datasets come complete with labels because the synthetic images were themselves generated from the labels. Using a modest dataset of manufactured pseudo-images and real TEE images, unpaired I2I models were trained to create synthetic TEEs that improved performance on a downstream task when used to augment a real dataset. However, whilst `HeartToEcho` majorly automates the generation of usable synthetic datasets, it still relies on manual efforts at certain stages in its pipeline, for example, the selection of landmarks needed to generate a specific view and the choice of transforms that convert from idealised slices to pseudo-images. Furthermore, its training relied solely on TEE images. Due to the scarcity of TEE data, this approach is very limited and constitutes a bottleneck to progress as better architectures and training methods are developed. Finally, the generative methods used with `HeartToEcho` excel in two of the desired properties of a generative model shown in Fig. 5.1, namely, ’Fast Sampling’ and ’High Quality’ samples. However, they struggle to produce a variety of synthetic data and often succumb to mode collapse, a flaw where the generator produces the same sample regardless of the input. These issues significantly limit the ease-of-use and efficacy of `HeartToEcho` in the long run.

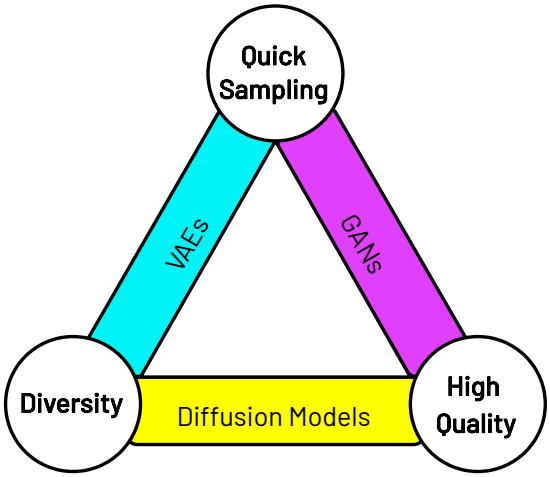


Figure 5.1: Desired Qualities of Generative Models. An ideal generative model would exhibit all three qualities—diversity, high quality, and quick sampling—simultaneously. In practice, these properties exist in tension; for instance, diffusion models typically achieve diverse and high-quality outputs but at the cost of longer sampling times during inference.

To further maximise the application of deep generative models in synthetic TEE dataset generation frameworks, a method that also has the following qualities is required:

- ☐ ***Ease-of-use.*** Minimise the pre-work that the user has to do. Examples of pre-work include hand-crafting the transformations to map idealised slices into pseudo-images

- ☐ ***Cross-domain.*** In practice, publicly available TEE data is too scarce to fully utilise the state of the art deep learning models. Therefore, a method that can extract information that is relevant to the current task from other domains is highly desirable.

5.1.1 Towards Diffusion-Based Approaches

Whilst the unpaired image-to-image models used in `HeartToEcho` (CycleGAN and CUT) provide a solid foundation for generating realistic synthetic data, their limitations motivate a transition towards diffusion-based models. These GAN-derived frameworks excel at producing sharp, high-quality images through a single forward pass, corresponding to the "Quick Sampling" quality in Fig. 5.1. However, the same design that enables fast inference constrains their ability to produce diverse outputs, often resulting in mode

collapse. This lack of diversity impairs the model's capacity to represent the full variability of cardiac anatomy and imaging artefacts observed in real TEE acquisitions.

Diffusion models, in contrast, offer a probabilistic formulation that captures a much richer data distribution. By modelling the generative process as a sequence of denoising steps, they inherently promote diversity and high fidelity, often surpassing adversarial models in sample realism and stability. Moreover, their iterative sampling allows multiple valid reconstructions given the same conditioning, reflecting the variability expected in noisy data like ultrasound images. Historically, this increase in quality and diversity came at a high computational cost. However, this is changing. Dhariwal et al. [89] noted that, with classifier guidance, their diffusion models matched the perceptual quality of BigGAN-deep using as few as 25 forward passes (GANs require a single forward pass), all while maintaining better coverage of the distribution. While conventional diffusion models [84, 85, 91] are computationally demanding and require many sampling steps, more recent formulations have begun to address these limitations. Among these, the Elucidated Diffusion Model (EDM) proposed by Karras et al. [92] provides a particularly effective balance between quality, diversity, and speed. EDM introduces a unified design space for solvers, time discretisation, and preconditioning, which enabled the authors to identify parts of the training scheme they could modify to reduce the number of denoising steps without compromising perceptual fidelity. In practice, this translates into faster inference while retaining the fine-grained detail essential for ultrasound synthesis. Furthermore, the improved solver design and skip connections enhance stability and controllability which are vital qualities when conditioning on anatomical structures or semantic masks. In this sense, EDM represents a natural evolution from adversarial approaches: it retains the diversity and high-quality sampling of diffusion models, yet alleviates their traditional computational burden.

Finally, as discussed in Section 2.4.2 *Conditioned Generation*, a defining feature of diffusion models is their flexibility in conditioning. Data from multiple modalities such as text and images can be used to steer the generation process. This presents an opportunity to improve the ease-of-use of the generative framework. Rather than conditioning on pseudo-images which require hand-crafted transformations to generate,

conditioning on the idealised slices or equivalently a segmentation mask would be much more desirable. Gilbert [115] compared generating synthetic TTE images using a CycleGAN and hand-crafted pseudo-images with generating straight from the idealised slice without any transformations. They observed the performance was notably worse and the images looked obviously fake. Collectively, these observations highlight the limitations of adversarial methods and the promise of diffusion-based frameworks for controllable synthetic echocardiography generation, thereby supporting a shift towards diffusion-based approaches.

5.1.2 Cross-Domain Adaptation via Low-Rank Fine-Tuning

Deep learning models, including diffusion models, typically require substantial data and compute resources to train effectively. Given the scarcity of TEE data, relying on TEE alone is a limited approach. A practical way to overcome this limitation is to extract useful information from other domains where data is more abundant. To this end, transfer learning is employed, whereby a model is first pretrained on one dataset using supervised, unsupervised, or self-supervised learning, and subsequently fine-tuned on another.

TTE data is relatively more abundant than TEE, so a natural approach is to pretrain on TTE and then fine-tune on TEE. Although the two modalities share several features, such as the ultrasound acquisition cone, speckle noise, and overall texture, there are also clear differences. For example, in Fig. 5.2, both scans image the four main chambers (RA, RV, LA, and LV), but the heart's orientation is inverted between the two modalities. Using transfer learning allows the model to learn features common to both domains during pretraining, reducing the domain gap that needs to be bridged during fine-tuning.

As discussed in Section 2.3 *Deep Learning*, several fine-tuning approaches exist, most of which involve updating all model parameters. However, as model sizes and sub-tasks increase, this approach scales poorly: each new task requires retraining the entire model and storing a full set of weights. It is also prone to catastrophic forgetting, where useful pretrained knowledge is lost during fine-tuning [130–132].

To address these issues and enable efficient domain adaptation, this work uses Low-Rank Adaptation (LoRA) [133]. LoRA adapts pretrained layers by inserting small sets

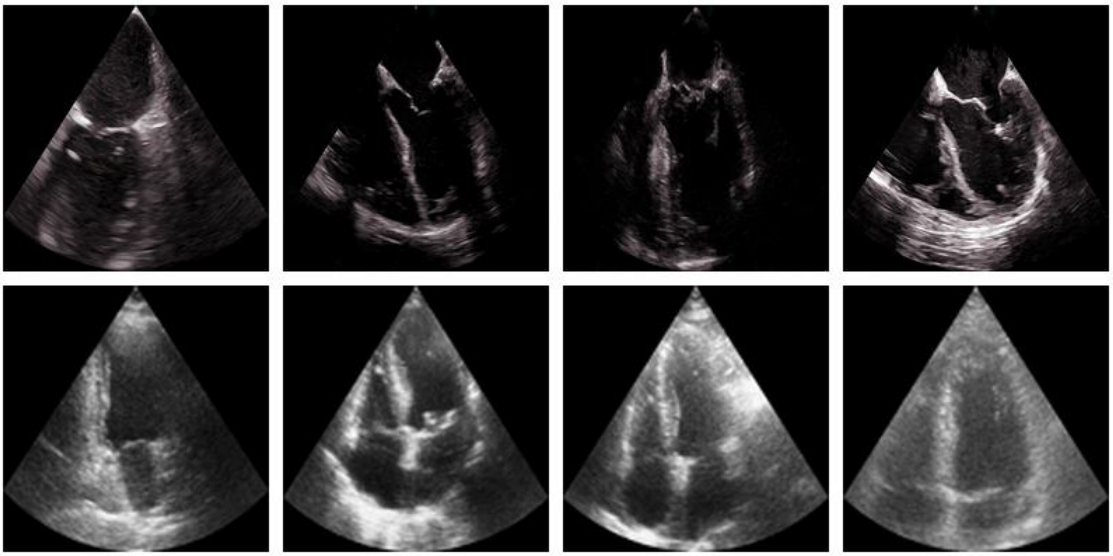


Figure 5.2: TEE & TTE. The first row shows TEE images of the ME4CH and ME2CH views. The second row shows acquisitions from the CAMUS dataset (more details in Sec. 3.2.2) and are of the A2CH and A4CH views.

of new trainable parameters into the model. During training, only these additional parameters are updated, while the original model weights remain frozen. This means that pretrained knowledge is preserved and that only a small subset of weights needs to be stored for each sub-task.

LoRA offers three main benefits: (i) it reduces the risk of overfitting when data is limited, (ii) it preserves the diversity and fidelity of the pretrained diffusion model, and (iii) it adds no extra latency during inference once the adapters are merged. In this work, LoRA provides a simple and efficient way to specialise a diffusion model pretrained on TTE data for use on TEE data. In doing so, it enables cross-domain use of knowledge learned from TTE while adapting effectively to the transesophageal domain.

5.1.3 Contributions

This chapter introduces `EchoLoRA`, a method developed to address the main limitations identified in `HeartToEcho`. `EchoLoRA` combines the EDM for its improved sampling efficiency, controllability, and fidelity with LoRA for its lightweight yet effective adaptability. Together, these choices provide a better foundation for developing a generative model that meets the desired qualities for synthetic TEE data generation—ease of use,

cross-domain learning, and high-quality, diverse outputs. The contributions of this chapter are as follows:

1. A novel LoRA-based method, `EchoLoRA`, is presented for efficient training on limited echocardiography data. This approach enables the generation of realistic synthetic TEE images that maintain strong adherence to image conditioning, thereby mitigating data scarcity in this domain.

2. A targeted adaptation strategy is developed to reveal the functional importance of layer groups within the diffusion model, identifying those most critical for effective adaptation to echocardiographic data.

3. A mask adaptation scheme, MaskR^2, is introduced to transform new semantic maps into the input format expected by pretrained base models. Despite this transformation, the resulting synthetic data substantially improve downstream multiclass segmentation performance, particularly for underrepresented anatomical classes.

5.2 Methods

5.2.1 Elucidated Diffusion Model

In a typical diffusion model, there are two processes: forward and reverse. The forward process gradually adds Gaussian noise to a data sample x_0 such that at time t the sample x_t has the following distribution:

$$q(\mathrm{x}_\mathrm{t} \mid \mathrm{x}_0) = \mathcal{N}\left(\mathrm{x}_\mathrm{t}; \mu_t \mathrm{x}_0, \sigma_t^2 \mathrm{I}\right). \tag{5.1}$$

where $\mu_t \mathrm{x}_0$ and $\sigma_t^2 \mathrm{I}$ are the mean and variance of the noised sample x_t at time t, produced by the forward diffusion process. If t is chosen to be large enough, the image becomes indistinguishable from random noise. For the reverse process, a denoiser is trained to iteratively remove the added noise.

In this work, EDM [92] is chosen. EDM is a unified framework that encompasses popular variants of diffusion models such as variance-preserving, variance-exploding[93], and DDIM[91], and highlights key design choices contributing to generative performance.

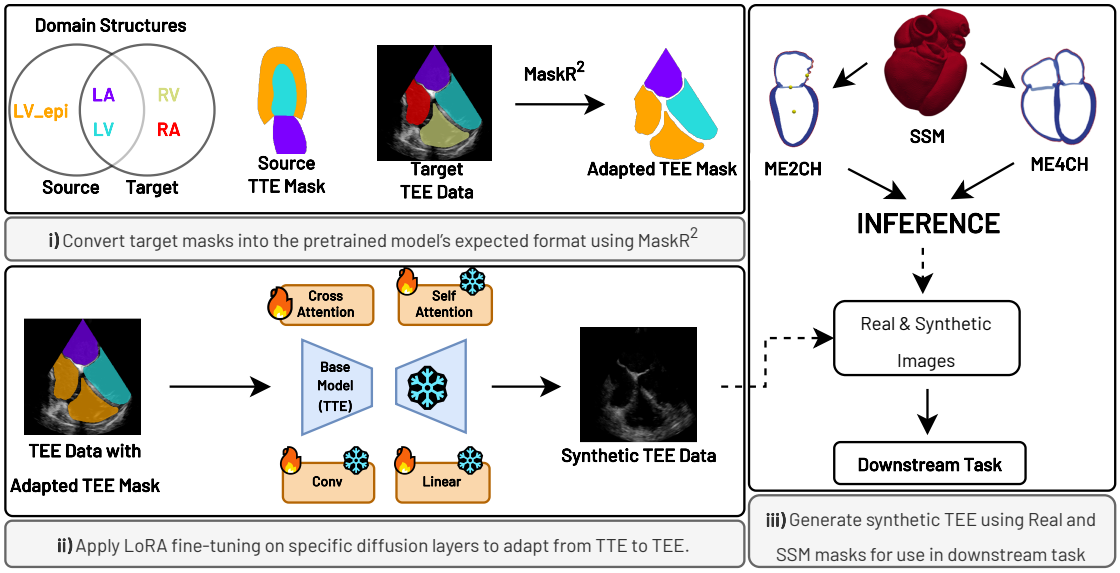


Figure 5.3: `EchoLoRA` TTE → TEE Pipeline. MaskR^2 is used to adapt the TEE Masks to the style expected by the TTE pretrained model. In this use case, the channel originally used for the left ventricular epicardium (LV$_{epi}$) conditioning is now responsible for generating the right-hand-side of the heart in TEE. Next, the base model is adapted in a targeted manner to generate synthetic TEE data. After training, inference is performed using real masks and masks sampled from SSMs to generate synthetic TEE datasets. Finally, the synthetic datasets are used to augment a real TEE dataset, which can be used for a downstream task.

Karras et al. [92] identify two major sources of error in the reverse step: inaccurate denoising by the neural network, and the discrete solver steps that follow incorrect trajectories during sampling. To mitigate these issues, they use a second-order Heun solver for the reverse step and propose a range of conditioning strategies. Given a dataset with variance σ^2_{data} and a noise schedule with variance $\sigma^2 = \sigma^2(t)$, EDM introduces the following preconditioning steps: scale the network input by $c_{\text{in}} = \frac{1}{\sqrt{\sigma^2_{\text{data}}+\sigma^2}}$; scale the skip connections by $c_{\text{skip}} = \frac{\sigma^2}{\sigma^2_{\text{data}}+\sigma^2}$; condition the network on noise using $c_{\text{noise}} = \frac{1}{4}\ln(\sigma)$; and scale the output by $c_{\text{out}} = \frac{\sigma \cdot \sigma_{\text{data}}}{\sqrt{\sigma^2_{\text{data}}+\sigma^2}}$. The resulting EDM loss is:

$$\mathbb{E}_{\sigma,x,n} \| \underbrace{F_\theta(c_{\text{in}}(\sigma)\cdot\tilde{\mathrm{x}}; c_{\text{noise}}(\sigma))}_{\text{network output}} - \underbrace{\frac{1}{c_{\text{out}}(\sigma)}\left(\mathrm{x} - c_{\text{skip}}(\sigma)\cdot\tilde{\mathrm{x}}\right)}_{\text{effective training target}} \|_2^2 \tag{5.2}$$

where $\tilde{\mathrm{x}} = \mathrm{x} + \mathrm{n}$, x is the clean training sample, $\mathrm{n} \sim \mathcal{N}(0, \sigma^2\mathrm{I})$, and F_θ is the network to be trained. They show that these reparametrisations significantly improve both training efficiency and the quality of the generated images.

5.2.2 Low-Rank Adaptation

LoRA [133] is an efficient method for adapting models to new tasks. Given a pretrained model with weight matrix $W_0 \in \mathbb{R}^{d\times k}$, LoRA decomposes the fine-tuning update $\triangle W$ into a product of two low-rank matrices, BA, where $A \in \mathbb{R}^{r\times k}$ and $B \in \mathbb{R}^{d\times r}$, with $r \ll \min(d,k)$. During training, the base model remains frozen and only A and B are updated. For an input x, the forward pass becomes $h = W_0x + \frac{\alpha}{r}BAx$, where the scaling factor α adjusts the influence of the adapter relative to the base model. A key advantage of LoRA adapters is that they introduce no additional inference-time latency, as they can be merged with the main model after training. LoRA was leveraged to efficiently adapt the base model to a different form of echocardiography.

5.2.3 MaskR^2

One-hot encoding is the standard way to condition diffusion models on semantic masks: a 2D label map of size $H \times W$ becomes an $N \times H \times W$ tensor, where N is the number of classes. However, one-hot encoding rigidly fixes the number of conditioning channels the model expects, forcing an advance decision on how many classes will ever be needed. If a model pretrained on a dataset with class set X is to be fine-tuned on a new dataset with a different class set Y, either channels will be wasted, or the new classes cannot be accommodated without architectural changes to, or retraining of, the base model. To solve this, MaskR^2 was proposed. MaskR^2 remaps any new-dataset labels Y into the pretrained model's label space X using just three simple operations - *Identity*, *Reduce*, and *Repurpose* - so that the condition architecture can remain unchanged. Concretely:

1. Identity: Leave labels in $X \cap Y$ unchanged
2. Reduce: If $|Y| > |X|$, merge extra labels in $Y \setminus X$ into 'super-classes'
3. Repurpose: Assign (super-)classes in $Y \setminus X$ to the classes in $X \setminus Y$

For example, suppose a base model is trained on labels for the LA, LV, and left ventricular epicardium (LV_{epi}), and is being adapted to a new dataset containing labels for LA, LV, RA, RV, MaskR^2 maps as follows: $\{LA \rightarrow LA, LV \rightarrow LV, \{RA, RV\} \rightarrow LV_{epi}\}$. Figure 5.3

part i) illustrates this mapping with real images. If $X \setminus Y = \emptyset$, no pretrained channels are free to be repurposed; in this case an additional conditioning channel can be appended to the input, its weights initialised either from scratch or by copying an existing pretrained channel, and the model finetuned to make use of it. This recovers the new classes at the cost of a small change to the conditioning architecture, which MaskR^2 otherwise avoids.

5.3 Experiments

5.3.1 Data

For training and evaluation, both the TEE-B and CAMUS datasets were utilised.

CAMUS Dataset

From the CAMUS dataset, 2,000 annotated frames were extracted (500 patients × {A4CH, A2CH} × {ED, ES}). Using a patient-wise split, such that all frames from a given patient were assigned to the same partition, 1,400 frames (70%) were allocated for training, 300 frames (15%) for validation, and 300 frames (15%) for testing.

As part of pre-processing, all CAMUS images and masks were resized to 224×224 pixels, using bilinear interpolation for images and nearest-neighbour interpolation for masks, and the images were converted to grayscale. As the objective is image generation, standard data augmentations such as random rotations, flips, and elastic deformations were not applied. Such augmentations can produce physiologically implausible echocardiographic configurations, for example flipped ventricles, distended boundaries, or inconsistent speckle patterns, which the model may otherwise learn to generate.

TEE Dataset

This study uses the TEE-B dataset, introduced in Section 3.1, which comprises 288 image–mask pairs drawn from 71 mid-esophageal two- and four-chamber (ME2CH/ME4CH) videos. Each frame is annotated for the LA and LV, and for the RA and RV when visible. The majority of annotations ($\sim$ 70%) were provided by two expert cardiologists with daily echocardiography experience, with the remaining annotations produced by me under their supervision. A total of 196 TEE images were allocated for training, 40 for

validation, and 52 for testing, with an even distribution across views and cardiac phases. The disparity in dataset size between the CAMUS and TEE-B datasets reflects real-world constraints on TEE data availability.

In addition, semantic masks for the ME2CH and ME4CH views were sampled using `HeartToEcho`, which provides planes corresponding to standard TEE and TTE views from 3D SSMs of the heart.

5.3.2 Image Generation

Figure 5.3 illustrates `EchoLoRA`, the proposed pipeline and Table 5.1 shows the model, training, and data configuration. The base model is an EDM trained at a resolution of 224×224, using a UNet with a depth of 3 and self-attention and cross-attention layers at specific stages. The channel dimensionality follows $64 \times [1, 2, 4]$ across the three stages. Within each stage, features pass through three residual blocks. Each block applies two 3×3 convolutions with channel-wise root-mean-square normalisation and SiLU activations, and receives diffusion-timestep information through a learned MLP that maps a 512-dimensional time embedding to per-channel scale and shift parameters. Spatial resolution is then halved on the encoder side (or doubled on the decoder side) via a pixel-unshuffle (or pixel-shuffle) operation combined with a 1×1 convolution, rather than strided convolutions or pooling. To improve efficiency, cross-attention is applied only in layers 1 and 2, injecting conditioning information early in the generation process. To improve global structural understanding, self-attention is added at layer 3 and the bottleneck, where the receptive field is largest and features are most abstract. Exponential moving average is used during training. The resulting UNet architecture contains 21.79 million parameters and serves as the base model.

Firstly, the base model is pretrained on the CAMUS dataset, after which the model weights are frozen. LoRA adapters are then attached to facilitate efficient adaptation to the target dataset. To this end, a targeted adaptation strategy is employed. The model's layers are categorised into five groups: *Cross-Attention*, *Self-Attention*, *Convolution*, *Linear*, and *Other*. LoRA adapters are subsequently attached independently to each group, as well as to their combinations, in order to identify which subsets contribute

Table 5.1: Image Generation Configuration for Pretraining and Finetuning. The hyperparameters and configurations used to train the base diffusion model and fine-tune the LoRA parameters. Feature dimensions and attention layers are applied symmetrically across the downsampling and upsampling blocks. *Acronyms:* Self-attention (SA), Cross-attention (CA).

Image Generation								
	Model			Training			Data	
	Pretrain	Fine-tune		Pretrain	Fine-tune		Pretrain	Fine-tune
Feature Dimensions	{64. 128, 256}	{64. 128, 256}	LR	1e-4	1e-3	Type	TTE	TEE
SA	{×, ×, ✓}	{×, ×, ✓}	Scheduler	Cosine	Cosine	Train	1400	194
CA	{✓, ✓, ×}	{✓, ✓, ×}	Steps	170,000	100,000	Valid	300	40
Trainable Parameters	21.79M	Variable	Batch Size	12	4	Test	300	52

most effectively to adaptation. For all adapters, $\alpha = r = 16$ is used. Cross-attention layers are always trained, as they provide a lightweight mechanism for incorporating conditioning signals while allowing different spatial locations to respond differently to the same global context (e.g. the current timestep), making them a practical component to adapt when transferring to new inputs. Following a hyperparameter search, all adapted models are trained using an identical setup: 100,000 training steps, 64 diffusion time steps, classifier-free guidance weight of 5, an initial learning rate of 1×10^{-3} with cosine decay, and a batch size of 4. As a baseline, a model with the same architecture as the base model is also trained from scratch on the TEE dataset.

Evaluation

At inference time, masks from the images in the test set are used to sample from all model configurations, which are then evaluated using common image quality metrics, namely FID, LPIPS, and SSIM. Notably, FID and LPIPS are primarily designed for evaluating natural scene images rather than medical echocardiograms, where features such as speckle noise and subtle texture variations are important. Nevertheless, these metrics are commonly used and reported in related literature and are therefore included here, subject to the above caveats.

In addition to performing inference using real masks, ME4CH and ME2CH masks were also extracted using the SSM dataset and `HeartToEcho` to generate synthetic

echocardiograms. This allows the similarity of the masks to the real data distribution to be assessed and enables the adapted models to be stress-tested using masks that were not observed during training.

Next, the synthetic images from each fine-tuned model were used to augment real datasets for a multiclass segmentation task, thereby enabling an assessment of their downstream utility. For each generative model, the masks in the TEE-B train set were used to sample 196 synthetic images. Next, 196 SSM masks were used to sample another 196 synthetic images. Finally, both of synthetic datasets were independently added to a copy of the real train dataset, resulting in two augmented datasets of sizes 386. This process was repeated for all generative models and nnUNet models were trained using the augmented datasets. Furthermore, an nnUNet model was trained on the real images from the train split alone to serve as a baseline. All models were evaluated on the same held-out validation and test splits containing real data only. Notably, all data augmentations provided by nnUNet were retained, as the unique benefits of the synthetic images were of interest.

Four Dice-based metrics are reported, which differ in how statistics are aggregated across the dataset. The following notation is defined:

$i \in \{1, \dots, N\}$	index for images (total N images)
$c \in \{1, \dots, C\}$	index for segmentation classes (total C classes)
$\mathrm{TP}_{i,c}$	true positive pixel count for class c in image i
$\mathrm{FP}_{i,c}$	false positive pixel count for class c in image i
$\mathrm{FN}_{i,c}$	false negative pixel count for class c in image i
$n_{i,c}^{\mathrm{ref}}$	:= number of reference (ground-truth) pixels for class c in image i
$n_{i,c}^{\mathrm{pred}}$	:= number of predicted pixels for class c in image i

First, the Global Dice metric, which treats the entire dataset as a single segmentation problem by pooling true positives, false positives and false negatives across every class and image before computing the Dice score:

$$\mathrm{Dice}_{\mathrm{global}} = \frac{2 \sum_{i,c} \mathrm{TP}_{i,c}}{2 \sum_{i,c} \mathrm{TP}_{i,c} + \sum_{i,c} \mathrm{FP}_{i,c} + \sum_{i,c} \mathrm{FN}_{i,c}}.$$

This metric reflects overall pixel-level performance. Second, the Class-Weighted Dice, which computes Dice scores at the image–class level and then weights each score by the number of ground-truth pixels for that class before averaging across the dataset:

$$\text{Dice}_{\text{weighted}} = \frac{\sum_{i,c} n_{i,c}^{\text{ref}} \, \text{Dice}_{i,c}}{\sum_{i,c} n_{i,c}^{\text{ref}}}, \qquad \text{Dice}_{i,c} = \frac{2\,\text{TP}_{i,c}}{2\,\text{TP}_{i,c} + \text{FP}_{i,c} + \text{FN}_{i,c}}.$$

This weighting reduces the influence of rare classes while still accounting for per-class segmentation quality. Third, Per-class Dice, where each class is evaluated independently by aggregating statistics across all images and computing a separate global Dice score for each class:

$$\text{Dice}_{c} = \frac{2 \sum_i \text{TP}_{i,c}}{2 \sum_i \text{TP}_{i,c} + \sum_i \text{FP}_{i,c} + \sum_i \text{FN}_{i,c}}.$$

Finally, the Average Dice, which computes a Dice score at the image–class level, averages those scores across the images in which a given class is present in the ground truth, and then averages the resulting per-class means across all classes:

$$\text{Dice}_{\text{avg}} = \frac{1}{C} \sum_{c=1}^{C} \overline{\text{Dice}}_{c}, \qquad \overline{\text{Dice}}_{c} = \frac{1}{N} \sum_{i} \text{Dice}_{i,c}.$$

Together, these metrics provide complementary pixel-level ($\text{Dice}_{\text{global}}$), prevalence-weighted ($\text{Dice}_{\text{weighted}}$), per-class ($\text{Dice}_{c}$) and class-balanced ($\text{Dice}_{\text{avg}}$) views of the models' performance, and are useful for identifying failure modes that may be obscured by any single aggregate metric.

5.4 Results & Discussion

Table 5.2 compares the different adapter configurations in terms of both image-quality metrics and downstream segmentation performance under real-mask and SSM-mask conditioning. Image quality is first observed to show a weak correlation with augmentation impact, suggesting that the two are not tightly coupled. That is, images that score better according to these metrics do not necessarily translate into greater usefulness for a segmentation task. Furthermore, reducing the number of trainable parameters is observed to have only a weak effect on image quality when the adapted models are

Table 5.2: Generation & Multiclass Segmentation Results. This table summarises the performance of the LoRA-adapted models on both image-quality metrics and a downstream multiclass segmentation task. "All Weights" refers to the model trained purely on TEE images. Each adapter updates only the specified layer groups—Cross-attention (CA), Self-attention (SA), Convolution (Conv), and Linear—via LoRA. "Mask Source" indicates whether the synthetic echos were generated with real TEE masks or SSM masks. The "Per-class Dice" column reports the Dice score for each chamber, shown here as the delta relative to the baseline model. **Bold** highlights the overall best scores, and colored entries mark the top performer for each individual class.†Best FID Score in [134]. ‡Scores achieved from nnUNet trained on purely real data.

Generative Model			Image Quality Metrics			Segmentation Scores		
Trainable Groups 🔥	Trainable Params (M/%)	Mask Source	FID (↓)	LPIPS (↓)	SSIM (↑)	Global Dice (↑)	Class-weighted Dice (↑)	Per-class Dice (↑) {LA, LV, RV, RA}
All Weights	21.79/100%	Real	155.7	0.31	0.55	89.05	88.78	{+0.22, +1.75, +5.82, +2.97}
		SSM	176.5	-	-	88.61	89.41	{+0.63, +1.46, +3.87, +2.56}
{CA, Linear SA, Conv}	2.70/11%	Real	117.7	0.31	0.54	88.53	88.79	{+0.07, +1.15, +4.48, **+3.49**}
		SSM	120.3	-	-	88.21	89.06	{+0.97, +0.56, +3.53, +2.91}
{CA, Conv}	2.13/9%	Real	134.5	0.31	0.55	88.14	88.75	{-0.11, +1.05, +4.23, -0.14}
		SSM	160	-	-	88.28	89.77	{**+1.03**, +1.15, +3.08, +0.68}
{CA, Linear}	0.69/3%	Real	154.4	0.33	0.53	89.40	89.72	{-0.11, +3.43, +3.64, +0.69}
		SSM	164.3	-	-	**89.60**	**90.14**	{+0.92, +2.62, +5.90, +3.09}
{CA, SA}	0.51/2%	Real	152.3	0.35	0.48	89.38	89.51	{+0.56, +3.24, +2.70, +0.28}
		SSM	151.4	-	-	87.56	88.45	{+0.83, +0.15,+1.90, -0.57}
Baselines	-	-	188†	-	-	87.16‡	88.00‡	{94.78, 86.65, 70.83, 84.71}

compared with the 'All-Weights' model. This indicates that the adapters are able to leverage prior knowledge learned by the base model from TTE data to generate TEE datasets using very few parameters.

Next, most adapters are observed to generalise reasonably to out-of-distribution SSM masks. Under SSM conditioning, FID increases only marginally for several configurations, and for the {CA, SA} adapter it remains unchanged; the 'All Weights' and {CA, Conv} configurations, however, show larger increases of roughly 13% and 19% respectively. These degradations are modest for most adapters and encouraging overall, particularly given that a perfect FID is unattainable when comparing across different data distributions. In Fig. 5.4, synthetic echocardiograms generated by the adapters are compared using in-distribution real masks from the held-out test set and out-of-distribution SSM masks that were not seen during training. Despite using only up to 11% of the original parameters, images with high visual fidelity are produced, and the SSM-conditioned outputs remain anatomically plausible despite the domain gap.

Finally, all models except {CA, SA} are observed to generate the right side of the heart

Table 5.3: Average Segmentation Dice. For each chamber the Dice is computed per patient and averaged over the patients in which the chamber is present in the reference; the four per-chamber means are then averaged into a single **Average Dice**. Per-class entries are shown as the delta relative to the baseline. **Bold** marks the best Average Dice; colored entries mark the best delta per chamber. ‡Scores from nnUNet trained on purely real data.

Generative Model			Segmentation Scores	
Trainable Groups 🔥	Trainable Params (M/%)	Mask Source	Average Dice (↑)	Per-class Average Dice (↑) {LA, LV, RV, RA}
All Weights	21.79/100%	Real	82.09	{+0.20, +1.00, +2.73, -2.47}
		SSM	84.38	{+0.73, +0.93, +6.85, +2.10}
{CA, Linear SA, Conv}	2.70/11%	Real	81.96	{-0.03, -0.08, +2.91, -1.86}
		SSM	83.97	{+1.09, +0.11, +6.82, +0.94}
{CA, Conv}	2.13/9%	Real	82.57	{-0.28, +0.34, +4.68, -1.36}
		SSM	83.89	{**+1.14**, +0.85, +8.23, -1.57}
{CA, Linear}	0.69/3%	Real	82.17	{-0.30, +3.31, +4.51, -5.74}
		SSM	**84.94**	{+0.87, +1.74, +7.42, **+2.81**}
{CA, SA}	0.51/2%	Real	82.05	{+0.59, +2.93, +3.32, -5.55}
		SSM	82.63	{+0.89, -0.24, +4.71, -1.75}
Baselines	-	-	81.73‡	{93.94, 87.02, 70.92, 75.02}‡

well, including valves. In the SSM-conditioned images, the valves appear less pronounced, as no gaps exist between the RA and RV in the masks used for conditioning. As a result, the model may not be prompted to generate a valve between these structures.

On the downstream task, all augmented datasets are observed to outperform the baseline trained solely on real images, confirming that segmentation benefits from the inclusion of synthetic images. The {CA, Linear} configuration is found to produce the largest improvement in overall performance, as well as the strongest performance when each mask source is considered independently. The 'Linear' group mainly consists of MLP layers, suggesting that these layers are the most important for learning features that are most useful for segmentation. For all but one segmentation model, the class-weighted Dice is observed to exceed the global Dice, indicating improved performance on the more prevalent classes.

When the per-class Dice is examined, which computes a separate global Dice score for each chamber, the underrepresented right-heart structures are generally found to benefit more from synthetic augmentation than the left. For example, improvements

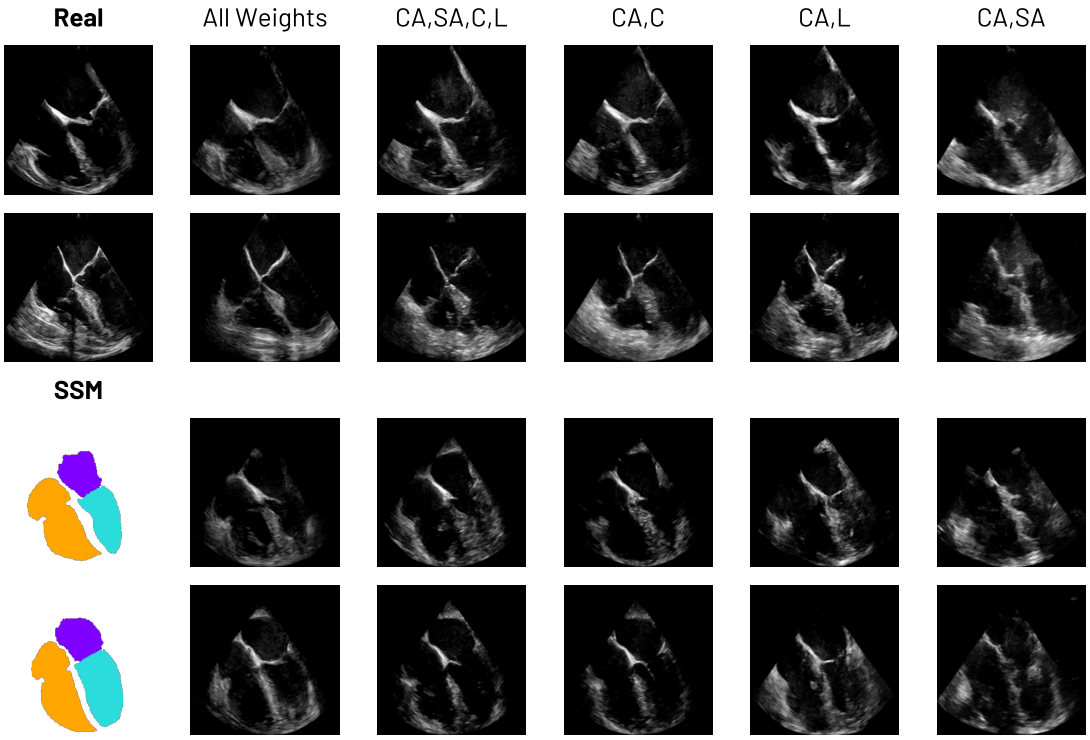


Figure 5.4: Synthetic TEE Images. This figure shows how the adapted models perform at inference using real masks from the test set and SSM masks. In all cases, the generative model sees the right side of the heart as one structure due to MaskR^2. 'All Weights' represents a generative model trained on TEE images from scratch. The layer groups are Cross-attention (CA), Self-attention (SA), Convolution (C), and Linear (L).

for the RV range from +1.9 to +5.9 Dice points, whereas improvements for the LV range from +0.15 to +3.43. Crucially, these gains are observed despite MaskR^2 collapsing the RA and RV into a single super-class, demonstrating that the adapted models are capable of generating synthetic images that enhance right-side segmentation without explicitly distinguishing between these two chambers.

Table 5.3 complements the per-class scores by reporting the per-patient averaged per-class Dice, with the four per-chamber means averaged into an overall Average Dice. Under this aggregation the right-ventricle gains highlighted above are also strong (up to $+8.23$ Dice points), and every adapter again exceeds the real-only baseline (81.73), with {CA, Linear} under SSM masks reaching the highest overall score of 84.94.

Overall, the results indicate that LoRA adapters are able to harness information learned from the more prevalent TTE data and build upon it to generate useful TEE images using as few as approximately 510,000 parameters. Furthermore, MaskR^2 is

shown to effectively map new mask formats into the base model's expected conditioning space without compromising downstream performance.

5.5 Conclusion

A lightweight, data-efficient pipeline, termed `EchoLoRA`, is proposed to adapt a TTE-trained diffusion model to TEE via LoRA using minimal TEE data. By conditioning on semantic masks, and by using MaskR^2 to remap novel mask formats into the model's original channel space, fine-grained control over anatomy is achieved even when new structures or mask conventions arise. The resulting synthetic TEE images capture the expected anatomical structures in the corresponding views and are of sufficient appearance to be useful as training data, though their human-judged perceptual realism is not formally evaluated. When used to augment real TEE cases, improvements in multiclass segmentation Dice score are observed, with the largest gains occurring for underrepresented right-heart chambers. In this way, the practical use of pretrained diffusion models for specialised echocardiographic imaging is demonstrated.

At the beginning of this chapter, the aim was to extend `HeartToEcho` by improving its *Ease-of-use* and *Cross-domain* capabilities. `EchoLoRA` builds on the original framework by addressing both of these aspects. In terms of *Ease-of-use*, `EchoLoRA` removes the need for the manual tuning of image transformation parameters to transform idealised anatomical slices to pseudo-images, while still producing higher-quality synthetic TEE images. Even the weakest-performing configuration according to FID, the All-Weights model, achieved a score of 177, outperforming the best-performing I2I translation baseline $\hat{G}_{cut}$, which achieved a score of 188. This indicates that the goal of improving *Ease-of-use* relative to `HeartToEcho` is achieved.

With respect to *Cross-domain* capability, `EchoLoRA` is not limited to training on TEE data alone. Instead, information learned from a related imaging domain, TTE, is effectively reused to generate higher-quality synthetic TEE images. This reduces reliance on large TEE datasets and addresses a key limitation identified earlier in this chapter.

Overall, this chapter shows that `EchoLoRA` provides a practical extension to `HeartToEcho` for synthetic echocardiographic image generation. By improving *Ease-of-use*, enabling

Cross-domain learning, and demonstrating clear downstream benefits, `EchoLoRA` offers a more flexible and reusable framework for generating synthetic TEE data. As the SSM masks used in this work are publicly available, the approach can be readily adopted by others. Finally, the framework is modality-agnostic and can be applied to other imaging domains where mask-conditioned synthesis is required.

For where your treasure is, there your heart will be also.

— Matthew 6:21

6

One-step Echocardiogram Generation via Latent Video Flow Matching

Contents

6.1 Introduction 98
6.1.1 Contributions 100
6.2 Methods 100
6.2.1 Linear Flow 100
6.2.2 Mean Flow 101
6.2.3 EchoLVFM 102
6.3 Experiments 107
6.3.1 Data 107
6.3.2 Training 108
6.3.3 Run Configurations 108
6.3.4 Evaluation 110
6.4 Results 112
6.4.1 VAE 112
6.4.2 Video Generation 113
6.5 Discussion 117
6.5.1 VAEs 117
6.5.2 Video Quality 119
6.6 Conclusion 120

The previous chapter introduced `EchoLoRA`, a LoRA-based framework for generating synthetic echocardiographic images through semantic mask conditioning. While `EchoLoRA` enables the generation of high-quality images across the cardiac cycle, it treats

each frame independently and does not explicitly model temporal relationships between frames. This chapter extends the scope of generative echocardiography from images to videos by introducing `EchoLVFM`, a latent video generation framework designed to produce temporally coherent echocardiogram sequences under realistic data constraints. By modelling both spatial and temporal structure, `EchoLVFM` builds upon the principles of efficient, data-aware/ease-of-use generation introduced in the previous chapter and adapts them to the video setting.

6.1 Introduction

Although an image, especially key frames like ED and ES frames, can provide insight into a patient's cardiac health, medical professionals rely on sequences rather than isolated images in practice. Unlike single frames, sequences capture the temporal dynamics of the heart, providing additional clinically relevant information. Over time, the contraction and relaxation of the ventricles and atria can be observed, and the resulting chamber and valvular dynamics make echocardiographic sequences especially useful for diagnosing pathologies such as valve regurgitation [23]. Furthermore, sequences enable the calculation of global functional metrics such as the EF which remains the most widely used global index of left-ventricular systolic function and is central to the diagnosis of many forms of heart failure [135–137].

Chapter 5 presented `EchoLoRA` which generates synthetic echos via semantic mask conditioning. While `EchoLoRA` is not restricted to generating images from a specific phase of the cardiac cycle and could, in principle, be used to generate a full cardiac cycle, independently generated frames lack the temporal smoothness and coherence that characterise real echocardiographic sequences. Generating temporally consistent videos therefore requires methods that explicitly learn both spatial and temporal structure. Architectures such as 3D CNNs [138] and spatiotemporal UNets [139] are capable of this, but tend to be computationally expensive, especially methods that rely on 3D convolutions. Furthermore, processing videos rather than images substantially increases memory and compute requirements as a forward pass on a video with T frames is effectively equivalent to processing a batch of T images.

To alleviate the computational load, latent image and video generation models are often used. This approach consists of downsampling the data from its original form into the latent space of a pre-trained Variational Autoencoder (VAE) and performing the generation in this space. Once a latent sample is produced, it is then decoded back to its original space by decoding. Downsampling in this manner saves considerable memory, particularly for attention-based models that rely on self- and cross- attention which have memory requirements that scale quadratically with spatial resolution.

The ability to generate echocardiograms whilst controlling clinical parameters like the EF is highly desirable. Such control facilitates creating counterfactuals at a personal level, thus aiding in providing personalised healthcare. Furthermore, specialist training can be enhanced as educators could generate under-represented examples, thus allowing them to become more familiar, in an educational setting, with examples they would rarely encounter in practice. Generating rare cases is also useful for artificially balancing datasets, thereby tackling localised data scarcity in datasets and enabling improved generalisation of models trained for downstream tasks.

While diffusion models are the common choice for latent video generation [99, 140, 141], flow matching has recently emerged as a compelling alternative. Flow matching combines the stability and generative framework of diffusion with the efficiency of normalising flows, allowing it to reduce sampling cost while maintaining realism. More recently, an approach that requires just one inference step was proposed by Geng et al. [142]. Flow Matching's simplicity, and ability to yield high-quality outputs in fewer steps make it an appealing option for video generation.

Despite these advantages, applications of flow matching to echocardiography remain limited. Yazdani et al. [104] apply linear flow matching to echocardiography but restrict generation to key frames, leaving the temporal dynamics of echocardiograms unaddressed. [14] extend flow matching to latent video generation, but their approach is effectively frame animation conditioned on EF, with input videos temporally normalised by cropping and resampling to a fixed number of frames.

In practice, echo data is highly heterogeneous. Sequences vary substantially in length, frame rate, and also quality depending on acquisition conditions. For example,

the CAMUS dataset used in this chapter contains videos depicting only ED to ES in as few as ten frames. Existing approaches [14, 100] simplify modelling by enforcing a fixed video length through temporal cropping or resampling. While effective for longer sequences, this strategy excludes shorter videos and discards potentially useful temporal information, limiting applicability to real-world datasets.

This motivates the development of generative models that can accommodate variable-length echocardiogram sequences without extensive preprocessing. By explicitly supporting partially observed sequence conditioning, the proposed approach can operate on heterogeneous data and enables a broader set of tasks, including frame animation, and video completion, rather than being restricted to single-frame conditioning.

6.1.1 Contributions

In this chapter, `EchoLVFM` is proposed to address the challenges of temporally coherent echocardiogram video generation under realistic data constraints. The contributions of this chapter are as follows:

1. A one-step latent video flow matching framework for controllable echocardiogram generation is introduced, enabling efficient synthesis of temporally coherent echo videos conditioned on a global parameter.

2. A variable-length video generation strategy is proposed that supports echocardiogram sequences of differing durations and frame rates, reflecting the heterogeneity of real-world data.

3. A masked video conditioning scheme is introduced, enabling tasks beyond frame animation, such as video completion from partially observed sequences.

6.2 Methods

6.2.1 Linear Flow

Given a data distribution $x \sim p_{\text{data}}$ and a Gaussian noise distribution $\epsilon \sim \mathcal{N}(0, I)$, flow matching aims to learn a velocity field $v(x_t, t) = \frac{dx_t}{dt}$ that defines how a latent

variable x_t evolves as the timestep t moves from 0 to 1. The trajectory transports a clean sample at $t = 0$ into a noise sample at $t = 1$; sampling reverses the trajectory to convert noise back into data.

For linear flow, the path between x and ϵ is defined as a straight line:

$$x_t = (1 - t)\, x + t\, \epsilon. \tag{6.1}$$

Differentiating this with respect to t gives an analytical expression for the velocity field:

$$v(x_t, t) = \epsilon - x. \tag{6.2}$$

Given x_t defined as above and a network $v_\theta(x_t, t)$ parameterised by θ, the training objective is

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{\epsilon,x,t}\left\| v_\theta(x_t, t) - (\epsilon - x) \right\|_2^2. \tag{6.3}$$

Once trained, $v_\theta(x_t, t)$ can be used for inference by solving the differential equation $\frac{dx_t}{dt} = v_\theta(x_t, t)$. Notably, inference in flow matching is deterministic, similar to diffusion models such as DDIM, but distinct from stochastic variants like DDPM. The ODE is typically discretised into N steps and solved using a numerical method such as Euler's method:

$$x_{t+\Delta t} = x_t + \Delta t\, v_\theta(x_t, t). \tag{6.4}$$

6.2.2 Mean Flow

An alternative formulation, MeanFlow [142], redefines the modelling objective by focusing on the average velocity over a time interval $[r, t]$ rather than the instantaneous velocity:

$$u(x_t, r, t) = \frac{1}{t - r} \int_r^t v(x_\tau, \tau)\, d\tau. \tag{6.5}$$

From this equation can be derived the Mean Flow Identity

$$(t - r)u(x_t, r, t) = \int_r^t v(x_\tau, \tau)\, d\tau. \tag{6.6}$$

Differentiating Eq. 6.6 with respect to t gives the relationship

$$u(x_t, r, t) = v(x_t, t) - (t - r)\frac{d}{dt}u(x_t, r, t), \tag{6.7}$$

where

$$\frac{d}{dt}u(x_t, r, t) = \frac{dx_t}{dt}\partial_x u + \frac{dr}{dt}\partial_r u + \frac{dt}{dt}\partial_t u = J \cdot [v(x_t, t), 0, 1], \tag{6.8}$$

and J denotes the Jacobian. Substituting $\frac{dx_t}{dt} = v(x_t, t) = \epsilon - x$ (from Eq. 6.2) yields the effective regression target

$$u_{\text{tgt}} \;=\; v(x_t, t) - (t - r)(v(x_t, t)\partial_x u_\theta + \partial_t u_\theta) \;=\; (\epsilon - x) - \mathcal{J}(x_t, r, t), \tag{6.9}$$

where $\mathcal{J}(x_t, r, t) = (t-r)(v(x_t, t)\partial_x u_\theta + \partial_t u_\theta)$ is the *Mean Flow correction term* arising from the Mean Flow Identity (Eq. 6.6). A network $u_\theta(x_t, r, t)$ is then trained to minimise

$$\mathcal{L}_{\text{MF}} = \mathbb{E}_{\epsilon,x,r,t}\left\|u_\theta(x_t, r, t) - \text{sg}(u_{\text{tgt}})\right\|_2^2, \tag{6.10}$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operation, which removes the need for double backpropagation through the Jacobian–vector product.

After training, sampling is performed using the mean velocity:

$$x_r = x_t - (t - r)u(x_t, r, t). \tag{6.11}$$

For one-step sampling, this simplifies to:

$$x = \epsilon - u(\epsilon, 0, 1). \tag{6.12}$$

MeanFlow supports classifier-free guidance while maintaining a single function evaluation (1-NFE) during inference, offering an efficient alternative to standard diffusion-based sampling.

6.2.3 EchoLVFM

`EchoLVFM`, shown in Fig. 6.1, takes in a partial echo video with as little as one frame, a desired $LVEF$, and outputs a generated video of temporal length $f \leq F$ where F is the maximum temporal length. Let $s \in \mathbb{R}^{f \times C \times H \times W}$ denote an echocardiogram sequence with f frames, where C is the number of channels and $H \times W$ is the spatial resolution. Each video is mapped to a latent representation $z \in \mathbb{R}^{f \times C' \times H' \times W'}$ via a learned encoder, where C', H', and W' denote the latent channel dimension and spatial resolution, respectively. The encoder is a 2D image VAE applied frame-by-frame,

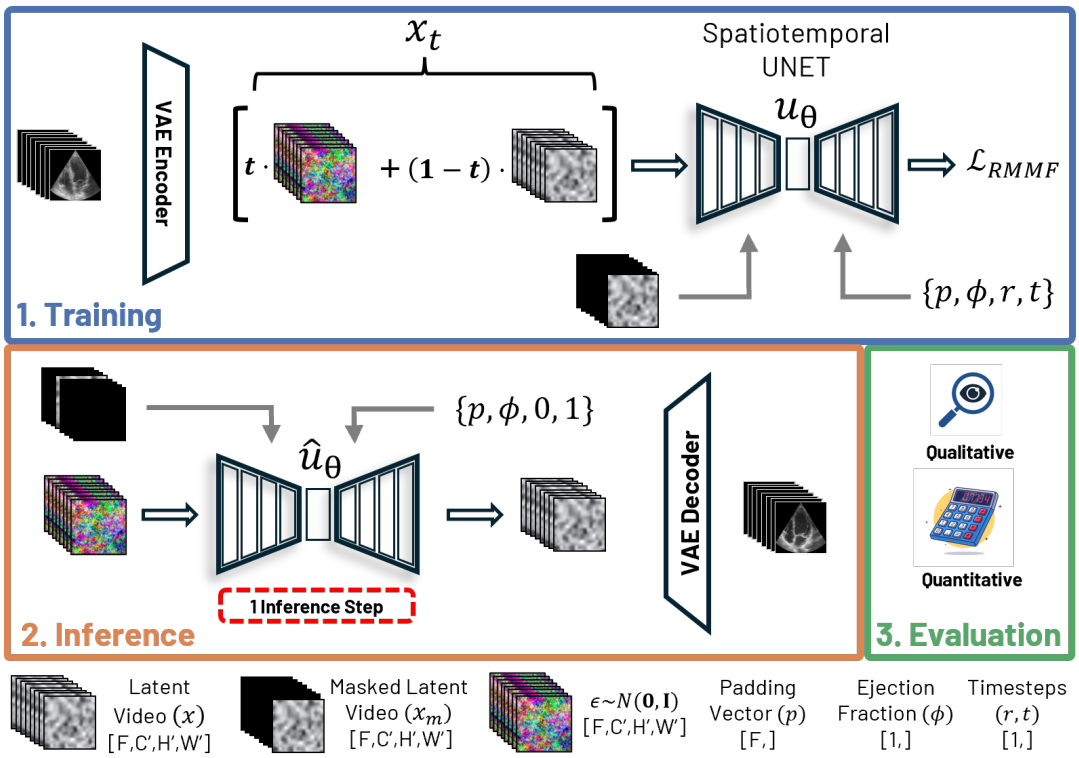


Figure 6.1: EchoLVFM Framework. *Training:* Each echocardiogram is encoded frame-by-frame into a latent representation, temporally augmented to F frames, and noised. A masked latent counterpart x_m is formed by zeroing all but a small subset of observed frames; x_m is supplied as conditioning. A binary padding vector $p \in \{0,1\}^F$ marks valid frames from padded frames in the augmented input. A global conditioning scalar ϕ (e.g. ejection fraction) is also supplied. With timesteps $r < t$, the spatiotemporal U-Net $\hat{u}_\theta$ predicts the conditional average velocity over $[r, t]$ from the noised input together with the conditioning bundle $\mathrm{c} = \{x_m, \phi, p\}$. *Inference:* a partial video containing as little as a single observed frame, together with a target ϕ and random noise, is passed to the trained model. One-step integration with `EchoLVFM-Mean` produces the generated video. The padding vector p additionally controls the output video length up to F.

so each latent frame is computed independently and z carries no learned temporal mixing; temporal structure is modelled exclusively by the downstream spatiotemporal U-Net. The masked latent video $z_m \in \mathbb{R}^{f \times C' \times H' \times W'}$ is created by zeroing out frames in z and represents a partial video.

To enable batched processing and support variable-length sequences, z is temporally augmented to a fixed length F. Notably, this is achieved in a way that preserves videos with $f < F$. If z contains more than F frames ($f > F$), it is uniformly downsampled to length F. For the CAMUS dataset used in this work, each video contains a single ED $\rightarrow$ ES portion of a cardiac cycle, so uniform temporal downsampling preserves a fixed

phase range and the resulting F-frame representation is well-defined. If z contains fewer than F frames ($f < F$), it is padded in the temporal dimension up to length F using zero frames $0 \in \mathbb{R}^{1\times H'\times W'}$. z_m is resampled identically.

Let $x \in \mathbb{R}^{F\times C'\times H'\times W'}$ denote the temporally augmented latent video tensor, and $x_m \in \mathbb{R}^{F\times C'\times H'\times W'}$ denotes the temporally augmented masked latent video tensor. Note that x here denotes the clean data sample used as the flow-matching target (consistent with 6.2.1), and that x_t in subsequent equations is the noised version of x at timestep t. A corresponding padding indicator vector $p \in \{0,1\}^F$ is defined, where $p_i = 1$ indicates that the i-th frame is a padding frame and $p_i = 0$ indicates a valid frame.

For brevity, the conditioning bundle $\{x_m, \phi, p\}$ is written c in the equations that follow.

Loss Masking

Because short input sequences may contain a large proportion of padded frames after temporal augmentation, a frame masking mechanism is introduced. The mask $LM \in \{0,1\}^{1\times F}$, defined as $LM = 1 - p$, allows the choice of which frames are considered by the training objective. The purpose of this mechanism is to ensure that the training objective can adapt to inputs x containing varying proportions of valid and padding frames, and that padded regions do not unduly influence optimisation. When LM appears inside a squared norm, it is broadcast across the latent channel and spatial dimensions to match the shape of the error tensor; the resulting broadcast mask is denoted M and applied element-wise ($M \odot \cdot$).

EchoLVFM-Linear

To develop `EchoLVFM-Linear`, the standard linear flow matching loss is updated to allow the processing and conditioning of partial videos. `EchoLVFM-Linear`'s loss function can be written as

$$\mathcal{L}_{\text{MLF}} = \mathbb{E}_{\epsilon,x,t}\left[\alpha \left\| M \odot (\hat{v}_\theta(x_t, t; \mathrm{c}) - (\epsilon - x))\right\|_2^2\right], \tag{6.13}$$

where $\phi \in [0,1]$ is a global conditioning parameter (e.g., LVEF), the conditioning bundle $\mathrm{c} = \{x_m, \phi, p\}$ groups all auxiliary inputs to the network, and the normalisation factor

$\alpha = [(\sum_{f=1}^{F} LM_f)\, C'\, H'\, W']^{-1}$ ensures that all videos contribute equally to the loss regardless of the number of unmasked frames. Compared with $\mathcal{L}_{FM}$ in Eq. 6.3, the only structural changes are (i) the velocity target $(\epsilon - x)$ inherits the linear-flow form from 6.2.1, (ii) the conditioning bundle c enters the network so it can attend to the observed frames and the desired ϕ, and (iii) the broadcast mask M zeroes the contribution of padded frames before summation.

EchoLVFM-Mean

For `EchoLVFM-Mean`, the model is trained to predict the average velocity field using a masked Mean Flow objective. Given a noised video sample x_t and times $r < t$, the Mean Flow target (see 6.2.2) is defined as

$$u_{\text{tgt}}(x_t, r, t) = v(x_t, t) - \mathcal{J}(x_t, r, t), \tag{6.14}$$

where $v(x_t, t) = \epsilon - x$ denotes the instantaneous velocity field and $\mathcal{J}(x_t, r, t)$ is the Mean Flow correction term arising from the MeanFlow identity (6.6) (see 6.2.2 for the explicit form). The model prediction is denoted $\hat{u}_\theta(x_t, r, t; \mathrm{c})$, and the error to be regressed is

$$e \;=\; \hat{u}_\theta(x_t, r, t; \mathrm{c}) - \mathrm{sg}(u_{\text{tgt}}(x_t, r, t)), \tag{6.15}$$

To account for variable-length videos, the broadcast mask M is applied so that padded frames do not contribute to the loss. The resulting masked Mean Flow loss is

$$\mathcal{L}_{\text{MMF}} = \mathbb{E}_{\epsilon,x,t,r}\left[\alpha \, \|M \odot e\|_2^2\right], \tag{6.16}$$

where α scales the contribution of each video in a similar fashion to `EchoLVFM-Linear`.

To improve robustness to large residuals, an adaptive per-sample loss weighting [143] may be employed. Specifically, the weight is defined as

$$w = (\|M \odot e\|_2^2 + \varepsilon)^{-h}, \tag{6.17}$$

where h controls the strength of the reweighting and ε is a small constant for numerical stability. Gradients are stopped through w, yielding the adaptively weighted masked Mean Flow loss

$$\mathcal{L}_{\text{MMF}}^{\text{adapt}} = \mathbb{E}\left[\text{sg}(w)\,\alpha\,\|M \odot e\|_2^2\right]. \tag{6.18}$$

This weighting upscales gradient contributions when the residual is small, avoiding vanishing gradients as the model approaches the target and stabilising optimisation across noise levels. Analyses of the MeanFlow framework [144, 145] have also highlighted training instability and optimisation challenges; to mitigate these, a masked reconstruction regulariser is included. Using the predicted Mean Flow, a reconstruction of the clean sample is formed as

$$\hat{x} \;=\; x_t \;-\; t\,(\hat{u}_\theta(x_t, r, t; \text{c}) + \mathcal{J}(x_t, r, t)), \tag{6.19}$$

where $\hat{u}_\theta + \mathcal{J}$ estimates the instantaneous velocity $v(x_t, t)$, so the factor t integrates the linear-flow path from x_t back to the clean sample at $t = 0$. The corresponding masked reconstruction loss is

$$\mathcal{L}_{\text{rec}} = \mathbb{E}\left[\alpha\,\|M \odot (\hat{x} - x)\|_2^2\right]. \tag{6.20}$$

The full `EchoLVFM-Mean` training objective is finally given by

$$\mathcal{L}_{\text{RMMF}} = \mathcal{L}_{\text{MMF}}^{\text{adapt}} + \lambda_{\text{rec}}\,\mathcal{L}_{\text{rec}}, \tag{6.21}$$

where λ_{rec} controls the contribution of the reconstruction term and the subscript RMMF stands for *Regularised Masked Mean Flow*. This formulation builds upon the Mean Flow regression objective by ensuring control over the frames in the loss function and that training remains stable for variable-length video sequences.

Geng et al. [142] reported that the original MeanFlow attains its best performance when alternating objectives during training, using the linear flow matching loss 75% of the time and the Mean Flow loss 25% of the time. The same schedule is adopted

here: at each training step $\mathcal{L}_{\text{MLF}}$ (6.13) is selected with probability $P(\mathcal{L}_{\text{MLF}})$ and $\mathcal{L}_{\text{RMMF}}$ with probability $P(\mathcal{L}_{\text{RMMF}}) = 1 - P(\mathcal{L}_{\text{MLF}})$.

In summary, the linear-flow and Mean Flow formulations, the adaptive per-sample loss weighting, and the alternating loss schedule are adopted from prior work; the loss-masking mechanism, the partial-video conditioning bundle c, and the masked reconstruction regulariser are the components introduced here to support conditional, variable-length video generation.

6.3 Experiments

6.3.1 Data

This chapter relies on the CAMUS dataset [9] and uses the same splits as the original authors. This results in a train set containing 800 videos from 400 patients, a validation set containing 100 videos from 50 patients, and a test set containing 100 videos from 50 patients. In order to train a latent flow matching model, all data needed to be converted to their latent representation using a VAE. The next set of experiments explored three strong options released by Reynaud et al. [14]. These VAEs were trained on large-scale TTE datasets totalling more than a million frames drawn from datasets like EchoNet-Dynamic [7], TMED2 [146], and others. Notably, the authors do not mention using the CAMUS dataset to train these VAEs, further necessitating experimentation.

Choice of VAE

Reynaud et al. [14] released three VAEs of varying latent dimensions, namely, 4f4 with resolution $[28, 28, 4]$, 4f8 with resolution $[14, 14, 4]$, and 16f8 with resolution $[14, 14, 16]$, where each triple denotes the latent height, width, and channel dimensions ($H' \times W' \times C'$). The ideal VAE would faithfully reconstruct the images, represent the data distribution well in its latent space to facilitate generation, and would also be computationally efficient. For this setting, the main computational bottleneck is GPU space, especially since the use of self-attention and cross-attention blocks in the model is desirable. Therefore, the ideal 'computationally efficient' VAE would be an xf8 variant due to its lower spatial resolution.

To determine the best VAE, each VAE was first tested on a reconstruction task. The whole dataset was resized to 112 x 112 which is the input resolution originally used to train all three VAEs, then passed through the encoder (E) and decoder (D) i.e., $(\hat{x} = D(E(x)))$. The reconstructed images were then compared with the original. Furthermore, preliminary video generation experiments were performed to evaluate to ensure the data was well-represented within each VAE's latent space. These experiments involved training `EchoLVFM-Linear` and `EchoLVFM-Mean` in a broad hyperparameter sweep.

6.3.2 Training

Due to computational constraints, all models were trained on a single NVIDIA L40S GPU with 48 GB of memory. Table 6.1 summarises the model configurations, parameters used for each flow-matching variant, and the corresponding training schedules. Classifier-free guidance was used, with conditioning dropped during training with probability p_{uncond}; however the null condition was included as a learnable parameter. The guidance scale G controls the strength of this correction at sampling time (see Section 2.4.2); $G = 1$ recovers the conditional model with no guidance applied, while $G > 1$ amplifies the conditional signal at the cost of sample diversity.

Training `EchoLVFM-Mean` requires computation of the Jacobian vector product (JVP). At present, flash attention does not support JVP computation, and computing attention without flash attention is prohibitively expensive in both memory and runtime. Consequently, a custom attention processor was developed using the `jvp-flash-attention` Python package [147].

6.3.3 Run Configurations

Loss Masking

Different loss masking schemes are considered to see in which setting the model learns best and how that relates to downstream performance.

No Mask (LM_{nm}) calculates the loss over the whole video tensor. For padded frames, where the original video was shorter than F, this means the padded frames (filled

Table 6.1: Video Generation Training Configuration. *Model Configuration*: Feature dimensions and attention layers are applied symmetrically across the downsampling and upsampling blocks unless otherwise stated. *Training*: Learning Rate (LR) . *Flow Parameters*: $P(\mathcal{L}_{\text{MLF}})$ and $P(\mathcal{L}_{\text{RMMF}})$ represent the probability of selecting the masked linear flow, and the masked mean flow loss function at training time. p_{uncond} represents the probability of dropping conditioning during training. λ_{rec} is the weighting applied to the reconstruction loss.

Training Configuration				
	Model Configuration		Training	Flow Parameters
Feature Dimensions	{128, 128, 256, 256}	LR	5e-5	`EchoLVFM-Linear`
Attention Heads	{4, 4, 8, 8 }	Scheduler	Cosine Annealing	$p_{uncond} = 0.15$ Guidance Scale = 1.0
Transformer Layers Per Block	{1, 1, 2, 2 }	Epochs	1000	`EchoLVFM-Mean`
Self- and Cross-Attention	Down: $\{\checkmark, \checkmark, \checkmark, \checkmark\}$ Up: $\{\times, \times, \times, \checkmark\}$	Optimiser	Adam	$P(\mathcal{L}_{\text{MLF}}) = 0.75$
Residual Layers Per Block	2	Batch Size	2	$P(\mathcal{L}_{\text{RMMF}}) = 0.25$
Trainable Parameters	76.8 M	Warmup	10	$\lambda_{\text{rec}} = 1$

with zeros) are included in loss calculations. Consequently, the model may be encouraged to generate pad frames.

Pad (LM_p) masks the padded frames when calculating the loss function. Therefore the model is evaluated on its ability to generate unseen frames *and* reconstruct seen frames.

Pad and Observed (LM_{po}) masks the padded frames and the valid frames visible in x_m. Therefore the model is evaluated only on its ability to generate frames not present in the conditioning.

Frame Sampling

The masked latent video $z_m \in \mathbb{R}^{f \times C' \times H' \times W'}$ is constructed by masking a subset of frames in z. To control how much temporal information is retained in x_m, three frame sampling strategies are considered: *1-frame*, *Geometric*, and *Uniform*. These strategies were selected to represent different levels of temporal conditioning, ranging from minimal

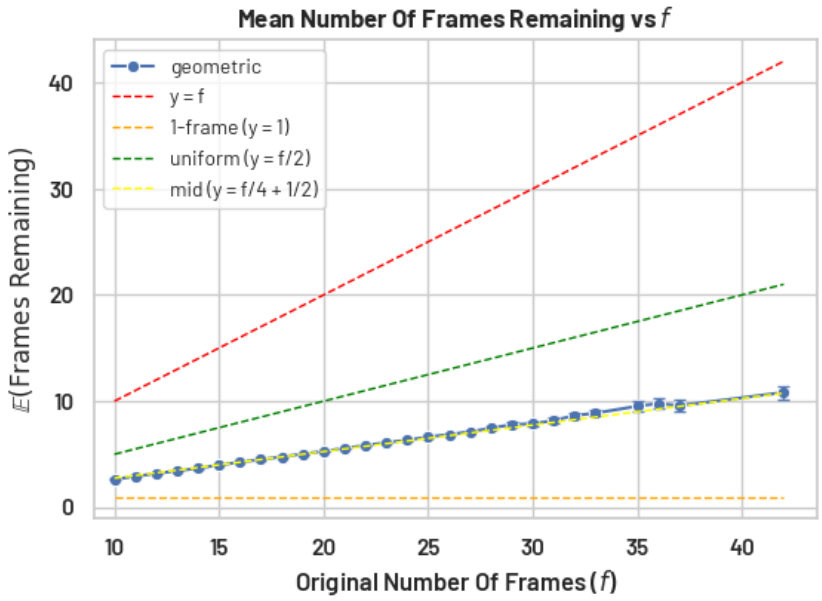


Figure 6.2: Expectations of Sampling Schemes. Frame sampling methods for generating x_m from x.

conditioning using a single observed frame (*1-frame*) to more generous conditioning where a larger subset of frames is retained (*Uniform*), with *Geometric* sampling providing an intermediate with a bias towards removing more frames. Each strategy defines the number of masked frames (NMF), denoted nmf. The expectation of the different strategies can be seen in Fig 6.2

1-frame $nmf = f - 1$, such that all but a random single frame are masked when constructing x_m.

Geometric $nmf = f - K$, where $K \sim \text{Geom}\left(p = \frac{3.7}{f}\right)$.

Uniform $nmf \sim \mathcal{U}(1, f-1)$.

6.3.4 Evaluation

Tasks

The models were evaluated on two tasks: *Reconstruction* and *Generation*. For the reconstruction task, each model was conditioned on the video's ground-truth LVEF, whereas for the generation task a deliberately different LVEF value was used. Algorithm 1

Table 6.2: Run Configurations. The configurations shown were explored to identify effective training settings for `EchoLVFM-Linear` and `EchoLVFM-Mean`. This range of configurations enables identification of the key factors influencing model performance.

Run ID	Loss Mask	Frame Sampling	p Conditioning
R01	No Mask	1-frame	False
R02	No Mask	1-frame	True
R03	Pad	1-frame	True
R04	Pad	geometric	True
R05	Pad	uniform	True
R06	Pad	1-frame	False
R07	Pad and Observed	geometric	True
R08	No Mask	geometric	False

Algorithm 1 Sample LVEF in Generation Task. Algorithm for sampling a suitably distinct LVEF.

Require: Dataset splits $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}$; ground-truth LVEF $LVEF_{\text{gt}}$; minimum difference δ
$LVEF_{\text{min}} \leftarrow \min(\mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{val}})$
$LVEF_{\text{max}} \leftarrow \max(\mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{val}})$
$LVEF_{\text{new}} \leftarrow LVEF_{\text{gt}}$
while $|LVEF_{\text{gt}} - LVEF_{\text{new}}| < \delta$ **do** ▷ default δ = 5
    Sample $LVEF_{\text{new}} \sim \mathcal{U}(LVEF_{\text{min}}, LVEF_{\text{max}})$
end while
return $LVEF_{\text{new}}$

details the LVEF sampling procedure. To ensure a rigorous assessment of generative performance, any sampled LVEF values falling within 5% of the ground-truth LVEF were resampled. Furthermore, the models were tested with masked inputs containing a varying amount of information at inference time. These settings were:

nmfmax corresponded to the case where x_m only contained one valid frame whilst the rest were padding frames

nmf{75p, 50p, 25p} corresponded to the case where x_m contained {75%, 50%, 25%} of the original frames in z.

The resulting videos were evaluated using both image- and video-quality metrics, specifically SSIM, PSNR, LPIPS, MAE, FID, and FVD. For the generation task, only the distribution-based metrics FID and FVD were reported, as no pairwise ground-truth videos are available.

Table 6.3: VAE CAMUS Image Reconstruction Metrics. The performance of each VAE on a reconstruction task using the CAMUS dataset. MAE denotes the mean absolute error in pixel space.

VAE CAMUS Image Reconstruction Metrics							
VAE	FID ↓	FVD_{10} ↓	SSIM ↑	LPIPS ↓	PSNR ↑	MAE ↓	MSE ↓
$4f4_{[28,28,4]}$	5.16	36.1	$0.958 \pm 1.14\times10^{-4}$	$0.0272 \pm 4.84\times10^{-5}$	35.3 ± 0.0166	$0.0100 \pm 2.10\times10^{-5}$	$0.00030 \pm 1.24\times10^{-6}$
$4f8_{[14,14,4]}$	26.2	74.9	$0.847 \pm 3.07\times10^{-4}$	$0.0693 \pm 7.10\times10^{-5}$	28.8 ± 0.0170	$0.0209 \pm 4.46\times10^{-5}$	$0.00136 \pm 5.58\times10^{-6}$
$16f8_{[14,14,16]}$	15.4	34.3	$0.930 \pm 2.16\times10^{-4}$	$0.0334 \pm 7.16\times10^{-5}$	34.6 ± 0.0193	$0.0117 \pm 3.66\times10^{-5}$	$0.00037 \pm 3.39\times10^{-6}$

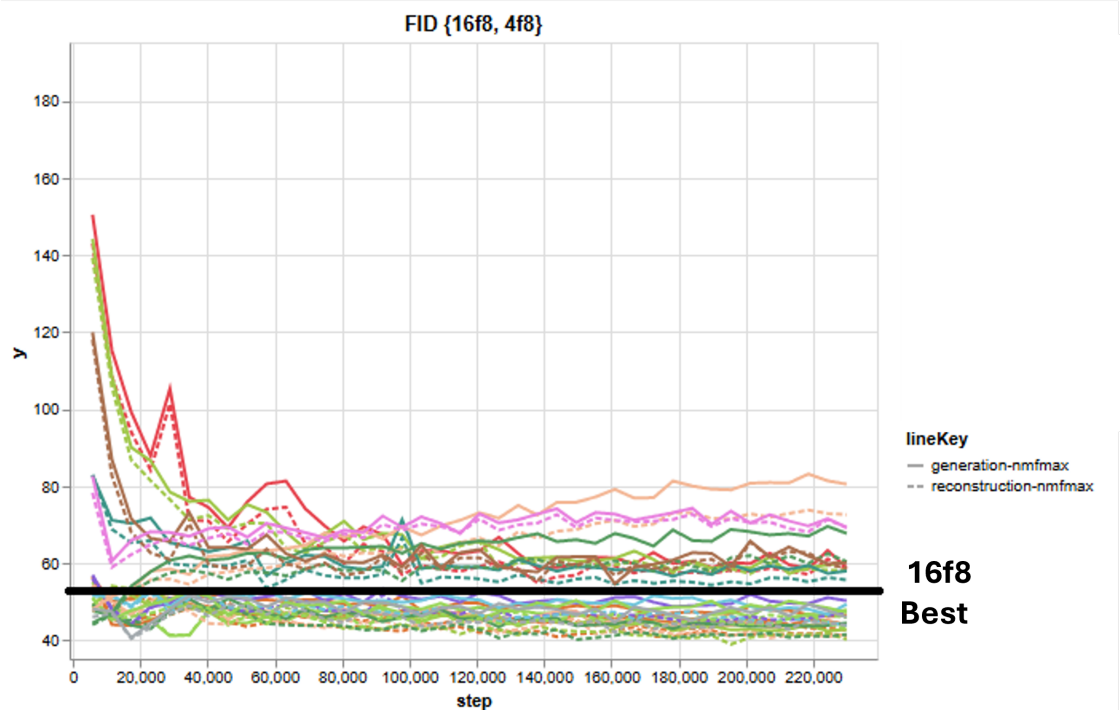


Figure 6.3: VAE Performance in Preliminary Experiments. The models are evaluated on both reconstruction and generation tasks using the FID metric at intervals of 50 epochs on the validation dataset. The black line indicates the lowest FID score achieved by a model trained in the 16f8 latent space. All lines above the black line correspond to 16f8, whilst all below correspond to 4f8 latent space results.

6.4 Results

6.4.1 VAE

Table 6.3 reports reconstruction metrics on the CAMUS dataset for each VAE configuration. For SSIM, LPIPS, PSNR, MAE, and MSE, the 4f4 configuration shows the strongest performance, with 16f8 performing next best and 4f8 performing worst. For FVD, however, the ordering differs, with 16f8 showing the strongest performance and both 4f4 and 4f8 performing less well.

Figure 6.3 shows the results of preliminary generation experiments for models trained in the 16f8 and 4f8 latent spaces. Across all runs, models trained in the 16f8 latent space exhibit poorer performance and require more training epochs to reach their minimum FID values. In contrast, models trained in the 4f8 latent space perform consistently better and approach their minimum FID by the first evaluation point at epoch 50. Notably, this ordering differs from that observed for reconstruction performance in Table 6.3. Experiments with 4f4 yielded more promising results, as seen in Table 6.3, and so were used for all generation.

6.4.2 Video Generation

Video Quality

This section presents the results of `EchoLVFM-Linear` and `EchoLVFM-Mean` when trained using the eight configurations shown in Table 6.2. In Fig. 6.5 and Fig. 6.6, the results for configuration R07 differ substantially in scale and therefore skew the plots, making trends difficult to visualise. For this reason, the values for R07 are reported separately in Table 6.7.

First, `EchoLVFM-Linear` is evaluated under varying guidance scales and numbers of inference steps. The results for the best- and worst-performing configurations in the linear setting are shown in Fig. 6.4. Performance across most metrics plateaus by 10 inference steps; however, FVD continues to improve up to 25 inference steps. In addition, the best-performing guidance scale is 1, corresponding to no classifier-free guidance. As a result, all subsequent evaluations of `EchoLVFM-Linear` are conducted using 25 inference steps and a guidance scale of 1.

The model trained with configuration R05 and evaluated without classifier-free guidance performs best across both reconstruction and generation tasks. This model approaches its optimal performance within as few as two inference steps. In contrast, the model trained using configuration R07 performs substantially worse and shows limited improvement even when evaluated with up to 250 inference steps.

Figures 6.5 and 6.6 show the performances of `EchoLVFM-Linear` and `EchoLVFM-Mean` on Reconstruction and Generation tasks with varying nmf levels.

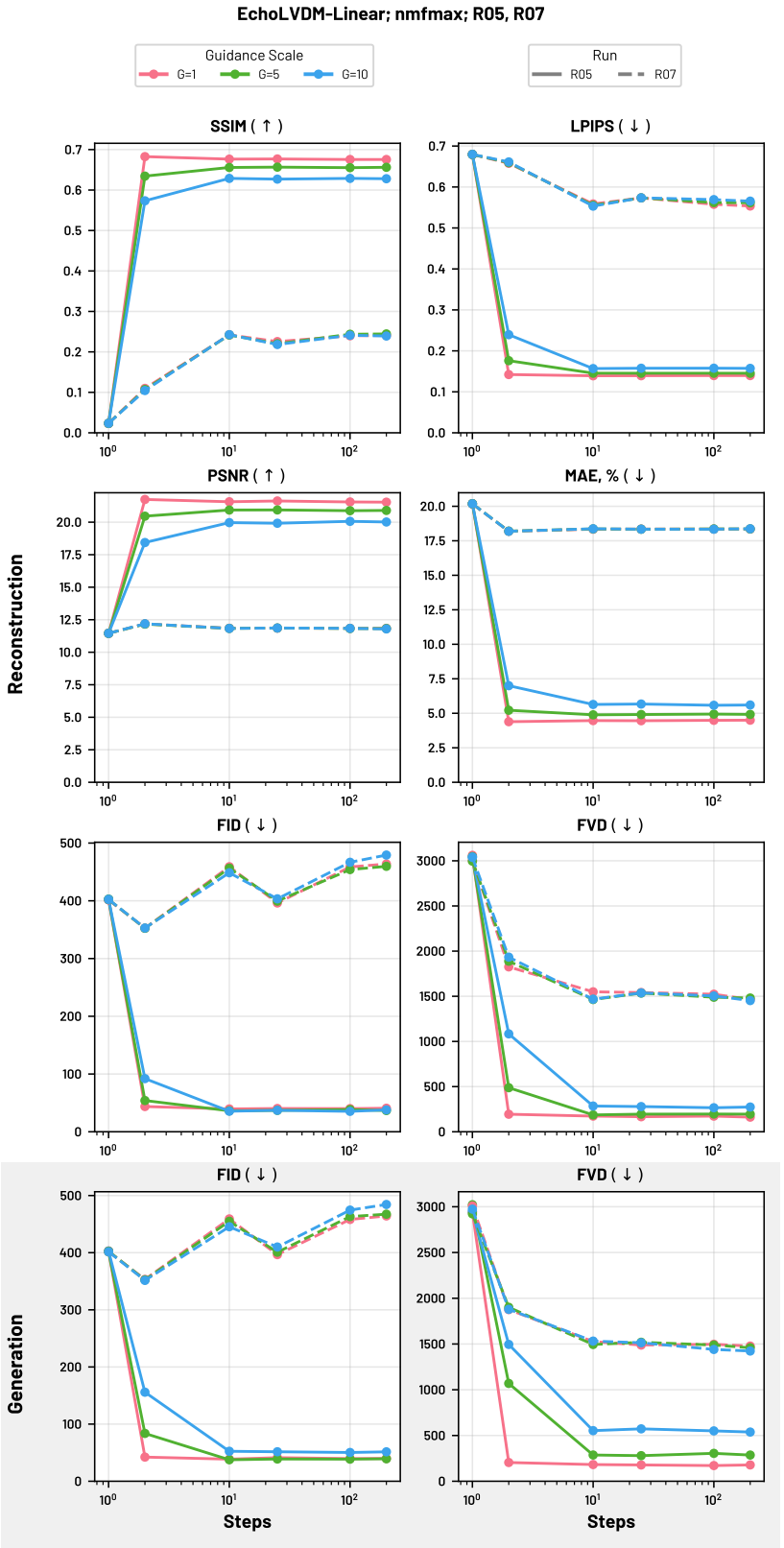


Figure 6.4: Optimal Inference Steps Linear. This figure shows how the best and worst performing run configurations progressed on Reconstruction and Generation tasks when only one frame is provided in x_m. Evaluations with differing guidance scales are also shown. The maximum number of inference steps is 250.

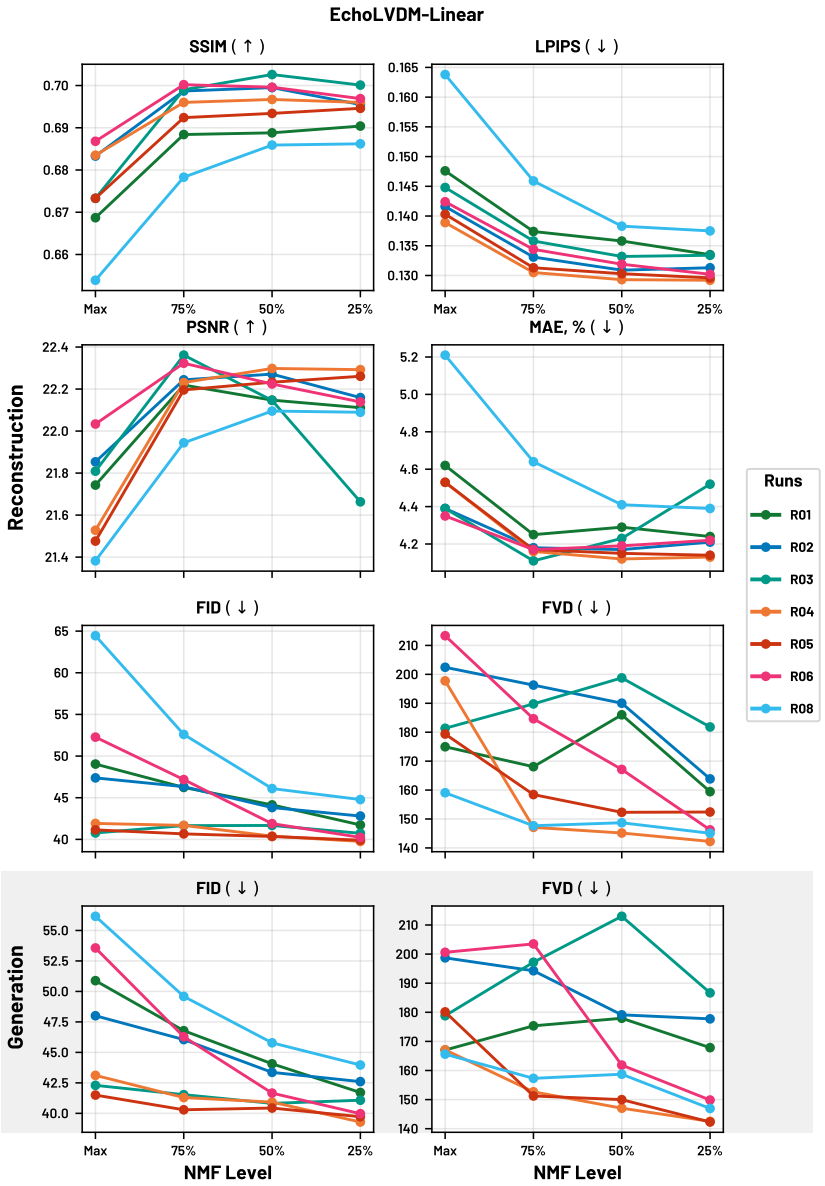


Figure 6.5: `EchoLVFM-Linear` Runs. Performance of `EchoLVFM-Linear` across all tasks and at different *nmf* levels. Configuration R07 is omitted from this plot, as its values are of a different order and significantly skew the visualisation.

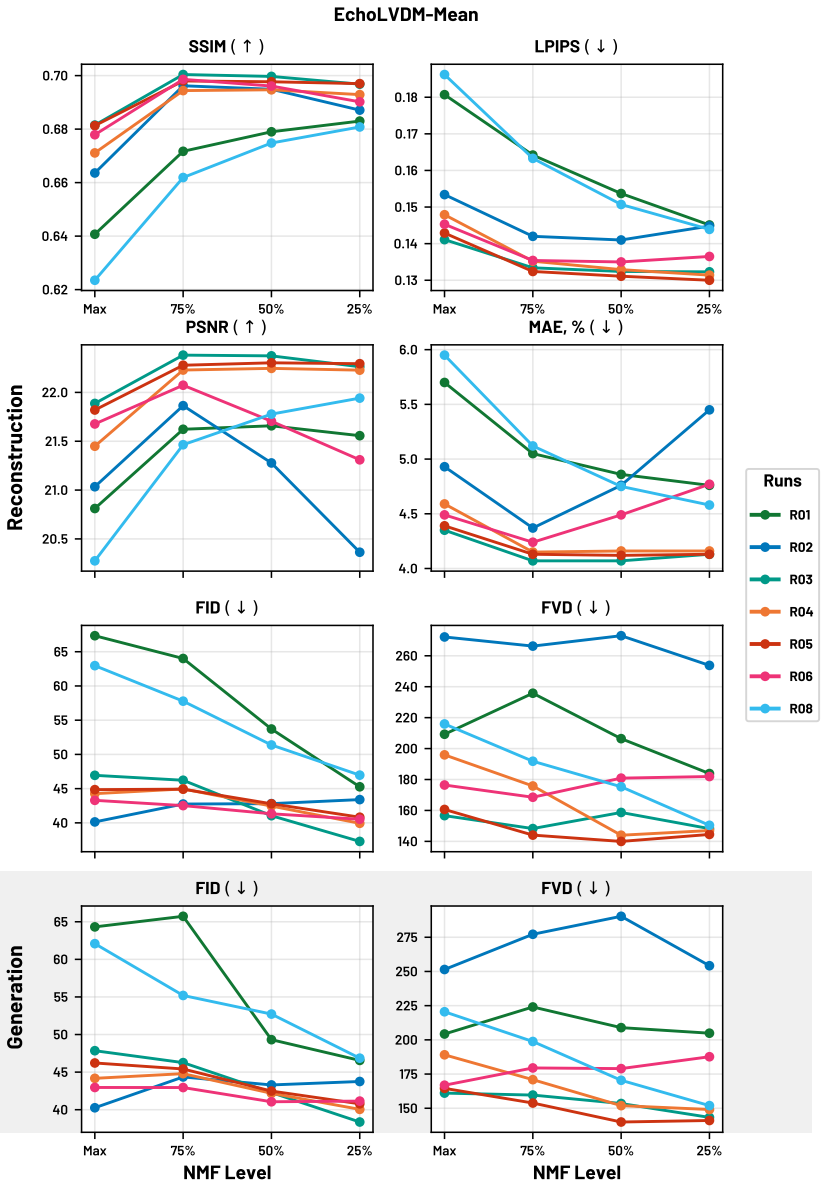


Figure 6.6: `EchoLVFM-Mean` **Runs.** Performance of `EchoLVFM-Mean` across all tasks and at different *nmf* levels. Configuration R07 is omitted from this plot, as its values are of a different order and significantly skew the visualisation.

Table 6.4: `EchoLVFM-Linear` Run Configuration Rankings. This table shows the Rank-Sum and Min–max performance of each run across all metrics. Rank-Sum ranks the runs for each metric and sums their positions. Min–max rescales each metric by its range, subtracts the minimum, and then normalises the scores. This approach preserves magnitude information, allowing (near-)ties between runs to be distinguished. Gold, Silver, and Bronze shading highlight the top 3 configurations.

`EchoLVFM-Linear` Run Configuration Rankings										
Rank	Max		75%		50%		25%		Overall	
	Rank-sum	Min–max	Rank-sum	Min–max	Rank-sum	Min–max	Rank-sum	Min–max	Rank-sum	Min–max
1	R02 (20)	R03 (6.1)	R04 (15)	R04 (7.1)	R04 (10)	R04 (7.5)	R04 (8)	R04 (7.6)	R04 (6)	R04 (28.1)
2	R03 (20)	R04 (5.9)	R03 (17)	R05 (6.8)	R05 (16)	R05 (6.7)	R05 (17)	R05 (7.1)	R05 (12)	R05 (26.2)
3	R04 (20)	R05 (5.7)	R05 (20)	R03 (5.7)	R02 (22)	R06 (5.9)	R06 (19)	R06 (6.6)	R02 (13)	R06 (22.4)
4	R06 (20)	R02 (5.3)	R06 (22)	R06 (4.8)	R06 (22)	R02 (5.0)	R02 (26)	R02 (4.5)	R03 (14)	R02 (19.5)
5	R05 (23)	R01 (5.1)	R02 (26)	R02 (4.7)	R03 (27)	R03 (4.3)	R03 (31)	R01 (4.4)	R06 (15)	R03 (19.2)
6	R01 (29)	R06 (5.1)	R01 (31)	R01 (4.5)	R01 (34)	R01 (2.7)	R01 (32)	R03 (3.0)	R01 (24)	R01 (16.7)
7	R08 (36)	R08 (2.2)	R08 (37)	R08 (2.1)	R08 (37)	R08 (1.8)	R08 (35)	R08 (3.0)	R08 (28)	R08 (9.1)
8	R07 (48)	–	R07 (48)	–	R07 (48)	–	R07 (48)	–	R07 (32)	–

Table 6.5: `EchoLVFM-Mean` Run Configuration Rankings. This table shows the Rank-Sum and Min–max performance of each run across all metrics. Rank-Sum ranks the runs for each metric and sums their positions. Min–max rescales each metric by its range, subtracts the minimum, and then normalises the scores. This approach preserves magnitude information, allowing (near-)ties between runs to be distinguished. R07 was not included in min-max normalisation as it strongly skews the values. Gold, Silver, and Bronze shading highlight the top 3 configurations.

`EchoLVFM-Mean` Run Configuration Rankings										
Rank	Max		75%		50%		25%		Overall	
	Rank-sum	Min-max	Rank-sum	Min-max	Rank-sum	Min-max	Rank-sum	Min-max	Rank-sum	Min-max
1	R03 (10)	R03 (7.4)	R03 (12)	R03 (7.5)	R03 (9)	R03 (7.6)	R05 (10)	R03 (7.6)	R03 (5)	R03 (30.2)
2	R05 (14)	R05 (7.4)	R05 (12)	R05 (7.5)	R05 (12)	R05 (7.5)	R03 (12)	R05 (7.2)	R05 (7)	R05 (29.6)
3	R06 (17)	R06 (7.1)	R06 (18)	R06 (6.9)	R04 (19)	R04 (7.2)	R04 (15)	R04 (6.9)	R04 (14)	R04 (27.3)
4	R04 (23)	R04 (6.3)	R04 (22)	R04 (6.9)	R06 (23)	R06 (6.0)	R06 (28)	R06 (4.6)	R06 (14)	R06 (24.7)
5	R02 (28)	R02 (4.7)	R02 (28)	R02 (4.7)	R08 (32)	R02 (3.3)	R08 (31)	R08 (3.4)	R02 (23)	R02 (13.8)
6	R01 (36)	R01 (2.2)	R01 (38)	R08 (2.1)	R02 (34)	R08 (2.6)	R01 (35)	R01 (2.6)	R08 (24)	R08 (9.3)
7	R08 (40)	R08 (1.3)	R08 (38)	R01 (1.3)	R01 (39)	R01 (2.0)	R02 (37)	R02 (1.1)	R01 (25)	R01 (7.9)
8	R07 (48)	-	R07 (48)	-	R07 (48)	-	R07 (48)	-	R07 (32)	-

6.5 Discussion

6.5.1 VAEs

The observed reconstruction performance differences between VAE configurations appear to be related to the spatial resolution of their latent representations. Although 4f4 and 16f8 have the same latent dimensionality, the higher spatial resolution of 4f4 enables more accurate reconstruction of fine image details, which is reflected in improved SSIM, LPIPS, and error-based metrics. In contrast, the lower FVD achieved by 16f8 suggests that its latent structure may better preserve short-term temporal consistency

Table 6.6: EchoLVFM Video Generation Performance. Image (I), Text (T), Semantic Mask (M), Motion Mask (MM), Video (V), Partial Video (pV), and Scalar (S). †Although the authors do not share the number of inference steps used, the authors use DDPM [84] which is known to typically require several hundreds of steps. The authors of the original DDPM paper use 1000. ‡ The original CAMUS split contains videos with as little as 10 frames in its test set. However, the default FVD processes 16 frame batches. **Bold** shows the best overall score per metric. **Colour** shows the best generation scores. Baseline methods are reported for the reconstruction task only: they condition on spatial inputs rather than a target ejection fraction, and no prior work evaluates ejection-fraction-conditioned video generation on the CAMUS dataset.

CAMUS Video Reconstruction Metrics								
Method	Condition	Inference Steps	FID ↓	FVD_{10}‡ / FVD_{16} ↓	SSIM ↑	LPIPS ↓	PSNR ↑	MAE, % ↓
HeartBeat - A2CH [141]	I, T	-†	36.34	-/**9.60**	0.61	-	-	-
HeartBeat - A4CH [141]	I, T	-†	36.00	-/12.58	0.61	-	-	-
Free-Echo [148]	I, M	15	**26.31**	-/195.41	0.48	-	20.43	-
Free-Echo [148]	I, M	35	53.71	-/312.12	0.27	-	16.54	-
Free-Echo [148]	I, M	55	67.15	-/412.55	0.12	-	13.45	-
ControlEchoSynth [149]	MM, V	1000	26.64	-/69.58	0.57	0.16	-	-
EchoLVFM-Linear$_{nmfmax}$	I, S	25	40.77	181.35/-	0.67	0.14	21.80	4.39
EchoLVFM-Linear$_{nmf25p}$	pV, S	25	40.73	181.81/-	**0.70**	**0.13**	21.66	4.52
EchoLVFM-Mean$_{nmfmax}$	I, S	**1**	46.94	156.65/-	0.68	0.14	21.89	4.35
EchoLVFM-Mean$_{nmf25p}$	pV, S	**1**	37.29	**148.23**/-	**0.70**	**0.13**	**22.26**	**4.13**
CAMUS Video Generation Metrics								
EchoLVFM-Linear$_{nmfmax}$	I, S	25	42.3	178.82/-				
EchoLVFM-Linear$_{nmf25p}$	pV, S	25	41.1	186.68/-				
EchoLVFM-Mean$_{nmfmax}$	I, S	**1**	47.85	161.08/-				
EchoLVFM-Mean$_{nmf25p}$	pV, S	**1**	**38.36**	**143.31**/-				

despite reduced spatial fidelity.

Despite 16f8 achieving better reconstruction metrics, preliminary experiments showed that the lower resolution 4f8 was better for training. This trend is counter to what was observed in the reconstruction results in Table 6.3, where models trained in the 16f8 latent space performed better. One possible explanation is that, while the 16f8 latent space is sufficient for reconstruction, it does not provide a representation that is as well suited for generation on the CAMUS dataset. In contrast, the 4f8 latent space appears to enable more stable training and faster convergence for generation tasks, despite its lower spatial resolution. Experiments with 4f4 yielded more promising results and so were used for all generation.

Table 6.7: R07 Results. *R07* run performed the worst out of all datasets. This is thought to be because of the Pad And Observed mask scheme used.

	`EchoLVFM-Mean` **Reconstruction Metrics**					
NMF	SSIM	LPIPS	PSNR	MAE (%)	FID	FVD
Mean						
Max	0.132	0.636	12.01	18.28	334.1	1787.8
75%	0.261	0.519	14.35	15.05	245.1	1535.3
50%	0.406	0.389	16.97	11.41	165.5	1197.4
25%	0.557	0.252	19.72	7.61	92.2	693.1
	`EchoLVFM-Mean` **Generation Metrics**					
NMF	SSIM	LPIPS	PSNR	MAE (%)	FID	FVD
Max	–	–	–	–	333.3	1762.7
75%	–	–	–	–	245.0	1498.3
50%	–	–	–	–	164.9	1187.8
25%	–	–	–	–	92.3	771.6
	`EchoLVFM-Linear` **Reconstruction Metrics**					
NMF	SSIM	LPIPS	PSNR	MAE (%)	FID	FVD
Linear						
Max	0.225	0.574	11.84	18.35	396.3	1529.8
75%	0.334	0.471	14.26	15.09	293.0	1219.7
50%	0.453	0.357	16.89	11.44	199.1	938.3
25%	0.580	0.237	19.69	7.61	109.5	585.1
	`EchoLVFM-Linear` **Generation Metrics**					
NMF	SSIM	LPIPS	PSNR	MAE (%)	FID	FVD
Max	–	–	–	–	397.0	1496.0
75%	–	–	–	–	294.0	1190.7
50%	–	–	–	–	199.1	919.5
25%	–	–	–	–	110.7	549.0

6.5.2 Video Quality

The video quality results in Fig. 6.4 indicate that `EchoLVFM-Linear` benefits from a relatively small number of inference steps, with most metrics reaching a plateau by 10 steps and only modest improvements in FVD observed beyond this point. This suggests

that the linear flow formulation captures much of the required temporal structure early in the inference process, while additional steps primarily refine temporal consistency rather than overall appearance. The continued improvement in FVD at higher step counts indicates that temporal coherence may require slightly longer integration even when spatial fidelity has stabilised.

The observation that a guidance scale of 1 performs best indicates that classifier-free guidance on the global parameter does not provide additional benefit in this setting. Since the masked video input and padding information are always provided to the model, the temporal and structural conditioning remains fixed across guidance scales. One possible explanation is that the EF conditioning signal alone is already sufficiently integrated during training, such that increasing its guidance strength does not further improve generation quality and may instead constrain the model unnecessarily. Alternatively, perhaps the null ϕ value learned during training is not an effective null and actually interferes with sampling.

The strong performance of configuration R05, which reaches near-optimal performance within only a few inference steps, suggests that this configuration provides a favourable balance between conditioning strength and model flexibility. In contrast, the consistently poor performance of R07, even when evaluated with a large number of inference steps, indicates that this configuration limits the model's ability to recover meaningful temporal dynamics once generation begins. Together, these observations motivate the selection of R05 and support the use of a small number of inference steps for subsequent experiments with `EchoLVFM-Linear`.

Results in Table 6.6 show that echocardiograms competitive with state-of-the-art generation methods can be produced via one-step flow matching.

6.6 Conclusion

A consistent finding across the VAE experiments was that the 4f4 latent configuration was better for flow matching, showing that spatial resolution was more important for the downstream generation task.

The one-step sampling framework further demonstrated that high-quality image and video generation can be achieved without iterative refinement, supporting the use of flow matching as an efficient alternative to multi-step diffusion in this setting.

Future work includes applying `EchoLVFM` for TEE video generation and also using the synthetic videos to train a model on a downstream task e.g., predicting the global parameter of choice.

For the Lord your God is bringing you into a good land ...

When you have eaten and are satisfied, praise the Lord your God for the good land he has given you.
Be careful that you do not forget the Lord your God, failing to observe his commands,
his laws and his decrees that I am giving you this day.

— Deuteronomy 8:7, 10–11

7 Conclusion

Contents

7.1 Contributions 123
7.1.1 TEE Generation using Anatomical Models - `HeartToEcho` . . 124
7.1.2 From Transthoracic to Transesophageal - `EchoLoRA` 124
7.1.3 One-step Echocardiogram Video Generation - `EchoLVFM` . . . 124
7.2 Limitations and Future Work 125
7.2.1 TEE Generation using Anatomical Models 125
7.2.2 From Transthoracic to Transesophageal 125
7.2.3 One-step Video Generation 126
7.2.4 Practical Considerations and Open Directions 126

7.1 Contributions

In this thesis, the use of deep generative models has been explored as a means of addressing key challenges in echocardiography, particularly those arising from data scarcity, operator dependence, and the difficulty of acquiring large, standardised TEE datasets. The work focuses on generating anatomically and physiologically meaningful synthetic echocardiographic data, with the aim of supporting specialist training, algorithm development, and downstream clinical tasks.

7.1.1 TEE Generation using Anatomical Models - HeartToEcho

The first contribution introduced HeartToEcho, a pipeline developed to extract standard TEE views from three-dimensional anatomical heart models and to generate corresponding pseudo-images. I2I translation methods were then evaluated for transforming these pseudo-images into realistic synthetic echocardiograms. Both qualitative assessment and quantitative evaluation demonstrated that the generated images were visually plausible and could improve performance on a downstream left ventricular segmentation task. In addition, disagreement between human perception and commonly used image quality metrics highlighted limitations in relying solely on automated measures of realism for medical image synthesis.

7.1.2 From Transthoracic to Transesophageal - EchoLoRA

Building on HeartToEcho, the second contribution proposed EchoLoRA, a data-efficient framework designed to address limitations related to ease of use and cross-domain learning in synthetic TEE image generation. By combining LoRA with pretrained diffusion models, EchoLoRA enabled effective transfer of information from TTE to TEE under limited data availability. The resulting synthetic images maintained strong adherence to anatomical conditioning and led to improved multiclass segmentation performance, particularly for underrepresented anatomical structures.

7.1.3 One-step Echocardiogram Video Generation - EchoLVFM

The final contribution extended generative echocardiography from images to videos through the introduction of EchoLVFM, a one-step latent video flow matching model for temporally coherent echocardiogram generation. This approach supports variable-length sequences and masked video conditioning, enabling tasks such as video completion and animation from partial inputs. Quantitative evaluation demonstrated that the generated sequences capture the spatial and temporal features of real echocardiograms.

7.2 Limitations and Future Work

7.2.1 TEE Generation using Anatomical Models

The first contribution demonstrated that anatomically derived pseudo-images can be used to generate realistic synthetic TEE images and to improve downstream segmentation performance when used for data augmentation. However, an observed limitation of this approach is that the resulting synthetic images do not achieve strong scores in image-quality metrics. Despite this, synthetic images with lower metric scores were often still beneficial for downstream tasks, while images with higher metric scores did not necessarily lead to improved segmentation performance. This behaviour was observed in both Chapter 4 and Chapter 5, where image-quality metrics did not reliably predict augmentation benefit.

These findings suggest that current image-quality metrics may not adequately capture properties that are important for downstream clinical tasks, and that high metric performance alone should not be interpreted as an indicator of clinical usefulness. In addition, while the pseudo-image to synthetic-image pipeline enabled the generation of labelled data at scale, the synthetic images produced in this chapter are not intended to replace real clinical data and are unlikely to be sufficient for tasks such as specialist training when used in isolation. Instead, their primary utility lies in augmenting existing datasets and supporting the development and evaluation of data-driven methods.

7.2.2 From Transthoracic to Transesophageal

The second contribution introduced `EchoLoRA`, which addressed several limitations of the `HeartToEcho` pipeline by generating synthetic TEE images directly from semantic masks. This removed the need for manually designed view-specific transformations and enabled more efficient cross-domain learning from TTE to TEE. Nevertheless, a limitation of this approach is that the use of LoRA adapters does not reduce the GPU memory requirements associated with training large models. As a result, the computational cost of training remains a bottleneck for many researchers, particularly those without access to high-memory hardware.

A potential direction for future work is the integration of knowledge distillation or model compression techniques to reduce memory requirements while preserving the benefits of data-efficient adaptation. In addition, the observation that image-quality metrics did not consistently correlate with downstream performance in this chapter further highlights the need for improved evaluation strategies when assessing synthetic medical images, particularly in augmentation settings.

7.2.3 One-step Video Generation

The final contribution proposed `EchoLVFM`, a one-step latent video flow matching model for temporally coherent echocardiogram generation. While this approach supports variable-length inputs and masked video conditioning, it imposes a fixed upper bound of F frames on the length of generated sequences. Consequently, the method is not directly suited to generating long echocardiogram videos that exceed this temporal window.

A natural extension of this work would be the development of a strategy for generating longer sequences in a staged or chunked manner, where multiple segments of length F are generated and stitched together while maintaining temporal coherence. In addition, the reliance on a pretrained latent video autoencoder introduces a dependency on the quality and availability of such models. Although recent public releases of video VAEs enabled the experiments presented in this chapter, further improvements in latent representations may yield additional gains in both quality and efficiency.

7.2.4 Practical Considerations and Open Directions

A practical bottleneck encountered throughout this thesis was the availability of suitable pretrained latent autoencoders. In particular, the work on video generation would not have been feasible without recent public releases of video VAEs, as pixel-space generation of echocardiogram videos is prohibitively expensive for most researchers. Continued progress in this area is therefore likely to have a significant impact on future research.

Finally, the LVEF-conditioning direction explored in this thesis has been developed further in subsequent work [150], which extends `EchoLVFM` to evaluate ejection fraction as a generation conditioning signal in greater depth.

Overall, this thesis demonstrates that anatomically conditioned generative models can be used to generate clinically meaningful echocardiographic data across both image and video settings. By reducing reliance on large annotated datasets, supporting more consistent generation of standard views, and explicitly handling variability in echocardiographic acquisition such as incomplete, short, or heterogeneous sequences, the proposed methods provide a practical foundation for data-driven echocardiography. These contributions highlight the potential of generative modelling as a complementary tool within broader pipelines for training, evaluation, and downstream cardiac analysis.

For lack of guidance a nation falls, but victory is won through many advisers

— Proverbs 11:14

References

[1] Islam Aly et al. “Cardiac ultrasound: An Anatomical and Clinical Review”. In: *Translational Research in Anatomy* 22 (Jan. 2021), p. 100083. URL: https://www.sciencedirect.com/science/article/pii/S2214854X20300224.

[2] Andy Potter, Keith Pearce, and Naguib Hilmy. “The benefits of echocardiography in primary care”. In: *British Journal of General Practice* 69 (684 July 2019), pp. 358–359. URL: https://bjgp.org/content/69/684/358%20https://bjgp.org/content/69/684/358.abstract.

[3] Roberto M Lang et al. “Recommendations for Cardiac Chamber Quantification by Echocardiography in Adults: An Update from the American Society of Echocardiography and the European Association of Cardiovascular Imaging”. In: *Journal of the American Society of Echocardiography* 28 (2015), pp. 1–39. URL: http://dx.doi.org/10.1016/j.echo.2014.10.003.

[4] Rebecca T Hahn et al. “ASE GUIDELINES AND STANDARDS Guidelines for Performing a Comprehensive Transesophageal Echocardiographic Examination: Recommendations from the American Society of Echocardiography and the Society of Cardiovascular Anesthesiologists”. In: (2013).

[5] Mohammad Ali Maraci et al. “Searching for structures of interest in an ultrasound video sequence”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8679 (2014), pp. 133–140.

[6] Jianzhe Lin et al. *Reciprocal Landmark Detection and Tracking With Extremely Few Annotations*. 2021.

[7] David Ouyang et al. “Video-based AI for beat-to-beat assessment of cardiac function”. In: *252 | Nature |* 580 (2020). URL: https://doi.org/10.1038/s41586-020-2145-8.

[8] Ali Madani et al. “Fast and accurate view classification of echocardiograms using deep learning”. In: *npj Digital Medicine 2018 1:1* 1.1 (Mar. 2018), pp. 6–. URL: https://www.nature.com/articles/s41746-017-0013-1.

[9] Sarah Leclerc et al. “Deep Learning for Segmentation Using an Open Large-Scale Dataset in 2D Echocardiography”. In: *IEEE transactions on medical imaging* 38.9 (Sept. 2019), pp. 2198–2210. URL: https://pubmed.ncbi.nlm.nih.gov/30802851/.

[10] Thorstensen A et al. “Reproducibility in echocardiographic assessment of the left ventricular global and regional function, the HUNT study”. In: *European journal of echocardiography : the journal of the Working Group on Echocardiography of the European Society of Cardiology* 11 (2 Mar. 2010), pp. 149–156. URL: https://pubmed.ncbi.nlm.nih.gov/19959533/.

[11] Armstrong AC et al. “Quality Control and Reproducibility in M-Mode, Two-Dimensional, and Speckle Tracking Echocardiography Acquisition and Analysis: The CARDIA Study, Year 25 Examination Experience”. In: *Echocardiography (Mount Kisco, N.Y.)* 32.8 (Aug. 2015), pp. 1233–1240.

[12] Olga Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision* 115 (3 Sept. 2014), pp. 211–252. URL: https://arxiv.org/abs/1409.0575v3.

[13] Andrew Gilbert et al. “Generating Synthetic Labeled Data from Existing Anatomical Models: An Example with Echocardiography Segmentation”. In: *IEEE Transactions on Medical Imaging* (2021).

[14] Hadrien Reynaud et al. “EchoFlow: A Foundation Model for Cardiac Ultrasound Image and Video Generation”. In: *IEEE TRANSACTIONS ON MEDICAL IMAGING* XX (Mar. 2025). URL: https://arxiv.org/pdf/2503.22357.

[15] AC Guyton and JE Hall. *Guyton and Hall Textbook of Medical Physiology*. Fifteenth Edition. 2026, pp. 113–126. URL: https://www.clinicalkey.com/student/content/book/3-s2.0-B9780443111013000093.

[16] Honghe Li et al. “EchoEFNet: Multi-task deep learning network for automatic calculation of left ventricular ejection fraction in 2D echocardiography”. In: *Computers in Biology and Medicine* 156 (Apr. 2023), p. 106705. URL: https://www.sciencedirect.com/science/article/pii/S0010482523001701.

[17] C.M. Rumack et al. “Diagnostic Ultrasound Forth Edition.” In: *Mosby Elsevier (Philadelphia)* 1 (4700 2011), pp. 648, 650.

[18] “Speed of Sound”. In: *Atlas of Ultrasound-Guided Regional Anesthesia* (2019), pp. 3–4.

[19] Jürgen Eggers. “Acute stroke: therapeutic transcranial color duplex sonography.” In: *Frontiers of neurology and neuroscience* 21 (2006), pp. 162–170.

[20] Timothy Ritter. “Single crystal pzn/pt-polymer composites for ultrasound transducer applications”. In: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 47 (4 July 2000), pp. 792–800.

[21] Aladin Carovac, Fahrudin Smajlovic, and Dzelaludin Junuzovic. “Application of Ultrasound in Medicine”. In: *Acta Informatica Medica* 19 (3 2011), p. 168. URL: /pmc/articles/PMC3564184/%20/pmc/articles/PMC3564184/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3564184/.

[22] G York and Y Kim. “Ultrasound processing and computing: Review and future directions”. In: *ecs.csun.edu* (1999). URL: http://www.ecs.csun.edu/~xhang/ECE501_Fall06/Ultrasound%20%5C%20Processing%20%5C%20and%20%5C%20Computing.pdf.

[23] Carol Mitchell et al. “Guidelines for Performing a Comprehensive Transthoracic Echocardiographic Examination in Adults: Recommendations from the American Society of Echocardiography”. In: (2018). URL: https://doi.org/10.1016/j.echo.2018.06.004.

[24] Rebecca T. Hahn et al. “Recommended Standards for the Performance of Transesophageal Echocardiographic Screening for Structural Heart Intervention: From the American Society of Echocardiography”. In: *Journal of the American Society of Echocardiography* 35 (2022), pp. 1–76.

[25] Rebecca T Hahn et al. “ASE GUIDELINES AND STANDARDS Guidelines for Performing a Comprehensive Transesophageal Echocardiographic Examination: Recommendations from the American Society of Echocardiography and the Society of Cardiovascular Anesthesiologists”. In: (2013). URL: http://dx.doi.org/10.1016/j.echo.2013.07.009.

[26] John R. Koza et al. “Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming”. In: *Artificial Intelligence in Design '96* (1996), pp. 151–170. URL: https://link.springer.com/chapter/10.1007/978-94-009-0279-4_9.

[27] Kunihiko Fukushima, Sei Miyake, and Takayuki Ito. “Neocognitron: A Neural Network Model for a Mechanism of Visual Pattern Recognition”. In: *IEEE Transactions on Systems, Man and Cybernetics* SMC-13.5 (1983), pp. 826–834. URL: https://ieeexplore.ieee.org/abstract/document/6313076.

[28] Ronan Collobert and Jason Weston. “A unified architecture for natural language processing”. In: (2008), pp. 160–167. URL: http://wordnet.princeton.edu.

[29] I Sutskever, J Martens, and GE Hinton. “Generating text with recurrent neural networks”. In: *openreview.net* (2011). URL: https://openreview.net/forum?id=SyEoB2-dZH.

[30] G Hinton et al. “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups”. In: *ieeexplore.ieee.org* (2012). URL: https://ieeexplore.ieee.org/abstract/document/6296526/.

[31] Christian Szegedy et al. “Going Deeper with Convolutions”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 07-12-June-2015 (Sept. 2014), pp. 1–9. URL: https://arxiv.org/abs/1409.4842v1.

[32] Y Taigman, M Yang, and MA Ranzato. “Deepface: Closing the gap to human-level performance in face verification”. In: *openaccess.thecvf.com* (2014). URL: http://openaccess.thecvf.com/content_cvpr_2014/html/Taigman_DeepFace_Closing_the_2014_CVPR_paper.html.

[33] J Zhang and C Zong. “Deep Neural Networks in Machine Translation: An Overview.” In: *nlpr-web.ia.ac.cn* (2015). URL: http://nlpr-web.ia.ac.cn/cip/ZongPublications/2015/IEEE-Zhang-8-5.pdf.

[34] A Karpathy and L Fei-Fei. “Deep visual-semantic alignments for generating image descriptions”. In: *cv-foundation.org* (2015). URL: https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Karpathy_Deep_Visual-Semantic_Alignments_2015_CVPR_paper.html.

[35] David Silver et al. “Mastering the game of Go with deep neural networks and tree search”. In: *Nature 2016 529:7587* 529 (7587 Jan. 2016), pp. 484–489. URL: https://www.nature.com/articles/nature16961.

[36] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86 (11 1998), pp. 2278–2323.

[37] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet classification with deep convolutional neural networks”. In: *Communications of the ACM* 60 (6 June 2017), pp. 84–90.

[38] Karen Simonyan and Andrew Zisserman. “VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION”. In: (2015). URL: http://www.robots.ox.ac.uk/.

[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9351 (May 2015), pp. 234–241. URL: https://arxiv.org/abs/1505.04597v1.

[40] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-December (Dec. 2015), pp. 770–778. URL: https://arxiv.org/abs/1512.03385v1.

[41] Gao Huang et al. “Densely Connected Convolutional Networks”. In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* 2017-January (Aug. 2016), pp. 2261–2269. URL: https://arxiv.org/abs/1608.06993v5.

[42] W Zhang et al. “Deep convolutional neural networks for multi-modality isointense infant brain image segmentation”. In: *Elsevier* (2015). URL: https://www.sciencedirect.com/science/article/pii/S1053811914010660.

[43] J Kleesiek et al. “Deep MRI brain extraction: A 3D convolutional neural network for skull stripping”. In: *Elsevier* (2016). URL: https://www.sciencedirect.com/science/article/pii/S1053811916000306.

[44] Guorong Wu et al. “Unsupervised deep feature learning for deformable registration of MR brain images”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8150 LNCS (PART 2 2013), pp. 649–656.

[45] HI Suk, SW Lee, and D Shen. “Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis”. In: *Elsevier* (2014). URL: https://www.sciencedirect.com/science/article/pii/S1053811914005540.

[46] Hoo-Chang Shin et al. “Learning to read chest x-rays: Recurrent neural cascade model for automated image annotation”. In: *openaccess.thecvf.com* (2016). URL: http://openaccess.thecvf.com/content_cvpr_2016/html/Shin_Learning_to_Read_CVPR_2016_paper.html.

[47] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. “Generating Videos with Scene Dynamics”. In: *Advances in Neural Information Processing Systems* (Sept. 2016), pp. 613–621. URL: https://arxiv.org/pdf/1609.02612.

[48] Ligong Han et al. “Show Me What and Tell Me How: Video Synthesis via Multimodal Conditioning”. In: (). URL: https://github.com/snap-research/MMVID.

[49] Heung Il Suk, Seong Whan Lee, and Dinggang Shen. “Latent feature representation with stacked auto-encoder for AD/MCI diagnosis”. In: *Brain Structure and Function* 220 (2 Mar. 2015), pp. 841–859.

[50] Heung-Il Suk and Dinggang Shen. “Deep Learning in Diagnosis of Brain Disorders”. In: (2015), pp. 203–213.

[51] HI Suk et al. “State-space model with deep learning for functional dynamics estimation in resting-state fMRI”. In: *Elsevier* (2016). URL: https://www.sciencedirect.com/science/article/pii/S1053811916000100.

[52] S Pereira, A Pinto, and V Alves. “Brain tumor segmentation using convolutional neural networks in MRI images”. In: *ieeexplore.ieee.org* (2016). URL: https://ieeexplore.ieee.org/abstract/document/7426413/.

[53] G Van Tulder and M De Bruijne. “Combining generative and discriminative representation learning for lung CT analysis with convolutional restricted Boltzmann machines”. In: *ieeexplore.ieee.org* (2016). URL: https://ieeexplore.ieee.org/abstract/document/7401039/.

[54] Q Dou et al. “Automatic detection of cerebral microbleeds from MR images via 3D convolutional neural networks”. In: *ieeexplore.ieee.org* (2016). URL: https://ieeexplore.ieee.org/abstract/document/7403984/.

[55] Dan C. Cireşan et al. “Mitosis detection in breast cancer histology images with deep neural networks”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8150 LNCS (PART 2 2013), pp. 411–418.

[56] H Chen et al. “Deep contextual networks for neuronal structure segmentation”. In: *aaai.org* (2016). URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/viewPaper/11789.

[57] Dong Nie et al. “Fully convolutional networks for multi-modality isointense infant brain image segmentation”. In: *Proceedings - International Symposium on Biomedical Imaging* 2016-June (June 2016), pp. 1342–1345.

[58] Kim Jinsol. *DenseNet (Densely connected convolution networks) - gaussian37*. 2020. URL: https://gaussian37.github.io/dl-concept-densenet/.

[59] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *ICLR 2021 - 9th International Conference on Learning Representations* (Oct. 2020). URL: https://arxiv.org/pdf/2010.11929.

[60] Ashish Vaswani et al. “Attention Is All You Need”. In: (June 2017), p. 1. URL: https://arxiv.org/pdf/1706.03762.

[61] Hanguang Xiao et al. “Transformers in medical image segmentation: A review”. In: *Biomedical Signal Processing and Control* 84 (July 2023), p. 104791. URL: https://www.sciencedirect.com/science/article/pii/S1746809423002240.

[62] Filipe Laitenberger et al. “Cephalometric landmark detection using vision transformers with direct coordinate prediction”. In: *Journal of Cranio-Maxillofacial Surgery* 53.9 (Sept. 2025), pp. 1518–1529. URL: https://pubmed.ncbi.nlm.nih.gov/40603150/.

[63] Hui Li et al. “Towards Accurate Facial Landmark Detection via Cascaded Transformers”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2022-June (Aug. 2022), pp. 4166–4175. URL: https://arxiv.org/pdf/2208.10808.

[64] Wilson Yan et al. “VideoGPT: Video Generation using VQ-VAE and Transformers”. In: (). URL: https://wilson1yan..

[65] Lijun Yu et al. “MAGVIT: Masked Generative Video Transformer”. In: (). URL: https://magvit.cs.cmu.edu..

[66] Zhiying Lu et al. “Bridging the Gap Between Vision Transformers and Convolutional Neural Networks on Small Datasets”. In: (). URL: https://github.com/ArieSeirack/DHVT..

[67] Hoo Chang Shin et al. “Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning”. In: *IEEE transactions on medical imaging* 35 (5 May 2016), pp. 1285–1298. URL: https://pubmed.ncbi.nlm.nih.gov/26886976/.

[68] A Gupta, M Ayhan, and A Maida. “Natural image bases to represent neuroimaging data”. In: *proceedings.mlr.press* (2013). URL: http://proceedings.mlr.press/v28/gupta13b.html?ref=https://githubhelp.com.

[69] JZ Cheng et al. “Computer-aided diagnosis with deep learning architecture: applications to breast lesions in US images and pulmonary nodules in CT scans”. In: *nature.com* (2016). URL: https://www.nature.com/articles/srep24454.

[70] Arnaud Arindra Adiyoso Setio et al. “Pulmonary Nodule Detection in CT Images: False Positive Reduction Using Multi-View Convolutional Networks”. In: *IEEE Transactions on Medical Imaging* 35 (5 May 2016), pp. 1160–1169.

[71] Wei Shen et al. “Multi-scale convolutional neural networks for lung nodule classification”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9123 (2015), pp. 588–599.

[72] Aimee van Wynsberghe. “Sustainable AI: AI for sustainability and the sustainability of AI”. In: *AI and Ethics 2021 1:3* 1 (3 Feb. 2021), pp. 213–218. URL: https://link.springer.com/article/10.1007/s43681-021-00043-6.

[73] Ian Goodfellow et al. “Generative adversarial nets”. In: *Advances in neural information processing systems* 27 (2014).

[74] Jun-Yan Zhu et al. “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks”. In: *Proceedings of the IEEE International Conference on Computer Vision* 2017-October (Mar. 2017), pp. 2242–2251. URL: https://arxiv.org/abs/1703.10593v7.

[75] Youbao Tang et al. “XLSor: A Robust and Accurate Lung Segmentor on Chest X-Rays Using Criss-Cross Attention and Customized Radiorealistic Abnormalities Generation”. In: *Proceedings of Machine Learning Research-XXXX* (Apr. 2019), pp. 1–11. URL: https://arxiv.org/abs/1904.09229v1.

[76] Agisilaos Chartsias et al. “Adversarial Image Synthesis for Unpaired Multi-Modal Cardiac Data”. In: ().

[77] Jafari MH et al. “Cardiac point-of-care to cart-based ultrasound translation using constrained CycleGAN”. In: *International journal of computer assisted radiology and surgery* 15 (5 May 2020), pp. 877–886. URL: https://pubmed.ncbi.nlm.nih.gov/32314226/.

[78] Amir H. Abdi, Teresa Tsang, and Purang Abolmaesumi. “GAN-enhanced Conditional Echocardiogram Generation”. In: (Nov. 2019). URL: https://arxiv.org/abs/1911.02121v2.

[79] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. “Unsupervised Image-to-Image Translation Networks”. In: (2017). URL: https://github.com/mingyuliutw/unit..

[80] Xun Huang et al. “Multimodal Unsupervised Image-to-Image Translation”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11207 LNCS (Apr. 2018), pp. 179–196. URL: https://arxiv.org/abs/1804.04732v2.

[81] Taesung Park et al. “Contrastive Learning for Unpaired Image-to-Image Translation”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 12354 LNCS (July 2020), pp. 319–345. URL: https://arxiv.org/abs/2007.15651v3.

[82] Prannay Khosla et al. “Supervised Contrastive Learning”. In: (2021). URL: https://t.ly/supcon.

[83] Jascha Sohl-Dickstein et al. “Deep Unsupervised Learning using Nonequilibrium Thermodynamics”. In: ().

[84] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising Diffusion Probabilistic Models”. In: *Advances in Neural Information Processing Systems* 2020-December (June 2020). URL: https://arxiv.org/pdf/2006.11239.

[85] Alex Nichol and Prafulla Dhariwal. “Improved Denoising Diffusion Probabilistic Models”. In: *Proceedings of Machine Learning Research* 139 (Feb. 2021), pp. 8162–8171. URL: https://arxiv.org/pdf/2102.09672.

[86] Chitwan Saharia et al. “Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 36479–36494.

[87] Robin Rombach et al. “High-Resolution Image Synthesis with Latent Diffusion Models”. In: (Apr. 2022). URL: http://arxiv.org/abs/2112.10752.

[88] David Stojanovski and Alberto Gomez. “Echo from noise: synthetic ultrasound image generation using diffusion models for real image segmentation”. In: (). URL: https://zenodo.org/record/7921055#.ZGYS.

[89] Prafulla Dhariwal and Alex Nichol. “Diffusion Models Beat GANs on Image Synthesis”. In: *Advances in Neural Information Processing Systems* 11 (May 2021), pp. 8780–8794. URL: https://arxiv.org/pdf/2105.05233.

[90] Jonathan Ho and Tim Salimans. “Classifier-Free Diffusion Guidance”. In: (July 2022). URL: https://arxiv.org/pdf/2207.12598.

[91] Jiaming Song, Chenlin Meng, and Stefano Ermon. “Denoising Diffusion Implicit Models”. In: *ICLR 2021 - 9th International Conference on Learning Representations* (Oct. 2020). URL: https://arxiv.org/pdf/2010.02502.

[92] Tero Karras et al. “Elucidating the Design Space of Diffusion-Based Generative Models”. In: (2022). URL: http://arxiv.org/abs/2206.00364.

[93] Yang Song et al. “Score-Based Generative Modeling through Stochastic Differential Equations”. In: *ICLR 2021 - 9th International Conference on Learning Representations* (Nov. 2020). URL: https://arxiv.org/pdf/2011.13456.

[94] Yang Song et al. “Consistency Models”. In: *Proceedings of Machine Learning Research* 202 (Mar. 2023), pp. 32211–32252. URL: https://arxiv.org/pdf/2303.01469.

[95] Tim Salimans and Jonathan Ho. “Progressive Distillation for Fast Sampling of Diffusion Models”. In: *ICLR 2022 - 10th International Conference on Learning Representations* (Feb. 2022). URL: https://arxiv.org/pdf/2202.00512.

[96] Andreas Blattmann et al. “Align your Latents: High-Resolution Video Synthesis with Latent Diffusion Models”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2023-June (Apr. 2023), pp. 22563–22575. URL: https://arxiv.org/pdf/2304.08818.

[97] William Harvey et al. “Flexible Diffusion Modeling of Long Videos”. In: ().

[98] Jonathan Ho et al. “Video Diffusion Models”. In: *Advances in Neural Information Processing Systems* 35 (Apr. 2022). URL: https://arxiv.org/pdf/2204.03458.

[99] Daquan Zhou et al. “MagicVideo: Efficient Video Generation With Latent Diffusion Models”. In: (Nov. 2022). URL: https://arxiv.org/pdf/2211.11018.

[100] Hadrien Reynaud et al. “Feature-Conditioned Cascaded Video Diffusion Models for Precise Echocardiogram Synthesis”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 14229 LNCS (2023), pp. 142–152.

[101] Yaron Lipman et al. “Flow Matching for Generative Modeling”. In: *11th International Conference on Learning Representations, ICLR 2023* (Oct. 2022). URL: https://arxiv.org/pdf/2210.02747.

[102] Yang Jin et al. “Pyramidal Flow Matching for Efficient Video Generative Modeling”. In: *13th International Conference on Learning Representations, ICLR 2025* (Oct. 2024), pp. 10672–10696. URL: https://arxiv.org/pdf/2410.05954.

[103] Lea Bogensperger et al. “FlowSDF: Flow Matching for Medical Image Segmentation Using Distance Transforms”. In: *International Journal of Computer Vision 2025 133:7* 133.7 (Mar. 2025), pp. 4864–4876. URL: https://link.springer.com/article/10.1007/s11263-025-02373-y.

[104] Milad Yazdani et al. “Flow Matching for Medical Image Synthesis: Bridging the Gap Between Speed and Quality”. In: (Mar. 2025). URL: https://arxiv.org/pdf/2503.00266.

[105] Zhou Wang et al. “Image quality assessment: From error visibility to structural similarity”. In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612.

[106] Stéfan Van Der Walt et al. “Scikit-image: Image processing in python”. In: *PeerJ* 2014.1 (2014). URL: https://github.com/scikit-image/scikit-image.

[107] Richard Zhang et al. “The Unreasonable Effectiveness of Deep Features as a Perceptual Metric”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2018), pp. 586–595.

[108] Martin Heusel et al. “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium”. In: *Advances in Neural Information Processing Systems* 2017-December (2017), pp. 6627–6638.

[109] Thomas Unterthiner et al. “Towards Accurate Generative Models of Video: A New Metric & Challenges”. In: (Dec. 2018). URL: https://arxiv.org/pdf/1812.01717.

[110] Darcy Mason and scaramallion. *Pydicom*. URL: https://github.com/pydicom/pydicom.

[111] G. Bradski. “The OpenCV Library”. In: *Dr. Dobb’s Journal of Software Tools* (2000).

[112] Marina Strocchi et al. “A publicly available virtual cohort of four-chamber heart meshes for cardiac electro-mechanics simulations”. In: (2020). URL: https://doi.org/10.1371/journal.pone.0235145.

[113] Rodero Cristobal et al. “Linking statistical shape models and simulated function in the healthy adult human heart”. In: *PLOS Computational Biology* 17 (4 Nov. 2021), pp. 1–28. URL: https://doi.org/10.1371/journal.pcbi.1008851.

[114] *CGAL/cgal: The public CGAL repository, see the README below*. URL: https://github.com/CGAL/cgal.

[115] Andrew Gilbert. *GilbertSupplementaryMaterial.pdf*. Jan. 2021. URL: https://www.researchgate.net/publication/348760427_SupplementaryMaterialpdf.

[116] Alhassan Mumuni, Fuseini Mumuni, and Nana Kobina Gerrar. “A Survey of Synthetic Data Augmentation Methods in Machine Vision”. In: (). URL: http://doi.org/10.1007/s11633-022-1411-7.

[117] Jun-Yan Zhu et al. “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks”. In: *Proceedings of the IEEE International Conference on Computer Vision* 2017-October (Mar. 2017), pp. 2242–2251. URL: https://arxiv.org/abs/1703.10593v7.

[118] Youbao Tang et al. “XLSor: A Robust and Accurate Lung Segmentor on Chest X-Rays Using Criss-Cross Attention and Customized Radiorealistic Abnormalities Generation”. In: *Proceedings of Machine Learning Research* (Apr. 2019), pp. 1–11. URL: https://arxiv.org/abs/1904.09229v1.

[119] Agisilaos Chartsias et al. “Adversarial Image Synthesis for Unpaired Multi-Modal Cardiac Data”. In: ().

[120] M Alessandrini et al. “A Pipeline for the Generation of Realistic 3D Synthetic Echocardiographic Sequences: Methodology and Open-Access Database”. In: *IEEE TRANSACTIONS ON MEDICAL IMAGING* 34.7 (2015).

[121] Amir H. Abdi, Teresa Tsang, and Purang Abolmaesumi. “GAN-enhanced Conditional Echocardiogram Generation”. In: (Nov. 2019). URL: https://arxiv.org/abs/1911.02121v2.

[122] Taesung Park et al. “Contrastive Learning for Unpaired Image-to-Image Translation”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 12354 LNCS (July 2020), pp. 319–345.

[123] Raphael Gontijo-Lopes, Sylvia J Smullin, and Ethan Dyer. “Affinity and Diversity: Quantifying Mechanisms of Data Augmentation”. In: (2020).

[124] Daniel T Ginat et al. “Cardiac imaging: Part 1, MR pulse sequences, imaging planes, and basic anatomy”. In: *American Journal of Roentgenology* 197.4 (Oct. 2011), pp. 808–815. URL: /doi/pdf/10.2214/AJR.10.7231?download=true.

[125] Andrew Gilbert et al. “Generating Synthetic Labeled Data from Existing Anatomical Models: An Example with Echocardiography Segmentation”. In: *IEEE Transactions on Medical Imaging* (2021).

[126] Abhishek Dutta and Andrew Zisserman. “The VIA annotation software for images, audio and video”. In: *MM 2019 - Proceedings of the 27th ACM International Conference on Multimedia* (Oct. 2019), pp. 2276–2279.

[127] Maximilian Seitzer. “pytorch-fid: FID Score for PyTorch”. In: *GitHub* (Aug. 2020). URL: https://github.com/mseitzer/pytorch-fid.

[128] Fabian Isensee et al. “nnU-Net: Self-adapting Framework for U-Net-Based Medical Image Segmentation”. In: *Informatik aktuell* (Sept. 2018), p. 22.

[129] Lennart Bargsten and Alexander Schlaefer. “SpeckleGAN: a generative adversarial network with an adaptive speckle layer to augment limited training data for ultrasound image processing”. In: *International Journal of Computer Assisted Radiology and Surgery* 15.9 (Sept. 2020), pp. 1427–1436.

[130] Thomas Mensink et al. “Factors of Influence for Transfer Learning across Diverse Appearance Domains and Task Types”. In: ().

[131] Hongbo Zhao et al. “Continual Forgetting for Pre-trained Vision Models”. In: (). URL: https://github.com/bjzhb666/GS-LoRA..

[132] Yuexiang Zhai et al. “Investigating the Catastrophic Forgetting in Multimodal Large Language Models”. In: *Proceedings of Machine Learning Research* 234 (Sept. 2023), pp. 202–227. URL: https://arxiv.org/pdf/2309.10313.

[133] Edward Hu et al. “LORA: LOW-RANK ADAPTATION OF LARGE LAN-GUAGE MODELS”. In: ().

[134] Emmanuel Oladokun et al. “Transesophageal Echocardiography Generation using Anatomical Models”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 14379 LNCS (2024), pp. 43–52.

[135] Theresa A. McDonagh et al. “2023 Focused Update of the 2021 ESC Guidelines for the diagnosis and treatment of acute and chronic heart failure”. In: *European heart journal* 44.37 (Oct. 2023), pp. 3627–3639. URL: https://pubmed.ncbi.nlm.nih.gov/37622666/.

[136] Amirata Ghorbani et al. “Deep learning interpretation of echocardiograms”. In: *NPJ digital medicine* 3.1 (Dec. 2020). URL: https://pubmed.ncbi.nlm.nih.gov/31993508/.

[137] Thomas Komanek et al. “Quantification of left ventricular ejection fraction and cardiac output using a novel semi-automated echocardiographic method: a prospective observational study in coronary artery bypass patients”. In: *BMC anesthesiology* 23.1 (Dec. 2023). URL: https://pubmed.ncbi.nlm.nih.gov/36855077/.

[138] Ziru Qin and Qun Dai. “A 3D-CNN and multi-loss video prediction architecture”. In: *Applied Intelligence* 55.6 (Apr. 2025), pp. 416–. URL: https://link.springer.com/article/10.1007/s10489-025-06328-1.

[139] Omer Bar-Tal. “Lumiere: A Space-Time Diffusion Model for Video Generation Citation in BibTeX format Lumiere: A Space-Time Diffusion Model for Video Generation”. In: ().

[140] Xin Ma et al. “Latte: Latent Diffusion Transformer for Video Generation”. In: *Transactions on Machine Learning Research* 2025-March (Jan. 2024), pp. 1–20. URL: https://arxiv.org/pdf/2401.03048.

[141] Xinrui Zhou et al. “HeartBeat: Towards Controllable Echocardiography Video Synthesis with Multimodal Conditions-Guided Diffusion Models”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 15007 LNCS (June 2024), pp. 361–371. URL: https://arxiv.org/pdf/2406.14098.

[142] Zhengyang Geng et al. “Mean Flows for One-step Generative Modeling”. In: (May 2025). URL: https://arxiv.org/pdf/2505.13447.

[143] Zhengyang Geng et al. “Consistency Models Made Easy”. In: *13th International Conference on Learning Representations, ICLR 2025* (Oct. 2024), pp. 96638–96665. URL: http://arxiv.org/abs/2406.14548.

[144] Huijie Zhang et al. “AlphaFlow: Understanding and Improving MeanFlow Models”. In: (Oct. 2025). URL: https://arxiv.org/pdf/2510.20771.

[145] Zhengyang Geng et al. “Improved Mean Flows: On the Challenges of Fastforward Generative Models”. In: ().

[146] Zhe Huang et al. “TMED 2: A Dataset for Semi-Supervised Classification of Echocardiograms”. In: (). URL: https://echonet.github.io/dynamic/.

[147] Alex Morehead. *JVP Flash Attention*. Sept. 2025. URL: https://github.com/amorehead/jvp_flash_attention.

[148] Van Phi Nguyen et al. “Training-Free Condition Video Diffusion Models for single frame Spatial-Semantic Echocardiogram Synthesis”. In: (Aug. 2024). URL: https://arxiv.org/abs/2408.03035v2.

[149] Nima Kondori et al. “ControlEchoSynth: Boosting Ejection Fraction Estimation Models via Controlled Video Diffusion”. In: (Aug. 2025). URL: https://arxiv.org/pdf/2508.17631.

[150] Emmanuel Oladokun et al. “EchoLVFM: One-Step Video Generation via Latent Flow Matching for Echocardiogram Synthesis”. In: (Mar. 2026). URL: https://arxiv.org/pdf/2603.13967.