Imperial College London
Department of Computing, Department of Surgery and Cancer

Robust Representation Learning with Topological and Geometric Constraints

Ainkaran Santhirasekaram

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy from Imperial College London, February 2024

I, Ainkaran Santhirasekaram, hereby declare that the work described in this thesis is my own, except where specifically acknowledged.

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

Under this license, you may copy and redistribute the material in any medium or format for both commercial and non-commercial purposes. You may also create and distribute modified versions of the work. This on the condition that you credit the author.

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

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

Acknowledgements

My PhD journey has been immensely satisfying where I have had the opportunity to work with a number of fantastic human beings. Throughout my time as a PhD student, I fulfilled my dream of working and contributing to a highly technical field in artificial intelligence. I was allowed to explore a myriad of interesting topics in mathematics which included algebraic topology, non-euclidean geometry and group theory and apply this to deep learning.

None of this would not have been possible without my supervisors. Firstly, I would like to thank my supervisors Professor Andrea Rockall, Professor Ben Glocker and Mr Mathias Winkler who gave me the opportunity to do this PhD as well as guide me during this Marathon. I would particularly like to mention Professor Ben Glocker who is an outstanding mentor who gave me the encouragement and freedom to pursue my passion.

A PhD would not complete without regular intellectual debate around the table. I would therefore like to thank the students and staff of BioMedia who provided me with much inspiration through thoughtful discussion.

Lastly, and most importantly I would like to thank my parents, without whom none of this would be possible.

Dedication

I dedicate this PhD to my late cousin, Ahilan who died of cancer 8 years ago and always encouraged me to pursue my passion.

'Just because we can't find a solution it doesn't mean that there isn't one'

Andrew Wiles

Abstract

Robust representation learning in medical imaging is crucial for the reliable application of deep learning models in critical tasks like clinical decision-making. The vulnerability of deep learning models to subtle perturbations, particularly when facing distributional shifts in test data not encountered during training, is well-documented. To enhance model robustness, common approaches involve data augmentation, adversarial training, and self-supervised learning strategies.

In this thesis, we will explore the importance of geometry and topology of the latent space in building robust representation for medical imaging. We start with restricting the latent space to a finite set of components using vector quantisation for downstream segmentation and classification tasks. We claim that imposing a compositional structure on the latent space enables a more robust representation. Next, we will explore the natural curvature in which to embed these components for segmentation. Specifically, we impose hierarchical compositionality in a latent space equipped with hyperbolic geometry to compose the segmentation output to improve robustness. We then explore an alternative geometric perspective by constraining the latent space to a dictionary of group equivariant shape components which is sampled to construct the segmentation output. We analyse equivariance to various groups and demonstrate how the order of the group affects model robustness. We follow up our exposition into robust segmentation with a discrete latent space by sampling the dictionary of shape components such that topology is preserved using persistent homology and cellular sheaf theory. We show the advantage of using cellular sheaf theory over persistent homology to extract richer geometric information. Finally, we show the importance of curvature for embedding symbols in neurosymbolic reasoning. Specifically, we develop robust hierarchical explanations in hyperbolic space for deep discriminative models.

We demonstrate the effectiveness of our methods to improve model robustness under various perturbations and in single domain generalisation tasks.

Contents

Declaration of Originality i

Copyright Statement i

Acknowledgements iii

Abstract vii

1 Introduction 1

1.1 Motivation 1

1.2 Research Objectives 3

1.3 Thesis Outline and Publications 4

2 Background Theory 10

2.1 Convolutional Neural Networks 11

2.1.1 Convolutional Layer 12

2.1.2 Activation Function 12

2.1.3 Normalisation Layer 13

2.1.4 Pooling Layer 14

2.1.5 Classification 14

2.1.6 Segmentation 14

2.2 Vision Transformers 15

2.2.1 Positional Encoding 17

2.2.2 Multi-Headed Self Attention 18

2.3 Training Deep Learning Models 19

2.4 Discrete Representation Learning . . . 20
2.5 Deep Learning in Medical Imaging . . . 22
2.5.1 Medical Imaging . . . 22
2.5.2 Magnetic Resonance Imaging . . . 23
2.5.3 Prostate . . . 24
2.6 Robust Representation Learning . . . 25
2.6.1 Robustness framework . . . 25
2.6.2 Domain Generalisation . . . 27
2.6.3 Robustness Methods . . . 28
2.7 Group Theory and Equivariance . . . 29
2.8 Topology and Geometry . . . 32
2.8.1 Introduction . . . 33
2.8.2 Topological Space . . . 34
2.8.3 Complexes . . . 35
2.8.4 Persistent Homology . . . 40
2.8.5 Riemannian Geometry . . . 42
2.8.6 Hyperbolic Geometry . . . 44
2.9 Categories and Sheaves . . . 49
2.9.1 Category Theory . . . 49
2.9.2 Sheaf Theory . . . 53
2.10 Measure Theory . . . 56

3 Datasets **61**

3.1 External Datasets . . . 61
3.1.1 Beyond the Cranial Vault (BTCV) dataset . . . 61
3.1.2 NCI-ISBI13 Challenge Dataset . . . 62
3.1.3 Segmentation Decathlon (Prostate) . . . 62
3.1.4 ProstateX challenge . . . 62
3.1.5 NIH chest X-ray Dataset . . . 64
3.1.6 JSRT chest X-ray Dataset . . . 64

3.1.7 M & M cardiac dataset . . . 64
3.1.8 MNIST . . . 65
3.1.9 AFHQ . . . 65
3.1.10 STL10 . . . 66
3.1.11 MIMIC . . . 66
3.2 Internal Dataset . . . 66
3.2.1 Dataset Description . . . 66
3.2.2 Anonymisation and Ethics . . . 68

4 Robust Disease Classification on MRI using Transformers with Discrete Representations 69

4.1 Introduction . . . 69
4.2 Related Work . . . 70
4.2.1 Advancements on the Vision Transformer . . . 70
4.2.2 Robustness of the Vision Transformer . . . 72
4.2.3 The Vision Transformer in medical imaging . . . 74
4.3 Motivation . . . 75
4.4 Methods . . . 76
4.4.1 Stage 1 . . . 76
4.4.2 Stage 2 . . . 77
4.4.3 Probabilistic Perspective . . . 79
4.5 Dataset Preparation . . . 81
4.5.1 Pre-processing and Augmentation . . . 81
4.5.2 Datasets . . . 81
4.6 Experimental setup . . . 84
4.6.1 Model . . . 84
4.6.2 Model Ablations . . . 84
4.6.3 Single Domain Generalisation . . . 85
4.6.4 Perturbation Experiments . . . 86
4.6.5 Training and Evaluation . . . 87

4.7 Results . . . 88
4.7.1 Ablations . . . 88
4.7.2 Single Domain Generalisation . . . 90
4.7.3 Perturbation Experiments . . . 92
4.8 Conclusion . . . 94

5 Vector Quantisation for Robust Medical Segmentation 95
5.1 Introduction . . . 95
5.2 Methods . . . 96
5.2.1 Robustness and Network Assumptions . . . 96
5.2.2 Quantisation for Robustness . . . 97
5.2.3 Perturbation Bounds . . . 98
5.2.4 Implementation Details and Data . . . 100
5.3 Experiments . . . 101
5.3.1 Codebook Study . . . 101
5.3.2 Domain Shift Study . . . 102
5.3.3 Perturbation Study . . . 103
5.4 Conclusion . . . 106

6 Hierarchical Compositionality in Hyperbolic Space for Robust Medical Image Segmentation 107
6.1 Introduction . . . 107
6.2 Related Work . . . 108
6.3 Methods . . . 109
6.3.1 Base Component Learning . . . 109
6.3.2 Hyperbolic embeddings . . . 111
6.3.3 Datasets and Training . . . 112
6.4 Experiments . . . 113
6.4.1 Perturbation experiments . . . 113
6.4.2 Single Domain Generalisation . . . 116
6.5 Conclusion . . . 118

7 A Geometric Approach to Robust Medical Segmentation 119

7.1 Introduction . . . 119

7.2 Group Convolutions Preliminaries . . . 121

7.3 Related work . . . 126

7.4 Methods . . . 127

7.4.1 Method 1: Contrastive Learning . . . 127

7.4.2 Method 2: Group Equivariant Convolutional Kernels . . . 135

7.5 Experiments . . . 137

7.5.1 Datasets and Training . . . 137

7.5.2 Method 1 Experiments . . . 138

7.5.3 Method 2 Experiments . . . 142

7.6 Conclusion . . . 147

8 Topology Preserving Compositionality for Robust Medical Segmentation 149

8.1 Introduction . . . 149

8.2 Background . . . 150

8.2.1 Related Work . . . 150

8.3 Methods . . . 151

8.3.1 Compositionality for Segmentation . . . 151

8.3.2 Topology Preserving Compositionality . . . 152

8.4 Experiments . . . 156

8.4.1 Datasets and Training . . . 156

8.4.2 Perturbation Experiments . . . 156

8.4.3 Single Domain Generalisation . . . 158

8.4.4 Model comparison . . . 160

8.4.5 Ablation Studies . . . 161

8.5 Discussion . . . 162

9 Sheaf Theoretic Perspective for Robust Segmentation 164

9.1 Introduction . . . 164

9.2 Related work . . . 165

9.3 Methods . . . 166

9.3.1 Cellular Sheaf Over a Poset . . . 166

9.3.2 Cellular Sheaves for Composing Shapes . . . 167

9.3.3 A Measure Theoretic Approach for Using Cellular Sheaves for Composing Shapes . . . 170

9.4 Experiments . . . 177

9.4.1 Topology Experiments . . . 177

9.4.2 SDG Comparison Experiments . . . 180

9.4.3 SDG Tumour Segmentation Experiments . . . 182

9.5 Conclusion . . . 185

10 Robust Hierarchical Symbolic Explanations in Hyperbolic Space for Image Classification 186

10.1 Introduction . . . 186

10.2 Related Work . . . 188

10.3 Preliminaries . . . 189

10.3.1 Hyperbolic Transformations . . . 189

10.3.2 Knowledge Tree . . . 189

10.3.3 Symbolic Relations . . . 191

10.4 Methods . . . 192

10.4.1 Symbol formation . . . 192

10.4.2 Hyperbolic Symbol Abstraction . . . 193

10.4.3 Explanations and Visual Semantics . . . 195

10.5 Training details and architectures . . . 196

10.5.1 Discrete surrogate model . . . 196

10.5.2 Binary Weights . . . 197

10.5.3 Decoder . . . 198

10.5.4 Pre-trained classifiers . . . 198

10.6 Experiments . . . 199
10.6.1 Qualitative Experiments . . . 199
10.6.2 Robustness Experiments . . . 201
10.6.3 Hyperbolic vs Euclidean Experiments . . . 203
10.6.4 Codebook Ablations . . . 205
10.7 Conclusion . . . 206

11 Conclusions **207**
11.1 Research Objectives . . . 207
11.1.1 Research Objective 1 . . . 207
11.1.2 Research Objective 2 . . . 207
11.1.3 Research Objective 3 . . . 208
11.1.4 Research Objective 4 . . . 208
11.1.5 Research Objective 5 . . . 208
11.2 Summary of Contributions . . . 209

Bibliography **213**

List of Tables

3.1 T2 weighted image acquisition parameters for the BMC and RUNMC dataset from the NCI-ISBI13 Challenge. 62

3.2 T2 and diffusion-weighted image acquisition parameters for the ProstateX dataset. ADC maps in both datasets are calculated with b-values; 50, 400 and 800. 63

3.3 Vendor description for each site in the M&M dataset. 65

3.4 T2 and diffusion-weighted image acquisition parameters for the internal dataset. ADC maps are calculated with b-values; 50, 400 and 800. 68

4.1 Train and test splits showing the number of scans from the internal and external dataset for each risk group. 82

4.2 T2 and diffusion-weighted image acquisition parameters for the 1.5T and 3T datasets. ADC maps in both datasets are calculated with b-values; 50, 400 and 800. 83

4.3 T2 and diffusion-weighted image acquisition parameters for the training and test datasets for the control experiments. ADC maps in both datasets are calculated with b-values; 50, 400 and 800. 83

4.4 VQ-VAE architecture. 84

4.5 Hybrid Vision Transformer architecture. 86

4.6 Comparison of model performance on the test set using 5 evaluation metrics ± standard deviation for different Euclidean codebook sizes. The metrics were averaged across the three classes. 88

4.7 Comparison of model performance on the test set using 5 evaluation metrics ± standard deviation for 6 different variations of our model. The results for no acquisition shift where the training and test dataset are both acquired on a 3T scanner. In the acquisition shift results, we show the results averaged for 1.5T to 3T and 3T to 1.5T domain shift experiments. The metrics were averaged across the three classes. The metrics were averaged across the three classes. 89

4.8 Comparison of 3 different model architectures with 3 different SDG methods with our discrete representation approach. We evaluate performance with 5 evaluation metrics ± standard deviation. We show the results averaged for the 1.5T to 3T and 3T to 1.5T domain shift experiments. The metrics were averaged across the three classes. 90

4.9 AUC scores ± standard deviation of 3 different model architectures with 3 different SDG methods with our discrete representation approach. We show the results averaged for 1.5T to 3T and 3T to 1.5T domain shift experiments. 92

4.10 The mean AUC ± standard deviation for each risk group comparing 4 model types to our model under various perturbations in the input space. The baseline refers to no perturbations applied. The results shown are averaged across all perturbation parameters i.e. overall noise levels for Gaussian, Poisson and Salt and Pepper (S&P) noise. 92

4.11 The relative corruption error (%) based on AUC of our hyper-spherical codebook 2-stage discrete model relative to the 4 other models in table 4.10. 94

5.1 Mean r ± 1 standard deviation for all 5 datasets. 101

5.2 Mean Dice and HD95 ± standard deviation on the validation sets from the same domain and the test set which is derived from another domain. The arrow represents the domain shift. 102

5.3 Average latent space variance in the UNet and VQ-UNet for all three datasets. 103

5.4 Gaussian noise perturbation results on all 3 datasets. 106

5.5 Salt and Pepper perturbation results on all 3 datasets. 106

5.6 Poisson noise perturbation results on all 3 datasets. 106

6.1 The mean Dice score and Hausdorff distance ± standard deviation under various perturbations in the input space for the Abdominal dataset of the UNet and TransUNet with hierarchical compositionality imposed with three different types of curvature constraints. We look at Euclidean (EucC: 0 curvature), hyper-spherical (SphC: +1 curvature) and hyperbolic (HypC: -1 curvature) geometry for hierarchical compositionality. We also show the results of the UNet and TransUNet with no constraints imposed. 114

6.2 Relative corruption error (%) in terms of the Dice score of our hierarchical compositional model under different curvature constraints as well as the VQ-UNet and VQ-TransUNet compared to the baseline UNet and TransUNet. 115

6.3 Dice score and maximum Hausdorff distance(HD) ± standard deviation comparing our method to various other SDG methods. This table shows the segmentation results for the prostate zones averaged across both single domain shift experiments (*BMC* → *RUNMC* and *RUNMC* → *BMC*). In these experiments all methods are augmented with random rotations and flips in both the horizontal and vertical axis. BigAug however uses a more aggressive augmentation pipeline including a combination of Gaussian noise, blur, contrast changes and random affine transformations as designed exactly by [ZWY+20]. 116

6.4 Hyper-parameters used in our experiments. The 'size' is the number of components in the dictionary. 'Channels' is the number of embedding features in the model bottleneck. 117

7.1 The angular component basis of SO(2)-steerable kernels that satisfy the irreducible representation kernel constraint. These are constructed for various combinations of input and output field irreducible representations ψ_n. 125

7.2 Dice score and Hausdorff distance(HD) ± standard deviation for different order dihedral group shape equivariant models trained via contrastive learning approach. This table shows the results for the prostate dataset. 139

7.3 Dice score and Hausdorff distance(HD) ± standard deviation for different order dihedral group shape equivariant models trained via contrastive learning. This table shows the results for the cardiac dataset. 140

7.4 The average Dice and maximum Hausdorff distance score ± standard deviation comparing the D8h and D12h group shape equivariant segmentation models to various other SDG methods. This table shows the results for the prostate dataset. 141

7.5 The average Dice and maximum Hausdorff distance scores ± standard deviation comparing the D8h and D12h group shape equivariant segmentation models trained via contrastive learning to various other SDG methods. This table shows the results for the cardiac dataset. 141

7.6 The average Dice and maximum Hausdorff distance score ± standard deviation for different order dihedral group shape equivariant models trained via the kernel constraint method. This table shows the results for the prostate dataset. 143

7.7 The average Dice and maximum Hausdorff distance score ± standard deviation for different order dihedral group shape equivariant models trained via the kernel constraint method. This table shows the results for the cardiac dataset. 144

7.8 The average Dice and maximum Hausdorff distance scores ± standard deviation comparing the D16 and C16 group shape equivariant segmentation models trained via the kernel constraint method to various other SDG methods. This table shows the results for the prostate dataset. . . . 146

7.9 The average Dice and maximum Hausdorff distance scores ± standard deviation comparing the D16 and C16 group equivariant segmentation models trained via the kernel constraint method to various other SDG methods. This table shows the results for the cardiac dataset. 146

7.10 The average Dice and maximum Hausdorff distance scores ± standard deviation demonstrating each component of our method. This table shows the results for the cardiac dataset. 147

8.1 The mean Dice score and Betti error ± standard deviation before and after TPC is applied to 3 segmentation models under various perturbations in the input space. The results for the Abdominal dataset are shown. The baselines refer to no perturbations applied. 157

8.2 The relative corruption error (RCE) (%) in terms of the Dice score and Betti error after TPC is included in the UNet, nnUNet and Swin-UNetr. . 157

8.3 Dice score and Betti error ± standard deviation comparing TPC to various other SDG methods. This table shows the results for the prostate dataset. 159

8.4 Dice score and Betti error ± standard deviation comparing TPC to various other SDG methods. This table shows the results for the cardiac dataset. 159

8.5 The average Dice score and Betti error before and after TPC is applied to 3 segmentation models after testing on the target domains. The results for the prostate dataset are shown. 160

8.6 The average Dice score and Betti error before and after TPC is applied to 3 segmentation models after testing on the target domains. The results for the prostate dataset are shown. 160

8.7 Hyper-parameters used in our experiments. The 'size' is the number of components in the dictionary. The 'dimensionality' is the dimension of the codebook components. 'Channels' is the number of embedding features in the model bottleneck. 161

9.1 Dice score,Betti error and maximum Hausdorff distance (HD) ± standard deviation comparing 'TopoLoss' to our cellular sheaf-based loss with different discretisations of the segmentation output and label. This table shows the results for the prostate dataset. 178

9.2 Dice score, Betti error and maximum Hausdorff distance (HD) ± standard deviation comparing 'TopoLoss' to our cellular sheaf-based loss with different discretisations of the segmentation output and label. This table shows the results for the cardiac dataset. 179

9.3 Dice score, Betti error and maximum Hausdorff distance (HD) ± standard deviation comparing the D8h shape equivariant and D8h shape equivariant+SheafLoss segmentation models trained via contrastive learning to various other SDG methods. This table shows the results for the prostate dataset. 181

9.4 Dice score, Betti error and maximum Hausdorff distance (HD) ± standard deviation comparing the D8h shape equivariant and D8h shape equivariant+SheafLoss segmentation models trained via contrastive learning to various other SDG methods. This table shows the results for the cardiac dataset. 181

9.5 Dice score, Betti Error and maximum Hausdorff distance (HD) ± standard deviation comparing the D8h shape equivariant+SheafLoss segmentation model trained via contrastive learning to various other SDG methods. This table shows the segmentation results for the prostate zones. 183

9.6 The average Dice score, Betti error, maximum Hausdorff distance (HD), sensitivity, specificity and positive predictive value (PPV) ± standard deviations using our method, several SDG methods and the nnUNet for tumour segmentation. 184

10.1 Average variance in the heatmaps generated by various post-hoc explainability methods under Gaussian addition. 201

10.2 Average variance in the heatmaps generated by various post-hoc explainability methods under Salt and Pepper (S&P) noise addition. 201

10.3 Average variance in the heatmaps generated by various post-hoc explainability methods under Poisson noise addition. 202

10.4 Knowledge distillation accuracy of different dimensional Euclidean and Poincaré embeddings. 205

10.5 Average Dice score overlap between the visual semantics of symbols sampled from ζ^0 for different dimensional Euclidean and Poincaré embeddings 205

10.6 Codebook ablations to develop the most sparse knowledge tree. 206

List of Figures

1.1 This figure provides a thesis overview. We start with the initial problem of robust representation learning and tackle this with discrete representation learning (purple box) with curvature based constraints (blue boxes), group equivariant based constraints (orange box) and topological constraints (green box). We compare our approach with the standard methods in the literature to improve model robustness (red boxes). . . . 3

2.1 Example UNet architecture. . . . 16

2.2 Overview of the vision transformer architecture. This figure is taken from [DBK+20]. . . . 17

2.3 Overview of the VQ-VAE architecture. . . . 21

2.4 The above DAG demonstrates that task T involves separating the classes of interest C from population factors E_p and environmental factors E_d. Scenes, formed by sampling from these variables, are passed through a sensor S, before being rendered R to form the acquisition process which produces the final image X. Labels Y are conditioned on both the task and the resulting image. This figure is adapted from [DSSU21]. . . . 26

2.5 A hierarchy demonstrating the increasing constraints imposed which connect a topological space with a Riemannian manifold. This figure is adapted from [Bod23]. . . . 33

2.6 A continuous maps between topological spaces X and Y where the the pre-image of $U \subseteq Y$, is in X. This figure is adapted from [Bod23]. . . . 35

2.7 We show an example of the k-simplex from $k = 0$ which is a point to $k = 3$ which is the tetrahedra. . . . 36

2.8 An example of a simplicial complex $\mathcal{K}$. . . . 37

2.9 Visualisation of a chain complex with its boundary and cycle groups. The image (green line) and kernel (blue line) of the boundary operators are demonstrated. . . . 39

2.10 Example of persistence diagrams (right) calculated from the softmax model output and segmentation label for the heart myocardium. The x and y axis range from 0 to 1. . . . 41

2.11 Here we provide a visualisation of the two-shift hyperboloid in green with the Poincaré disc visualised in blue with a purple geodesic. 45

2.12 Here we show the logarithmic mapping (hyperboloid to tangent space) and exponential mapping(tangent space to hyperboloid). The tangent space is demonstrated on the origin of the hyperboloid. 46

2.13 Here we show a pre-sheaf over the topological space X where open subsets A, B and its intersection C is mapped with the contravariant functor $\mathcal{F}^{op}$ to $\mathcal{F}(A), \mathcal{F}(B)$ and $\mathcal{F}(C)$ respectively such that inclusion maps in X are mapped with the contravariant functor $\mathcal{F}^{op}$ to restriction maps. This figure is adapted from [Bod23]. 54

3.1 Exclusion criteria applied to 409 patients to obtain an image dataset from 205 patients. 67

4.1 a. Loss landscape visualizations demonstrating the flatter landscapes in the vision transformer compared to the ResNet. b. The visualisations are in keeping with the lower maximum Hessian eigenvalues in the vision transformer. This figure is taken from [PK21]. 73

4.2 AlterNet architecture. The blue, white, and grey blocks indicate subsampling, conv, and MHSA block respectively. The network is a modification of the pre-activation ResNet-50 network. This figure is taken from [PK21]. 74

4.3 Second stage of training. We train a cross-attention architecture with input sampled for the T2 and ADC dictionary. 79

4.4 Bar Chart showing the AUC performance ± acquisition shift for 6 model types. 89

4.5 Bar Chart comparing AUC performance of 3 domain generalisation applied 3 different models compared to our model without any domain generalisation method applied. 91

4.6 Bar Chart showing the AUC performance under various perturbations for 5 model types. 93

5.1 Sampled image input and Segmentation output for 2 domain shifts in chest X-ray (top row) and prostate (bottom row). 103

5.2 Variance heatmap of UNet(left) and VQ-UNet(right) latent space under 4 Gaussian noise levels for the NIH dataset. The X-axis indicates a unique subset of features from a latent space, the Y-axis corresponds to 100 randomly sampled test set images, and the value at each location indicates the variance of a specific feature for a given image across 100 test time augmentations with Gaussian noise. 104

5.3 Sampled abdomen CT (Top row), chest X-ray (Middle Row) and prostate MRI (Bottom row) input images and Segmentation outputs for the UNet and VQ-UNet under 0% (1^{st} 3 columns) and 30% (2^{nd} 3 columns) s&p noise. 105

6.1 Our method (HypC) learns base dictionaries embedded in hyperbolic space for each class. The embedding features, z are split into m (number of classes) groups with each group sampling its respective class hyperbolic dictionary, $\mathbb{D}^i$ to form zq^i which is composed to form class segmentation outputs, c^i. 110

6.2 The figure shows the segmentations of an abdominal CT slice with 20% Gaussian noise addition (bottom row) and Gaussian blur (top row) produced by the UNet and TransUNet before and after incorporating our method. 113

6.3 Segmentation of a prostate slice from the BMC dataset when training on the RUNMC dataset with different SDG methods and our method. . . . 117

7.1 Overview of our method demonstrating using the ADC map to learn to shape equivariant features which are quantised to construct a shape dictionary, $\mathbb{D}$. We assume the dictionary is complete and sufficient to capture the entire distribution of segmentation outputs after the composition of the discrete shape space using the decoder. 128

7.2 Cayley Diagram for D4(Right) and D4h(Left). This image is taken from [WC22]. 130

7.3 The average Dice and maximum Hausdorff distance scores ± standard deviation for the respective dihedral group equivariant CNNs trained via contrastive learning and applied to the prostate dataset. The scores are averaged across all classes. 139

7.4 The average Dice and maximum Hausdorff distance scores ± standard deviation for the respective dihedral group equivariant CNNs trained via contrastive learning and applied to the cardiac dataset. The scores are averaged across all classes. 140

7.5 This figure shows an example prostate segmentation output comparing the D8h and D12h equivariant segmentation model trained using contrastive learning with the 5 other SDG methods. 142

7.6 This figure shows an example cardiac segmentation output comparing the D8h and D12h equivariant segmentation model trained using contrastive learning with the 5 other SDG methods 142

7.7 The average Dice and maximum Hausdorff distance score ± standard deviation for the respective group equivariant CNNs trained via kernel constraints and applied to the prostate dataset. The density of the equivariant constraints imposed on the kernel increases from C4 to SO(2) (left to right). The scores are averaged across all classes. 143

7.8 The average Dice and maximum Hausdorff distance score ± standard deviation for the respective group equivariant CNNS trained via kernel constraints and applied to the cardiac dataset. The density of the equivariant constraints imposed on the kernel increases from C4 to SO(2) (left to right). The scores are averaged across all classes. 144

7.9 This figure shows an example prostate segmentation output comparing the C16 and D16 shape equivariant segmentation model trained via the kernel constraint method with 5 other SDG methods. 145

7.10 This figure shows an example cardiac segmentation output comparing the C16 and D16 shape equivariant segmentation model trained via the kernel constraint method with 5 other SDG methods. 145

8.1 Overview of our Topology Preserving Compositionality (TPC) method incorporated in a segmentation model with p levels (scales). $\mathcal{D}$ signifies a persistence diagram derived from a cubical complex C. For example, the persistence diagram for the foreground label segmentation at level p(top right) is expressed as $\mathcal{D}_p^{FL}$ derived from $C(FL^p)$. Background segmentations omitted. 151

8.2 Toy example demonstrating the basic principle of TPC. 153

8.3 In this figure, we show the segmentations of an abdominal CT slice with 20% Gaussian noise addition (top row) and Gaussian blur (bottom row) by the UNet, nn-UNet and Swin-UNetr and after including TPC in each model. 157

8.4 Segmentation of a cardiac slice in the top row and prostate slice in the bottom row when testing on a target domain with different SDG methods and our method (TPC). 159

8.5 Segmentation of a cardiac slice in the top row and prostate slice in the bottom row when testing on a target domain before and after incorporating TPC into 3 segmentation architectures. 160

8.6 The mean Dice score (left) and Betti error (right) after each component of TPC is sequentially added (left to right) into the Baseline Model to assess SDG segmentation for the cardiac and prostate datasets. 'Cls Topology' refers to incorporating the topological loss over each class segmentation. 'Hier Topology' is adding the second topological loss for the foreground segmentation. 162

9.1 Example of 3 images with the same persistence diagrams and Betti numbers. 165

9.2 Cellular sheaf constructed from the poset P visualised through a commutative diagram. 167

9.3 a. 2 patches (Y_1, Y_2) glued together in an ordered manner defined by the poset, P, via inclusions maps to form the prostate segmentation output $Y = Y_1 \cup Y_2$. b. cellular sheaf $\mathcal{F}$ over the poset P. 168

9.4 a.Original shape b,c,d. Edges between patches defined by the cellular sheaf over a poset defined in our method on a b. 2×2, c. 4×4, d. 8×8 grid. 174

9.5 In this figure, we show the segmentation outputs from the baseline model trained with the Dice loss (Baseline), Dice loss plus TopoLoss ('TopoLoss') and Dice loss plus our cellular sheaf-based loss (SheafLoss). We apply our sheaf-based loss to $2 \times 1, 2 \times 2, 4 \times 4, 8 \times 8$ discretisations of the segmentation. The segmentation outputs of a cardiac slice are shown in the top row and a prostate slice in the bottom row when testing on an unseen target domain. 179

9.6 In this figure, we show the segmentation outputs comparing the D8h shape equivariant model with the *D8h* shape equivariant model trained with our cellular sheaf loss. We also show the segmentation output of the baseline model trained with various SDG techniques. The segmentation outputs of a cardiac slice are shown in the top row and a prostate slice in the bottom row when testing on an unseen target domain. 182

9.7 Tumour and zonal segmentation of a prostate slice from the RUNMC dataset when training on the internal dataset. 184

10.1 Example knowledge tree. Here, symbols ζ_1^0 and ζ_2^0 are being abstracted to form ζ_1^1 which forms class0. Symbols ζ_2^0 and ζ_3^0 are being abstracted to form ζ_2^1 which forms class1. 190

10.2 Overview of the proposed framework, in which the feature extractor and classifier describe the trained blocks of the given model. The Euclidean codebook forms a discrete representation of the continuous latent space from the feature extractor, followed by hyperbolic codebooks and reasoning blocks to obtain a knowledge tree, which is later used in extracting explanations. The decoder block is independently trained to obtain visual semantics for the extracted hierarchical symbolic rules. . . 192

10.3 Explanations obtained using the proposed framework for an MNIST classifier. (a) Shows the 4 post-hoc XAI methods we compared with. (b) Demonstrates the obtained class-level tree, which is a complete set of FOL chain rules responsible for 'class 2', (c) indicates an image-level tree formed from the sampled chain rules responsible for making a decision for a given image, and (d) demonstrates a subset of the visual $PartOf$ relations obtained from an image-level tree. 200

10.4 Explanations obtained for an AFHQ classifier deciding upon the 'class cat'. The subset of the visual $PartOf$ relation is shown. 200

10.5 Explanations obtained using the proposed method for an STL10 classifier deciding upon 'class 9 (Boat)'. 201

10.6 This figure describes the hierarchical visual explanations obtained using our proposed framework for the MIMIC classifier classifying class 'abnormal(class 0)', with atoms corresponding to $pf(class0, \zeta_0^2), pf(class0, \zeta_1^2)$ described in the first row. The second row visualises the obtained atoms corresponding to $pf(\zeta_0^2, \zeta_{37}^1)$. The last row visualises the atoms corresponding to symbols in ζ^0 which are part of ζ_{37}^1. 202

10.7 Robustness comparison of considered methods with our proposed method. In our method, we show the visual semantic of a single symbol sampled for the last codebook; ζ_0^2. The top shows the heatmaps with no noise added. The second row shows heatmaps with 30% Gaussian noise added. 203

10.8 Robustness experiments with 30% Gaussian noise on a sample from the MNIST dataset. The symbol for ζ_3^2 is shown. The image level tree did not change under 30% Gaussian noise for this example. 203

10.9 Robustness experiments with 30% salt and pepper noise on a sample from the MIMIC dataset. The symbol for ζ_0^2 is shown. The image level tree did not change under 30% S&P noise for this example. 204

10.10 Robustness experiments with 20% Poisson noise on a sample from the STL10 dataset. The symbol for ζ_0^2 is shown. The image level tree did not change under 20% Poisson noise for this example. 204

10.11 2D Poincaré embedding of symbols obtained for MNIST. Red, blue, and green nodes indicate symbols from $\zeta^0, \zeta^1, \zeta^2$ respectively. 205

Chapter 1

Introduction

1.1 Motivation

In medical imaging, hand crafted extracted features such as textural and statistical measures have been the predominant features used for various downstream tasks ranging from survival prediction to classification. However, Convolutional neural networks (CNNs) have revolutionised various computer vision tasks. We are now as result seeing a shift to learning features through CNNs in medical imaging. Transformer networks have revolutionised natural language processing through using self-attention to better model long range dependencies to extract more global information [VSP+17]. Transformer networks have now been extended to being used in image classification in what is called the vision transformer [DBK+20].

Unfortunately, the robustness of deep learning models have been called into question, which makes the safe translation of these models into clinical practise an issue. Model robustness is generally defined as some distribution shift from the training to test data which leads to performance degradation in some metric. In general, one can divide distribution shifts as follows:

- **Population Shifts:** This describes a shift in the population characteristics including sex, age, pathology and other variables which affect the structural appearances of objects in an image.

- **Textural Shifts:** This describes a shift in the acquisition of images which affect the textural appearances of an image and includes contrast, noise profile and resolution.

Adversarial robustness is well studied in the robustness literature, where perturbations are explicitly optimized to deceive the model. Adversarial machine learning offers well defined quantifiable notions of robustness to image perturbations. However, this relies upon explicitly stated constraints which often fail to align with real-world scenarios. For example, L_p-norm bounds $|\eta|_p \leq \epsilon$ are commonly used to constrain adversarial perturbations, yet typical corruptions in real-world scenarios, such as salt-and-pepper noise, can easily exceed these bounds [DSSU21].

In this thesis, as opposed to adversarial examples which are explicitly constructed to degrade model performance, we investigate more natural occurring perturbations which arise in the natural environment or acquisition process. Non-adversarial robustness in medical imaging occurs in the form of naturally induced corruption such as motion and noise artefacts as well as acquisition shifts leading to textural and contrast changes. We find that there is relatively less focus in the literature in regards to tackling this form of non-adversarial robustness and forms the motivation of this thesis. We will look at both synthetic corruptions such as Gaussian noise as well as acquisition shifts which occurs in the task of single domain generalisation.

The literature thus far centres around tackling model robustness with data augmentation strategies, optimisation techniques and architectural design choices. In this thesis, we tackle model robustness by analysing the geometry and topology of the latent space. We specifically impose a discrete structure on the latent space and embed the components in this space with certain geometric and topological constraints. Our intuition in this proposal lies in the way humans reasons about the visual world through discrete or categorical representations [TT20] which ignores noise.

The notion of imposing compositionality into the latent space for building robust representations as well as improving model interpretability has recently been explored in the literature [TWH19, YCLX23, LWU+20]. It is therefore quite established that structure in the latent space enables more robust representation. However, we also hypothesise

the geometry and topology in which to embed these discrete representations to be vital for model robustness. For example, the curvature of the latent space should play a role in embedding the correct relational structure between finite components in the latent space such that the composition of these components are natural. An example would be to use hyperbolic geometry to build discrete hierarchical structures in the latent space which we explore in this thesis for segmentation and symbolic reasoning. It is also natural that certain symmetries and structures in the input space should be preserved in the mapping to the latent space to enable more robust representations. This is explored in this thesis through the notion of preserving topology and equivariance in a discrete latent space for robust segmentation.

1.2 Research Objectives

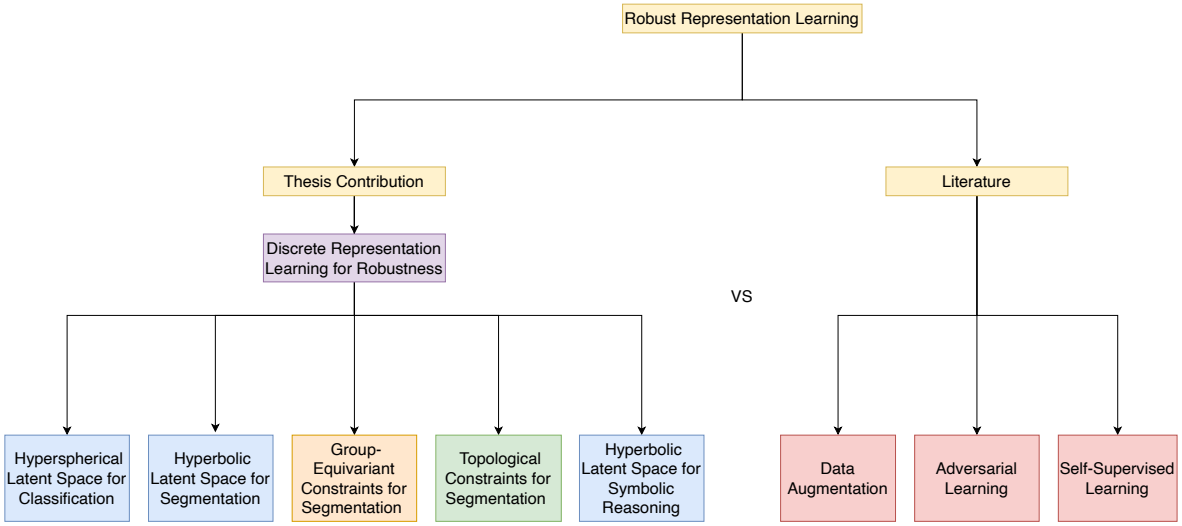


Figure 1.1: This figure provides a thesis overview. We start with the initial problem of robust representation learning and tackle this with discrete representation learning (purple box) with curvature based constraints (blue boxes), group equivariant based constraints (orange box) and topological constraints (green box). We compare our approach with the standard methods in the literature to improve model robustness (red boxes).

We provide an overview of our thesis in the diagram in figure 1.1. Our overall goal is to build robust representations with a focus on non-adversarial robustness. We tackle this problem with building discrete representations with geometric and topological constraints which breaks down this thesis into the following 5 research objectives.

1. The first objective of our thesis is to formalise and demonstrate discrete representation learning improves model robustness for disease classification (chapter 4) and segmentation (chapter 5). This is highlighted in the purple box in figure 1.1.

2. The second objective is to highlight the importance of curvature of the latent space to building robust representations in downstream tasks. We impose hyperspherical constraints on a discrete representation for robust disease classification (chapter 4). We also aim to embed a hierarchical discrete structure on a hyperbolic manifold to increase robustness in segmentation (chapter 6) and generating hierarchical explanation for deep discriminative models (chapter 10). This objective is highlighted in the blue boxes in figure 1.1.

3. The next objective is to impose a group equivariant discrete shape latent space to improve the robustness of segmentation models (chapter 7). We also aim to explore equivariance to various groups on model robustness. This is highlighted in the orange box in figure 1.1.

4. Our final objective is to explore topological constraints using persistent homology (chapter 8) and cellular sheaves (chapter 9) for robust segmentation. This is highlighted in the blue box in figure 1.1.

5. We justify our approach by comparing to a myriad of methods in the literature for building robust representations including data augmentation, adversarial learning and self-supervised learning. This is highlighted in the red boxes in figure 1.1.

1.3 Thesis Outline and Publications

The thesis outline with a brief description of each chapter is shown below. The start of each chapter recalls the publication(s) whose content is reproduced or adapted therein.

1.3.1 Chapter 2

This chapter provide the required background for this thesis. We first give an introduction to CNNs and the vision transformer followed by discussing robust representation learning. We then provide a short primer on Non-Euclidean geometry (hyperbolic geometry), group theory, algebraic topology (persistent homology), category theory, sheaf theory and measure theory.

This chapter is adapted from the background sections of the following publications.

- **Santhirasekaram, A.**, Pinto, K., Winkler, M., Aboagye, E., Glocker, B. and Rockall, A., 2021. Multi-scale Hybrid Transformer Networks: Application to Prostate Disease Classification. In Multimodal Learning for Clinical Decision Support: 11th International Workshop, ML-CDS 2021, Held in Conjunction with MICCAI 2021, Strasbourg, France, October 1, 2021, Proceedings 11 (pp. 12-21). Springer International Publishing.

- **Santhirasekaram, A.**, Winkler M, Rockall A, Glocker B. Robust prostate disease classification using transformers with discrete representations. International Journal of Computer Assisted Radiology and Surgery. 2024 May 13:1-0.

- **Santhirasekaram, A.**, Winkler, M., Rockall, A.G. and Glocker, B., 2024. A Geometric Approach to Robust Medical Image Segmentation. *Under review*.

- **Santhirasekaram, A.**, Winkler, M., Rockall, A. and Glocker, B., 2023. Topology Preserving Compositionality for Robust Medical Image Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 543-552).

- **Santhirasekaram, A.**, Pinto, K., Winkler, M., Rockall, A. and Glocker, B., 2023, October. A Sheaf Theoretic Perspective for Robust Prostate Segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 249-259). Cham: Springer Nature Switzerland.

- **Santhirasekaram, A.**, Kori, A., Winkler, M., Rockall, A., Toni, F., Glocker, B., 2023. Robust Hierarchical Symbolic Explanations in Hyperbolic Space for Image

Classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 561-570).

1.3.2 Chapter 3

In this chapter, we provide a description of the internal and external datasets used in this thesis.

1.3.3 Chapter 4

This chapter is adapted from the following two publications.

- **Santhirasekaram, A.**, Winkler M, Rockall A, Glocker B. Robust prostate disease classification using transformers with discrete representations. International Journal of Computer Assisted Radiology and Surgery. 2024 May 13:1-0.
- **Santhirasekaram, A.**, Kori, A., Winkler, M., Rockall, A. and Glocker, B., 2022, September. Vector Quantisation for Robust Segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 663-672). Cham: Springer Nature Switzerland.

This chapter is focused on the robustness of the vision transformer for prostate disease classification. We begin by realising that the vision transformer and CNN in low data regimes are not robust to domain shifts which is often the case in medical imaging. We improve robustness by using discrete representations of the image using a VQ-VAE for input to the vision transformer. We then show model robustness increases further by embedding the discrete representation on a hypersphere. We analyse model robustness in prostate MRI disease classification from a 1.5T to a 3T scanner and vice versa.

1.3.4 Chapter 5

This chapter is adapted from the following publication.

- **Santhirasekaram, A.**, Kori, A., Winkler, M., Rockall, A. and Glocker, B., 2022, September. Vector Quantisation for Robust Segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 663-672). Cham: Springer Nature Switzerland.

In this chapter, we suggest and substantiate the idea that enhancing robustness of a segmentation model is possible by learning a low dimensional discrete representation of the image through a dictionary learning method called vector quantisation which is then used to decode the segmentation output. This is based on the assumption that there is limited anatomical variability among patients.

1.3.5 Chapter 6

This chapter is adapted from the following publication.

- **Santhirasekaram, A.**, Winkler, M., Rockall, A. and Glocker, B., 2023, October. Hierarchical Compositionality in Hyperbolic Space for Robust Medical Image Segmentation. In MICCAI Workshop on Domain Adaptation and Representation Transfer (pp. 52-62). Cham: Springer Nature Switzerland.

In this chapter, we impose a hierarchical structural prior on our discrete latent space and consider the natural curvature to embed our hierarchy for segmentation tasks. We therefore embed the fundamental components on a hyperbolic manifold, asserting that this choice facilitates a more natural composition which leads to improved model robustness.

1.3.6 Chapter 7

This chapter is adapted from the following two publications.

- **Santhirasekaram, A.**, Winkler, M., Rockall, A.G. and Glocker, B., 2023, April. Shape Equivariant Learning for Robust MRI Segmentation. In Medical Imaging with Deep Learning, short paper track.

- **Santhirasekaram, A.**, Winkler, M., Rockall, A.G. and Glocker, B., 2024. A Geometric Approach to Robust Medical Image Segmentation. *Under review*.

In this chapter, we posit that by constraining the latent space to a finite set of group equivariant shape components, we can enhance the robustness of segmentation models to textural shifts, under the assumption that the segmentation output exclusively contains shape information.

1.3.7 Chapter 8

This chapter is adapted from the following publication.

- **Santhirasekaram, A.**, Winkler, M., Rockall, A. and Glocker, B., 2023. Topology Preserving Compositionality for Robust Medical Image Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 543-552).

In this chapter, we again constrain the latent space to a finite set of shape-only components. However, here we sample the shape dictionary in topology-preserving manner using persistent homology which improves the robustness of the segmentation model. We further enhance robustness by deep topological supervision applied in an hierarchical manner.

1.3.8 Chapter 9

This chapter is adapted from the following publication.

- **Santhirasekaram, A.**, Pinto, K., Winkler, M., Rockall, A. and Glocker, B., 2023, October. A Sheaf Theoretic Perspective for Robust Prostate Segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 249-259). Cham: Springer Nature Switzerland.

In this chapter, we realise persistent homology is a very gross descriptor of the shape information contained in the segmentation output as it is location invariant and does not provide important geometric properties such as area and curvature. We tackle this drawback, by drawing on the perspective of cellular sheaf theory to model compositionality driven by global as well as local topological correctness.

1.3.9 Chapter 10

This chapter is adapted from the following publication.

- **Santhirasekaram, A.**, Kori, A., Winkler, M., Rockall, A., Toni, F., Glocker, B., 2023. Robust Hierarchical Symbolic Explanations in Hyperbolic Space for Image Classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 561-570).

In this chapter, our emphasis is on developing explanations with multiple levels of abstraction for black-box models, particularly deep discriminative models for images. In the literature, the post-hoc explanations are often sensitive to perturbations in the input space. Here, we leverage the inherent properties of hyperbolic geometry to efficiently model hierarchical relationships of symbolic features while minimising distortion. The result is robust hierarchical explanations formulated as chain rules, guided by the principles of inductive logical programming (ILP).

1.3.10 Chapter 11

In this chapter, we conclude our work with a discussion of our initial research objectives followed by a summary of our thesis achievements and future work.

Chapter 2

Background Theory

This chapter provides the required background theory to understand the content of this thesis. We will first delve into areas of deep learning such as CNNs and discrete representation learning before introducing the notion of robustness in deep learning from a causal perspective. We lastly will provide the required mathematical details in algebraic topology, Riemannian geometry, category theory and measure theory in order to understand the contributions in this thesis which are largely centered on model robustness from a topological and geometric perspective.

This chapter is adapted from the background sections of the following publications.

- **Santhirasekaram, A.**, Pinto, K., Winkler, M., Aboagye, E., Glocker, B. and Rockall, A., 2021. Multi-scale Hybrid Transformer Networks: Application to Prostate Disease Classification. In Multimodal Learning for Clinical Decision Support: 11th International Workshop, ML-CDS 2021, Held in Conjunction with MICCAI 2021, Strasbourg, France, October 1, 2021, Proceedings 11 (pp. 12-21). Springer International Publishing.
- **Santhirasekaram, A.**, Winkler M, Rockall A, Glocker B. Robust prostate disease classification using transformers with discrete representations. International Journal of Computer Assisted Radiology and Surgery. 2024 May 13:1-0.
- **Santhirasekaram, A.**, Winkler, M., Rockall, A.G. and Glocker, B., 2024. A Geometric Approach to Robust Medical Image Segmentation. *Under Review*.

- **Santhirasekaram, A.**, Winkler, M., Rockall, A. and Glocker, B., 2023. Topology Preserving Compositionality for Robust Medical Image Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2023 (pp. 543-552).

- **Santhirasekaram, A.**, Pinto, K., Winkler, M., Rockall, A. and Glocker, B., 2023, October. A Sheaf Theoretic Perspective for Robust Prostate Segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 249-259). Cham: Springer Nature Switzerland.

- **Santhirasekaram, A.**, Kori, A., Winkler, M., Rockall, A., Toni, F., Glocker, B., 2023. Robust Hierarchical Symbolic Explanations in Hyperbolic Space for Image Classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 561-570).

2.1 Convolutional Neural Networks

This thesis largely focuses on the robustness of CNNs for classification and segmentation. Therefore here we provide a background into the fundamentals of the CNN and its different components as well as the application of CNNs to classification and segmentation.

CNNs have dominated the field of computer vision in recent years [LB+95]. CNNs are now being used in image classification, object detection, segmentation, image rendering, image reconstruction and many more computer vision applications. CNNs are a type of regularised feed-forward network which extracts a set of features from an image via optimising a set of kernels [LB+95]. There are 4 main components which make up a CNN backbone, the convolutional layer, activation function, normalisation layer and pooling layer.

2.1.1 Convolutional Layer

The centrepiece of the CNN is the convolutional layer. The convolutional layer consists of a set of kernels each of which is translated across the entire image. The translation process normally involves placing the kernel in the top left pixel of the image and computing the Hadamard product between the kernel weights and the input volume plus bias offsets. The input volumes is the same as the kernel size. The images are often padded with 0 or 1 for the input volume to match the kernel size at the edges and corners of the image. This is repeated for every pixel in the grid. One would normally apply multiple different kernels over the image. This means that given a 64×64 grid, if a set of 8, 3×3 kernels are applied across the entire image, then one only has to optimise 72 weights plus a bias term for each kernel. This local inductive bias creates a much sparser connection between layers in a CNN compared to a fully connected layer. Weight sharing also means far fewer weights are learnt compared to a fully connected layer. Furthermore, this process is translation invariant which is a useful property as the position of an object does not affect the CNN performance. Hyper-parameters to choose when passing the kernel over the image are the stride length, kernel size and padding which determine the output dimensionality. We formalise in equation 2.1 the applications of a $K \times K$ kernel with weights w to the activation at position $A_{i,j}$ in layer, $l-1$ of the CNN to produce $X^l_{i,j}$. b_l is the bias term for layer l and σ is the activation function which is applied.

$$X^l_{i,j} = \sigma\left(\sum_{m=0}^{K-1}\sum_{n=0}^{K-1} w^l_{m,n} A^{l-1}_{i+m,j+n} + b^l\right) \tag{2.1}$$

2.1.2 Activation Function

The activation function maps the input to a co-domain where computation over this co-domain is desirable. It also incorporates non-linearities to increase the expressivity of the function space and prevent the CNN collapsing to a simple linear function. The most common activation function is the rectified linear unit (ReLU) activation function

shown in Equation 2.2 [NH10].

$$\text{ReLU}(x) = \max(0, x) \tag{2.2}$$

The ReLU activation removes all negative values and sets them to 0 which can lead to the 'dying' ReLU problem where neurons can become inactive. Therefore, activation functions which dampen the negative input have been introduced such as the leaky ReLU and Swish activation shown in equation 2.3 [RZL17]. β is a hyper-parameter in equation 2.3 which controls the non-linear smooth interpolation between a linear and ReLU function.

$$\text{Swish}(x) = x \cdot \text{Sigmoid}(\beta x) \tag{2.3}$$

The GELU (Gaussian Error Linear Unit) activation function is a type of activation function used in neural networks, particularly popular in transformer-based models [HG16]. It is designed to combine the non-linearity of activation functions with smooth gradients. The GELU activation function can be defined with the following equation:

$$\text{GELU}(x) = x \cdot \Phi(x) = \frac{x}{2} \cdot \left(1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right) \tag{2.4}$$

where: - $\Phi(x)$ is the cumulative distribution function of the standard normal distribution. - $\text{erf}(x)$ is the Gaussian error function.

This equation indicates that the GELU activation function applies a non-linear transformation to its input based on the Gaussian error function, allowing it to maintain continuity and smooth gradients, which can be advantageous in deep learning models.

2.1.3 Normalisation Layer

Normalisation layers are typically applied between the convolution layer and activation function to standardise the statistics of the neural layers. Normalisation layers are essential to stabilizing the training process by reducing internal covariate shift as well as acting as a form of regularisation and preventing overfitting. Additionally, it also allows gradients to flow smoothly during backpropagation which prevents exploding or vanishing gradients. There are many types of normalisation layers which include

batch normalisation [SLJ+15], instance normalisation [UVL16] and layer normalisation [BKH16].

2.1.4 Pooling Layer

Pooling layers are used to down-sample the feature maps and extract the most important information in each layer. This is most commonly performed using max pooling which divides the feature map into a set of non-overlapping partitions and extracts the maximum value from each partition. This makes pooling layers locally translation invariant.

2.1.5 Classification

Classification is the most popular task employed by CNNs. The CNNs for classification traditionally consist of multiple consecutive convolutional layers with interspersed pooling layers which down-sample features before being flattened as input to a fully connected layer. The output of the fully connected layer is then passed through a softmax function to produce a vector of probabilities which sum to one. Each value in this vector corresponds to the posterior probability of a particular class given an input.

There have been a variety of CNN architectures which achieved state-of-the-art (SOTA) performance over the years. For example, Alex-Net; a CNN with 8 layers achieved a top-5 error of 15.3% which was 10.8% lower than the runner-up [KSH12]. Thereafter, there were numerous architectural design choices which allowed to the building of larger CNNs such as the use of residual connections and depth-wise convolutions; the ResNet-152 which consists of 152 layers now achieves a top-5 error of 3.57 % [HZRS16a].

2.1.6 Segmentation

Segmentation in image processing and computer vision involves partitioning an image into distinct and semantically coherent segments [DZ00]. The primary objective is to identify and delineate meaningful regions within an image, resulting in a labelled image

where each pixel is associated with a specific class. Various types of segmentation tasks exist, including: (i) semantic segmentation for pixel-wise categorization; (ii) instance segmentation for distinguishing individual instances of the same class and (iii) pan-optic segmentation that combines semantic and instance segmentation for a comprehensive scene understanding [DZ00]. This process is fundamental in computer vision and finds applications in diverse fields such as medical imaging, autonomous vehicles and satellite image analysis by providing the spatial structure of images.

In this thesis we focus on semantic segmentation and the UNet is the common semantic segmentation architecture used [RFB15]. The UNet derives its name from its U-shaped architecture. It consists of an encoder to map the input to a low dimensional bottleneck before being up-sampled by a decoder to map to the segmentation output. Skip connections enhance the model's ability to recover spatial information by providing detailed information from the contracting path directly to the corresponding layers in the expansive path. This helps mitigate the vanishing gradient problem and allows the network to localise objects more accurately. In figure 2.1 we provide an example of a 3D UNet architecture which maps a $160 \times 160 \times 64$ image to 256 features in the bottleneck before being mapped to the segmentation output by the decoder.

2.2 Vision Transformers

In chapter 4, we delve into the robustness of the vision transformer for disease classification. Therefore, we will now give an overview of the vision transformer and the multi-headed self attention (MHSA) mechanism which underpins its workings.

The transformer architecture [VSP+17] was first proposed for natural language processing (NLP) tasks, primarily focusing on machine translation. The key idea behind transformers is the self-attention mechanism, which models weighted dependencies between different parts of the input sequence. Transformers can model dependencies within a sequence in a parallel manner making it well equipped to model long-range interactions. This is in contrast to recursive neural networks (RNN) or long short-term memory networks (LSTMs) which process input sequences sequentially and therefore struggle to model significantly distant relationships due to gradient vanishing

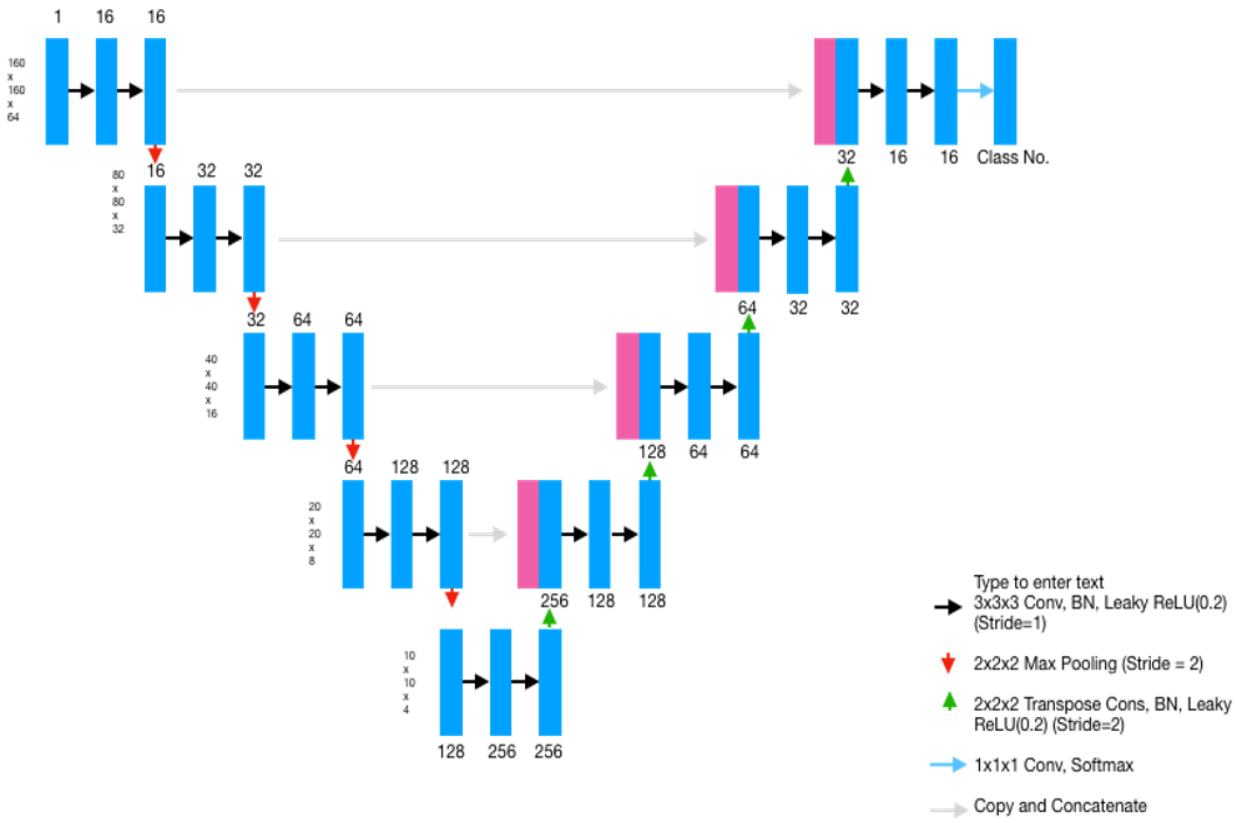


Figure 2.1: Example UNet architecture.

[RHW86, GSC00, GSK+16]. This breakthrough innovation in NLP laid the foundation for the development of the vision transformer.

Computer vision tasks such as image classification, segmentation and object detection have been dominated by CNNs. Recently, the vision transformer has changed the computer vision landscape and has now become a new tool for tackling computer vision problems. An overview of the vision transformer is shown in figure 2.2a. In this figure, the input image is segmented into distinct, non-overlapping patches of a fixed size, such as 16x16 pixels. A learnable linear projection is then applied to transform these patches into flattened vectors. To encode the spatial information of each patch, positional embeddings are incorporated into the patch embeddings as shown in figure 2.2a.

In figure 2.2b, we show the architecture for the vision transformer. The vision transformer employs the conventional transformer architecture, which is composed of a series of identical layers, each containing two sub-layers: Multi-head Self-Attention (MHSA) and Multi-Layer Perceptron (MLP). MHSA enables the model to assess the

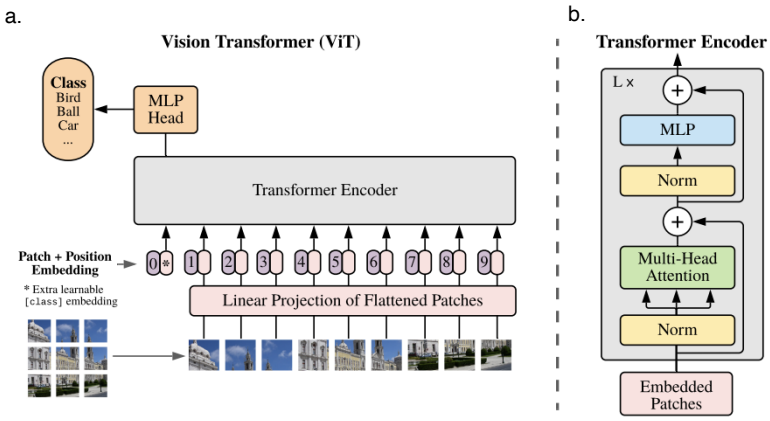


Figure 2.2: Overview of the vision transformer architecture. This figure is taken from [DBK+20].

significance of various patches in relation to each other. The MLP is made up of two fully-connected layers with a GELU activation function sandwiched between them. These layers are connected through residual connections and layer normalisation to strengthen the flow of information.

For image classification tasks, a trainable classification token is incorporated into the input patch embeddings. This token progresses through the transformer layers and is subsequently utilised to predict the input image's class. The final layer is a linear layer followed by a softmax function to obtain class probabilities.

2.2.1 Positional Encoding

The attention mechanism is invariant to positional information of the patches. In order to incorporate this vital spatial information, positional encodings are added to the patches. Specifically, a d dimensional vector is added to each d dimensional flattened patch. In equation 2.5, we show absolute positional encodings for the j^{th} patch. Here, positional encoding corresponds to a sine wave function, which enables the transformer model to acquire the ability to focus on relative positions within the input sequence. One can use relative [SUV18] and learnt [GAG+17] positional encodings which are used in various other transformer-based models [DCLT18]. However, this is not used in our

work.

$$p_{i,j} = \begin{cases} \sin\left(\frac{i}{1000^{\frac{j}{d}}}\right), & \text{if j is even.} \\ \cos\left(\frac{i}{1000^{\frac{j-1}{d}}}\right), & \text{if j is odd.} \end{cases} \tag{2.5}$$

2.2.2 Multi-Headed Self Attention

The first stage of the MHSA mechanism is to convert each input vector into a d dimensional query q, key k and value v vector with a linear layer, which is then concatenated respectively over all input vectors to form Q, K and V matrices. The general equation for self-attention is shown in equation 2.6. Attention scores between different input vectors are calculated using the dot product between Q and V. The attention matrix A is normalised in order to enhance gradient stability for improved training. The softmax is applied to translate A into probabilities W before multiplying with the value matrix V to form the output O. If the vectors in V are flattened patches in the image, then the output is a weighted linear combination of the input patches. Hence, patches with larger weights/ attention scores have a larger influence on the output.

$$\text{Output} = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) \times V \tag{2.6}$$

In multi-headed self-attention, one boosts the performance of single-head self-attention by applying multiple attention heads with different learnt Q, K and V matrices for each head. This means each attention head can project the input vector into different representation subspaces leading to learning more diverse representations. The equation for MHSA is formally defined in equation 2.7 where the representations learnt for each head are concatenated before applying a projection layer with weights, $W_0 \in \mathcal{R}^{d \times d}$.

$$MultiHead(Q, K, V) = \underset{h \in [N_h]}{Concat}[Head_h] \times W_{out} \tag{2.7}$$

The learned embeddings then go through a series of alternating multi-head self-attention and multi-layer perceptron layers. The vision transformer benefits greatly from being able to attend to every patch of the image in every layer which is not possible in CNNs. It is hypothesised that the vision transformer can learn better global features. The vision transformer has achieved SOTA performance outperform-

ing CNNs on several image classification benchmarks such as ImageNet and CIFAR-10 [DBK+20].

2.3 Training Deep Learning Models

We introduce the underlying mechanisms of learning in deep learning to provide background to the optimisers and loss functions used in this thesis

The goal of learning is to update the parameters corresponding to weights and biases in a deep learning model to minimise a loss function. The gradient of the weights with respect to the loss function is computed using backpropagation [RHW86]. The gradients are used to update the weights to minimise the error with an optimiser such as stochastic gradient descent [RM51] and adaptive moment estimation (Adam) [KB14].

In classification and segmentation, the minimisation problem can be formulated as: discover the optimal parameters $\hat{\theta}$ to minimise the expected loss over the joint distribution $P(X, Y)$ which is formalised below.

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \, \mathbb{E}_{(x,y)\sim P(X,Y)} \left[L(y, f(x; \theta)) \right] \tag{2.8}$$

In equation 2.8, $x \in X$ is the input image and $y \in Y$ is the corresponding label. In practise, the true distribution $P(X, Y)$ is not known, so it is approximated by minimising the empirical loss over a given training dataset, $\mathcal{D}_{tr}$. This is shown in equation 2.9 and is known as empirical risk minimisation.

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \, \mathbb{E}_{(x,y)\sim \mathcal{D}_{tr}} \left[L(y, f(x; \theta)) \right] \tag{2.9}$$

The loss function typically used for global or pixel-wise classification is the cross-entropy loss function shown below.

$$H(y, p) = -\sum_{i} y_i \log p_i \tag{2.10}$$

In equation 2.10, y is the true probability distribution (one-hot encoded) and p is the

predicted probability distribution which is the output of a softmax function.

One can also use the Dice loss function shown in equation 2.11 for segmentation tasks. In equation 2.11, y_i is the binary ground truth for pixel i and p_i is the predicted probability for pixel i.

$$\text{Dice Loss} = 1 - \frac{2\sum_i (y_i \cdot p_i)}{\sum_i y_i^2 + \sum_i p_i^2} \tag{2.11}$$

The Hausdorff distance is another metric used to compare the similarity of the segmentation output to the ground truth. It is a measure of the distance between two sets of point. The maximum Hausdorff distance H is the maximum distance between any two points in two respective sets. This is formally defined in equation 2.12 where O is the set of points in the output segmentation mask and G is the set of points in the ground truth segmentation mask. $d(o, g)$ is the distance between $o \in O$ and $g \in G$ respectively.

$$H(O, G) = \max_{o \in O} \left(\min_{g \in G} d(o, g) \right) \tag{2.12}$$

2.4 Discrete Representation Learning

Discrete representation learning forms the basis for this entire thesis for building robust representations before imposing geometric and topological constraints as shown in figure 1.1. We therefore will give a detail overview of discrete representation learning.

Discrete representation learning aims to learn a low dimensional representation of data which is a categorical distribution. In deep learning, this means that the latent space can only be represented by a finite set of d dimensional vectors. Discrete representations for visual data were used prior to deep learning such as the bag of visual words model [CDF+04, SZ03]. The seminal paper by [VDOV+17] proposed using vector quantisation for learning a discrete latent space. The proposed architecture is called the vector quantised variational auto-encoder (VQ-VAE). They achieved competitive performance with their continuous counterparts in terms of the log-likelihood and circumvented the issues of a large variance and posterior collapse seen in the VAE. They also managed to generate high-quality samples in speech and video generation when pairing the VQ-VAE with a powerful auto-regressive prior. An extension of the

VQ-VAE is the VQ-GAN which uses adversarial training to generate more realistic images [ERO21]. In this work, a patch-wise discriminative loss function is applied to encourage local as well as globally realistic-looking images [ERO21].

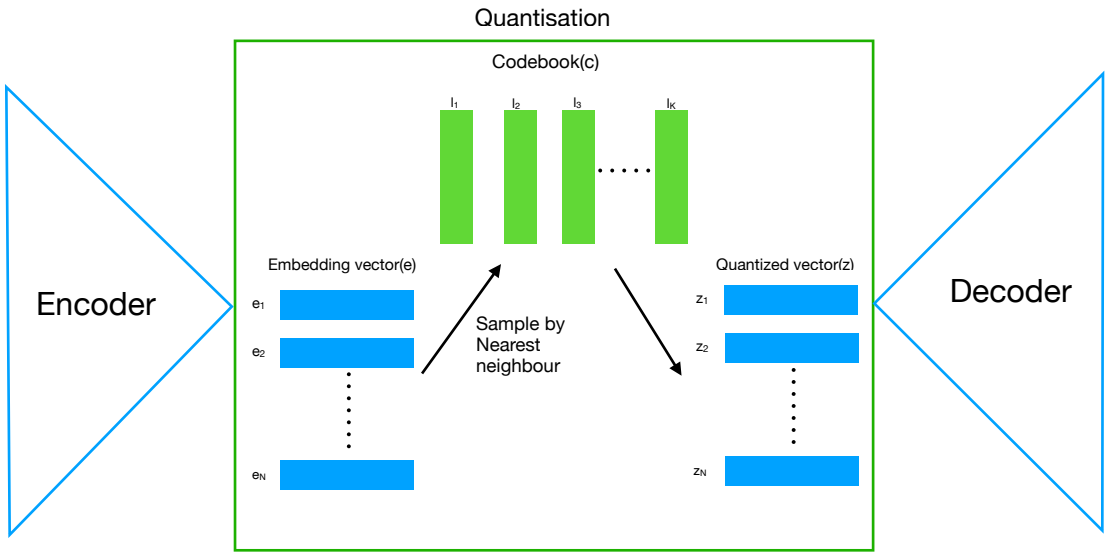


Figure 2.3: Overview of the VQ-VAE architecture.

The VQ-VAE architecture is shown in figure 2.3. The VQ-VAE firstly consists of an encoder ϕ_e which maps the inputs space x to the continuous latent vectors e which are then discretised using vector quantisation to form discrete variables $\hat{z}$. The decoder ϕ_d, maps the discrete latent vectors to the output space y. The quantisation of the continuous space is performed by first dividing e into m vectors. We spatially divide e, the continuous latent space of size, $c{\times}x{\times}y{\times}z$ into vectors e_i of size $c{\times}1{\times}1{\times}1$ where c is the number of channels. We then replace $\forall e_i \in e$ with the nearest element, $l_k \in \mathbb{D}$ sampled by Euclidean distance to form the discrete latent variables $\hat{z}$ where $k = \arg\min_j \|e_i - l_j\|_2$. $\hat{z}$ is then reshaped back into the original feature maps' dimension of size $\mathbb{R}^{c\times x\times y\times z}$. Note, the approximate posterior categorical distribution denoted $q(\hat{z}_i = l_j \mid x)$ is deterministic and one hot encoded as shown in equation 2.13. We assume a uniform prior over $\hat{z}_i$. This means one can derive a constant KL-divergence equal to $\log K$. Therefore the KL-divergence term is ignored during training.

$$q(\hat{z}_i = l_j \mid x) = \left\{ \begin{array}{ll} 1 & \text{for } l_k \in \mathbb{D}, \text{where } k = \arg\min_j \|e_i - l_j\|_2 \\ 0 & \text{otherwise} \end{array} \right\} \tag{2.13}$$

The log-likelihood of the model is shown in equation 2.14.

$$\log p(x) = \log \sum_{i=0}^{m} p(x \mid \hat{z}_i) p(\hat{z}_i) \tag{2.14}$$

We cannot backpropagate through this sampling operation to update the codebook and therefore approximate the gradients for $\mathbb{D}$ using straight-through gradient approximation [BLC13]. This is achieved by copying the gradients from the decoder input z to the encoder output e. Given that the encoder's output representation and the decoder's input inhabit the same d dimensional space, the gradients provide valuable insight on how to adjust the encoder's output to minimize the reconstruction loss. Therefore $\mathbb{D}$ does not receive any gradients. To learn the codebook, we move l_k closer to e_i by Euclidean distance. This is captured in the second term of the loss function shown in equation 2.15 where a stop gradient (sg) is applied to e_i which sets the gradient attached to e_i to 0 and constrains e_i to a non-updated constant. The volume of the continuous embedding space can grow arbitrarily large during training and therefore a commitment loss is applied shown in the third term in equation 2.15. This term updates the encoder parameters in order to commit to an embedding by bringing e_i closer to l_k. The first term in equation 2.15 is the reconstruction loss term computed using the mean square error which optimises the encoder and decoder parameters. The codebook elements are initialised uniformly from $-1/K$ to $1/K$.

$$\mathcal{L}_{Euc} := \log p(x \mid \hat{z}) + \sum_{i=0}^{m} \|sg(e_i) - l_k\|_2 + \sum_{i=0}^{m} \beta \|e_i - sg(l_k)\|_2 \tag{2.15}$$

2.5 Deep Learning in Medical Imaging

2.5.1 Medical Imaging

We focus on applying the methods developed in this thesis to medical imaging. Hence, it is important to provide a brief background into the current application of deep learning methods to medical imaging in the literature.

Over the past decade, deep learning in medical imaging has garnered significant re-

search attention, with its application poised to expand further in clinical practice. The intersection of deep learning, computer vision, and artificial intelligence (AI) has led to a surge in industrial AI applications and holds substantial potential for the future of medical imaging [SWS17]. The proliferation of breakthroughs in AI is fueled by enhanced access to extensive 'big data', facilitated by advancements in computing power. In clinical settings, where vast amounts of valuable data are generated, deep learning emerges as a pivotal tool, adept at integrating, analysing, and predicting outcomes from large and diverse datasets.

2.5.2 Magnetic Resonance Imaging

In this thesis, Magnetic Resonance Imaging (MRI) data is used throughout and it is hence important to give a background into MRI. MRI data has a complex acquisition process which makes it prone to artefacts and textural/contrast shifts from one scanner type to another. These textural shifts arise because acquisition parameters for the myriad of MRI sequences often subtly vary from one vendor to the next. Furthermore, the absence of a standardised intensity scale in MRI poses a significant challenge for deep learning inputs. Conventional MRI intensities are typically acquired in arbitrary units, resulting in textural shifts. This lack of uniformity complicates the application of deep learning algorithms in MRI analysis. Furthermore, MRI images are often acquired with anisotropic resolution which often means 3D images have to be processed as 2D slices for input into a deep learning model.

MRI also often suffer from bias field artefacts which is a low-frequency variation signal that corrupts the image, inducing a gradual change in intensities. This effect is caused by inhomogeneities in the magnetic field [EHN+04]. Artifacts in MRI can also arise from other factors such as from patient-related voluntary and physiological motion or metallic implants/foreign bodies which causes an inhomogenous magnetic field [EHN+04]. Sampling-based artefacts can result in aliasing and Gibbs artefacts [EHN+04]. Pulse sequence characteristics may contribute to artefacts like black boundary, Moiré, and phase-encoding artefacts [EHN+04]. MRI hardware-related issues can also lead to radio-frequency overflow artefacts [EHN+04].

2.5.3 Prostate

Prostate segmentation and disease classification is the major clinical application which runs throughout this thesis. Prostate disease classification on multi-parametric MRI (mpMRI) is performed by radiologists using a standardised reporting lexicon called PI-RADS. This has greatly improved the diagnostic accuracy of clinically significant prostate cancer [BAAA+18]. However, it is still somewhat subjective and results in a false-positive rate of 30-40 percent leading to a large number of unnecessary biopsies [BPC+20]. It would be greatly beneficial to develop an automated classification method to improve diagnostic accuracy.

In the medical domain including prostate disease classification on MRI, we are now seeing a shift from extracting hand-crafted features (radiomics) such as textural and statistical measures to using CNNs to learn the features. The ProstateX challenge is a popular benchmark used for classifying lesions on prostate MRI and CNNs have become the most popular method yielding state-of-the-art (SOTA) performance [ALDP20, CTAN+20, IMU+18, SZY+18, WLC+18, YLW+17].

Early CNN approaches for prostate segmentation focused on the entire gland, employing either 2D slice segmentation or 3D volume segmentation techniques. [MNA16] introduced the use of sub-volumes/patches for segmentation, employing volumetric ConvNets with mixed residual connections. [ZDY19] addressed the challenge of distinguishing the prostate boundary by incorporating a boundary-weighted transfer learning approach to modify the U-net architecture. [JSH+19] further enhanced whole gland segmentation on the PROMISE12 dataset by emphasising boundary learning, achieved through the combination of a 3D segmentation decoder and a 2D boundary decoder.

Currently, CNNs are widely employed for lesion detection, leveraging their effective hierarchical feature extraction capabilities. [KBS+17] integrated a U-net with an adversarial network, achieving a Dice score of 0.41 across a dataset of 152 patients. In contrast, [FDC+19] utilised a mask R-CNN and achieved an AUC of 0.46 with a dataset of 63 patients, underscoring the considerable challenge in tumour segmentation on prostate MRI. Notably, [CZS+19]. demonstrated promising results in prostate MRI tu-

mour segmentation, achieving a Dice score of 0.57. Their approach involved a 3D CNN implementing a focal loss function to address high class imbalance.

2.6 Robust Representation Learning

The goal of this thesis is to tackle the problem of improving the robustness of deep learning models. We therefore give an overview of robust representation learning in the literature and provide a causal perspective on robustness of models used for image related tasks.

Deep learning has made significant progress in various computer vision tasks such as image classification. However, deep learning models which include CNNs are susceptible to both adversarial and natural image manipulations. Research showed imperceptible image perturbations also known as adversarial attacks could alter model predictions drastically [SZS+13]. In the real-world setting, CNNs are also vulnerable to non-adversarial corruptions such as noise and artefacts which can reduce accuracy by 30 to 40 % [HD18]. This raises concerns about the reliability of these models in high-stakes systems such as healthcare. Consequently, researchers have developed various approaches to evaluate and enhance model robustness.

2.6.1 Robustness framework

We introduce a formal framework for non-adversarial robustness through a causal lens as proposed by [DSSU21]. In this approach, the data generating process (DGP) for an image is represented as a structural causal model (SCM) [Pea09, PJS17] which is visualised with a directed acyclic graph (DAG). In figure 2.4, we show the DAG for the DGP of an image. Here, the task T related to an image such as segmentation or classification decomposes the regions of interest in the image from population factors E_p such as pathology, sex and age which contribute to the content in the image and from the environmental factors E_d which affect capture of the image such as motion and temperature. The content of the image is acquired with sensor S such as the detection of radio-frequency signal in MRI which can be affected by physiological motion (E_d).

The signal acquired is then used to render R or reconstruct the image, $X \in \mathbb{R}^d$ which together with the image-related task leads to the label Y. In figure 2.4, each variable is expressed as a function of its parents and an exogenous noise term, $\mathcal{E}$ (not included in figure 2.4). The value of each variable can therefore be determined by structural equations representing the causal relationships between variables in the DAG. This DAG yields the following structural equations demonstrated in [DSSU21]:

$$\alpha : T \times \mathcal{E}_d \to \mathbb{R}^{E_p}, \quad \beta : T \times E_d \times \mathcal{E}_C \to \mathbb{R}^C, \quad \Gamma : C \times E_d \times \mathcal{E}_S \to \mathbb{R}^S, \tag{2.16}$$

$$\psi : S \times \mathcal{E}_R \to \mathbb{R}^R, \quad \rho : R \times \mathcal{E}_X \to \mathbb{R}^X, \quad \lambda : T \times X \times \mathcal{E}_Y \to \mathcal{Y}. \tag{2.17}$$

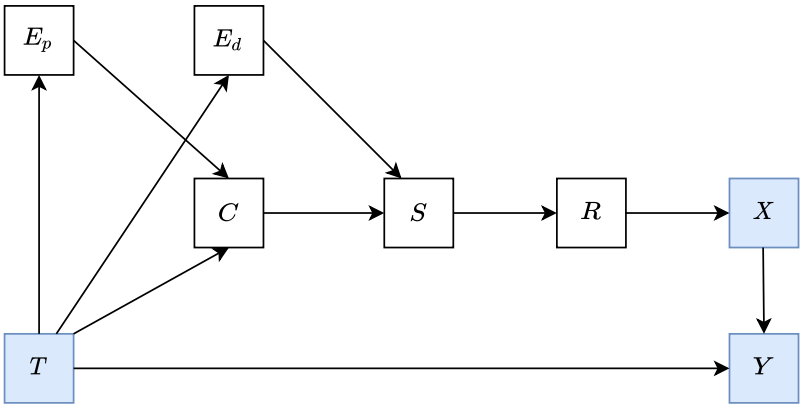


Figure 2.4: The above DAG demonstrates that task T involves separating the classes of interest C from population factors E_p and environmental factors E_d. Scenes, formed by sampling from these variables, are passed through a sensor S, before being rendered R to form the acquisition process which produces the final image X. Labels Y are conditioned on both the task and the resulting image. This figure is adapted from [DSSU21].

The joint distribution over all the variables in figure 2.4 decomposed into conditionals which satisfies the SCM in figure 2.4. A key difference between SCMs and statistical graphical models is the ability to perform counterfactual reasoning. Here, we can set variables to certain values called interventions and observe the effects on causally subsequent variables. Note that the prior causal variables are not affected. One can also perform soft interventions where one performs a shift in the distribution of a variable.

Hence, a logical strategy for assessing model robustness involves examining the model's performance on data derived from counterfactual images containing artefacts derived

through soft or hard interventions. In this thesis, we ignore interventions on E_p and focus on interventions on E_d, S and R. We look at robustness to artefacts and contrast/textural distributional changes. Artefacts in an image correspond to samples from the underlying joint distribution, $P(T, E_p, E_d, C, S, R, X, Y)$ where the values of E_d, S and R are of low probability. For example, a motion artefact corresponds to the tails of the marginal, $P(E_d)$. Therefore, a model is robust to artefacts if Y is unaltered given a hard intervention on E_d, S or R such that the probability of the value E_d, S or R lies in the tails of marginals, $P(E_d)$, $P(S)$ or $P(R)$ which corresponds to events which are rarely encountered during training.

2.6.2 Domain Generalisation

Domain generalisation aims to learn robust features that are invariant across multiple domains for downstream tasks, without requiring access to the target domain data during training. To assess robustness to distributional changes, soft interventions are required through counterfactually altered versions of figure 2.4. This requires a new realisation of the underlying joint distribution in the SCM. In our case, this arises from a distributional shift in the marginals, $P(E_a)$, $P(S)$ or $P(R)$ through soft intervention. Therefore robustness to a distributional shift can be assessed by the stability of Y under a soft intervention of E_a, S or R. For example, in MRI a textural shift which arises from a $1.5T$ to $3T$ scanner is equivalent to applying a soft intervention on S leading to distribution shift in the marginal $P(S)$.

In our work, we require a measure of robustness. Robustness is relative, therefore, we provide a measure of robustness compared to a control model. We define the error rate of the control and model of interest, with corruption $c \in C$ as $E^c_{control}$ and E^c_{model} respectively. We define the absence of the corruption as *clean* and the severity of the corruption as s which range from 1 to n levels of severity. The relative corruption error (rCE) is shown in equation 2.18 [DSSU21].

$$rCE(\%) := \sum_{c \in C} \sum_{s=1}^{n} \frac{E^c_{model,s} - E^{clean}_{model}}{E^c_{control,s} - E^{clean}_{control}} \times 100 \tag{2.18}$$

2.6.3 Robustness Methods

The most common techniques used to tackle model robustness are centred around data augmentation, adversarial training and architecture design. In data augmentation, we enhance model robustness by synthesising new training data from the existing training set to encourage model invariance to various perturbations. It was shown that model accuracy is not maintained under different types of noise and distortions [GTR+18]. Therefore, augmenting training data with specific corruptions helps preserve accuracy for those corruptions, but not for unseen ones [GTR+18]. In medical imaging, BigAug is an aggressive augmentation scheme which generates training data with a series of augmentation techniques which significantly improve segmentation performance in the SDG setting [ZWY+20]. It was highlighted by [GTR+18] that neural networks overly rely on texture information. This is in contrast to humans who prefer shape for classification. Therefore, style transfer was used by [GTR+18] to augment ImageNet with textured variants. They achieved improved accuracy on unseen common perturbations in ImageNet-C and provided evidence that this increases the shape bias of CNNs. Similarly, RandConv [XLY+20] proposes using randomised convolutions in the initial layers of CNNs to extract shape-biased features. Recently, there have been augmentation strategies which force neural networks to learn even more generalisable features such as CutOut [DT17] and MixUp [ZCDLP17]. In CutOut [DT17], random patches of an image are cropped out which encourages more global feature learning for classification. MixUp [ZCDLP17] linearly combines randomly sampled training images and labels which leads to more stable predictions on data sampled outside the training distribution such as those degraded by artefacts. Puzzlemix [KCS20] adopts a more sophisticated approach using optimal transport between pairs of original images to optimise the degree of information retained from each image which best preserves the salient features in the combined output.

Adversarial training schemes akin to data augmentation methods also manipulate the input data but with a specific objective. In methods such as [MMS+18], a min-max problem is constructed in which the inner maximisation seeks effective perturbations from a distribution such as Gaussian noise, while the outer minimisation updates the model parameters to minimise expected error. In an extension of this method, [RSZ+20]

learns the noise distribution from which to sample the perturbations. There have since been more sophisticated adversarial training schemes developed. For example, M-ADA [QZP20] introduces adversarial data augmentation through a meta-learning approach, utilising a Wasserstein autoencoder to create new domains. In medical imaging, AdvBias [CQQ+20] develops an adversarial training scheme specific to MRI data. Here, they employ adversarial data augmentation in the input space to learn the generation of bias field deformations.

A popular architectural design choice to improve the robustness of CNNs is centred around spectral methods which remove high-frequency information. For example, [Zha19] introduces a general-purpose computational layer by enforcing anti-aliasing in the max/average pooling and strided convolutional operations. Similarly, [LSGL20] replaces up/down sampling operations in the CNN with a wavelet-based method to improve robustness to noise. [LWL+20] aims to minimise feature redundancies across space and channels by using the discrete cosine transform to learn a series of sparse transforms which they denote as LST. This can be easily integrated into common CNN architectures and enhanced accuracy was demonstrated across all ImageNet-C corruption types after LST is incorporated into the ResNet-50.

2.7 Group Theory and Equivariance

Throughout this thesis, group theoretic terminology is prescribed, particularly in chapter 7. Therefore, group theory prerequisites are given to provide a basic understanding. This is not an exhaustive coverage of group theory, but rather the fundamental concepts to understand the content in this thesis.

Group. A group consists of a set, G of elements and a binary operator $\cdot\colon G \times G \to G$, termed the group product. This operator dictates how to combine pairs of elements $g_1, g_2 \in G$. To qualify as a group product, the binary operator must adhere to four conditions:

1. Closure: G is closed under $\cdot$: $\forall g_1, g_2 \in G$, $g_1 \cdot g_2 \in G$.

2. Identity: There exists an identity element e such that, $\forall g \in G, e \cdot g = g \cdot e = g$.

3. Inverse: For every element, $g \in G$, there is an element $g^{-1} \in G$, s.t. $g \cdot g^{-1} = e$.

4. Associativity: For any set of elements, $g_1, g_2, g_3 \in G$, $(g_1 \cdot g_2) \cdot g_3 = g_1 \cdot (g_2 \cdot g_3)$.

Semi-direct product. A semi-direct product is a generalisation of the direct product which describes a group as the product of more than one subgroup. In this work we view the construction of a group G as the inner semi direct product $\rtimes$, of subgroups $\mathcal{H}$ and $\mathcal{N}$ ($G = \mathcal{N} \rtimes \mathcal{H}$). This intrinsic view enforces $\mathcal{N}$ or $\mathcal{H}$ to be a normal subgroup. Given $\mathcal{N}$ is the normal subgroup, the following statements are then equivalent.

- $\mathcal{N}\mathcal{H} = G$ and $\mathcal{N} \cap \mathcal{H} = \{e\}$.
- Every $g \in G$ can be written uniquely as $g = nh$ with $n \in \mathcal{N}, h \in \mathcal{H}$.
- Define $\psi : \mathcal{H} \to G/\mathcal{N}$ in the natural way: $\psi(h) = \overline{h} = h\mathcal{N}$. Then ψ is an isomorphism.

Group Action. A group G can interact with a space made up of a set $\mathcal{X}$. Given a group element $g \in G$, the group operation T_g outlines the effect on any element $x \in X$ when the transformation specified by element g, $T_g(x)$ is applied to it. The action is defined formally as follows:

$$T : G \times X \to X \quad \text{and} \quad T_g : X \to X \tag{2.19}$$

Group representation. In group theory, a representation ρ is defined as a function that maps each element in the group to an invertible $n \times n$ matrix. Here, n refers to the dimension of the representation, which can be any positive integer, including infinity. A representation ρ of a group G must fulfill the equation $\rho(gg_0) = \rho(g)\rho(g_0)$, where gg_0 represents the composition of two group elements $g, g_0 \in G$, and $\rho(g)\rho(g_0)$ symbolises matrix multiplication. This condition ensures that the matrix multiplication in the representation corresponds to the operation within the group, thereby preserving the group's structure.

Left group representation. Any group action on a set $\mathcal{X}$ implicitly alters functions on $\mathcal{X}$ because it changes the elements of $\mathcal{X}$. This change is typically articulated via a left-regular representations $\mathcal{L}$. Therefore given a function, $f : \mathcal{X} \to \mathbb{R}$, we can determine

the nature of a transformed function f' after applying a group element g, by assessing a function's value for $x \in \mathcal{X}$ by reversing the transformed function. For instance, to find the value of f' for a transformed element a', we determine the original value of f for a before g was applied. This is achieved by applying the inverse of g to a'. For the set of functions f on $\mathcal{X}$, the left regular representation for g is shown below.

$$L_g : f \rightarrow f' \text{ and for } a' \in T_g(X) : f'(a') = f(T_{g^{-1}}(a')) \tag{2.20}$$

Equivariance A function is equivariant with respect to a group if it commutes with the action of the group. Therefore a function $\Phi : X \rightarrow Y$ is equivariant if the following holds true:

$$\forall g \in G : T_g \circ \Phi = \Phi \circ T_g. \tag{2.21}$$

We will first describe two finite groups which we use in this thesis.

Cyclic Group. A cyclic group is a group which is generated by a single element X which we describe as the group generator. Cyclic groups are also Abelian which is a group in which the law of composition is commutative. The order of the group is n and the identity $X^n = I$ holds in the cyclic group where I is the identity. An example of the cyclic group is the ring of integers $\mathbb{Z}$ under addition which is an infinite group.

Dihedral Group. For any integer $n \geq 3$, the dihedral group D_n represents the set of symmetry operations for a regular n-gon, with the group operation being the composition of these actions. The group actions consist of rotations and reflections. The group order is 2n. We can formally represent the group as: $\langle x, y | x^2 = 1, y^n = 1, (xy)^2 = 1 \rangle$.

We will now formally describe three compact different types of groups which are necessary for this thesis. Firstly, we describe the translation group in two dimensions $R2$ with matrix representation, ρ. This group has a group product and inverse for two elements $x, x_0 \in R2$ shown below.

$$\rho(x) \cdot \rho(x_0) = (x + x_0) \tag{2.22}$$

$$\rho^{-1}(x) = -x \tag{2.23}$$

The logarithmic map is given by:

$$\log \rho(x) = x. \tag{2.24}$$

We also consider the rotation group in two dimensions $SO(2)$ which consists of the set of all continuous rotation transformations around a plane. These rotation matrices are orthogonal matrices R with determinant 1. Its group product and inverse for two elements $R_\theta, R_{0_\theta} \in SO(2)$ is given by:

$$R_\theta R_{\theta_0} = R_{\theta+\theta_0} \tag{2.25}$$

$$R_\theta^{-1} = R_\theta^T. \tag{2.26}$$

The final group we consider in our work is the special Euclidean group $SE(2)$ in two dimensions. The 2-dimensional Special Euclidean group represents a series of geometric transformations comprised of rotations and translations in a 2D plane. Every element of the group can be identified using two parameters: the rotation angle (θ) and the translation vector (t), denoted as $g = (t, \theta)$ within G. The group product and inverse for any two elements g, g_θ in SE(2) are defined as follows:

$$g \cdot g_0 = (t, \theta) \cdot (t_0, \theta_0) = (T_\theta(t_0) + t, \theta + \theta_0) \tag{2.27}$$

$$g^{-1} = (-T_{-\theta}(t), -\theta) \tag{2.28}$$

In essence, to amalgamate the two elements g and g_0, we first employ the rotation part of g to the translation of g_0, followed by adding it to the translation of g. This is a semi-direct combination of elements. As such, SE(2) is a semi-direct product, made up of the translation group $R2$ and the rotation group $SO(2)$, often notated as $SE(2) = R2 \circ SO(2)$. The above groups are specific to the 2D plane but easily extendable to 3D space.

2.8 Topology and Geometry

This thesis is centred on improving the robustness of deep learning models by imposing topological and geometric constraints in the latent space. Therefore, we will provide a

brief primer on certain topics in algebraic topology and Riemmanian geometry required to understand the tools which are used in this thesis.

2.8.1 Introduction

There is a series of increasing number of properties which connect a topological space and Riemannian manifold. We start with a set and equip this set with the notion of a well-defined neighbourhood structure around each point to construct a *topological space*. Next, if a locally defined neighbourhood around each point 'looks' Euclidean, then we refer to this as a *topological manifold*. We can then add a further constraint on this manifold such that it is differentiable everywhere making this now a *differentiable manifold*. A differentiable manifold however can be embedded in Euclidean or Non-Euclidean space. Finally, if we define the inner product between tangent vectors on each point of a differentiable manifold, we construct a *Riemmanian* manifold which allows us to measure distance, area, angles and curvature on a Euclidean or Non-Euclidean manifold. A summary of the hierarchy connecting the topological space and the Riemannian manifold is shown in figure 2.5.

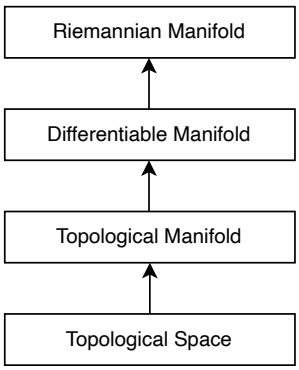


Figure 2.5: A hierarchy demonstrating the increasing constraints imposed which connect a topological space with a Riemannian manifold. This figure is adapted from [Bod23].

In this work, we will focus on the properties of a topological space and Riemannian manifold as required for this thesis.

2.8.2 Topological Space

A topological space is a set X with subsets T which satisfy the following axioms:

Definition 1. *A topological space is a set X with subsets T which satisfy the following properties:*

- *The empty set $\emptyset$ and X are in T.*
- *Any finite intersection of sets in T are also in T.*
- *Any finite union union of sets in T are also in T.*

Given, a point a in the topological space X, all subsets Y of X which contain a, $a \subseteq Y$, is the *neighbourhood* of a. We next provide examples of topological spaces generated by the set $A = \{1, 2, 3\}$. The discrete topological space is all subsets of A which include, $\{\emptyset, 1, 2, 3, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$. The trivial topological space is the empty set and full set; $\{\emptyset, \{1, 2, 3\}\}$. An example of a set which is not a topological space is the subsets, $\{\emptyset, \{1, 2\}, \{1, 3\}, \{1, 2, 3\}\}$. This is because the intersection of $\{1, 2\}$ and $\{1, 3\}$ is 1 which is not in the set. An important topological space is Euclidean space generated by all open balls in $\mathbb{R}^n$. Euclidean space is equipped with a metric structure and all metric spaces are topological spaces. However, topological spaces do not require a metric structure.

We can now describe the useful notion of a topological basis for a topological space.

Definition 2. *The basis of a topological space X is the collection of open subsets $\mathcal{B}$ in X if any subset of X can be expressed as some finite union of elements in $\mathcal{B}$.*

The topological basis is a compact way of expressing topological spaces. For example, the base topology for the discrete topological space would consist of the empty set and all singletons in the discrete topological space.

We will next define the open cover of a topological space.

Definition 3. *An open cover, C of X is a collection of open subsets U_i whose union is X formalised in the below equation.*

$$C = \cup_{i \in I} U_i \tag{2.29}$$

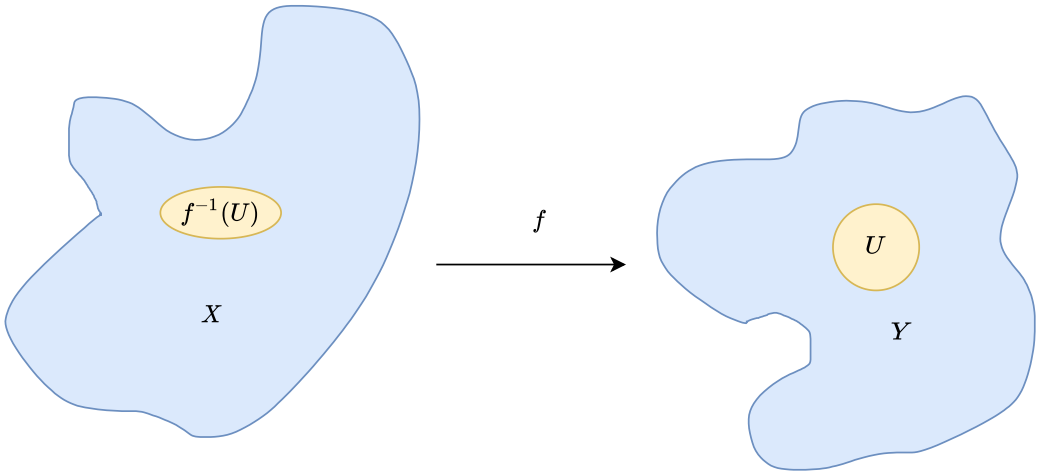


Figure 2.6: A continuous maps between topological spaces X and Y where the the pre-image of $U \subseteq Y$, is in X. This figure is adapted from [Bod23].

In this thesis, we also use the notion of continuous maps and homeomorphisms between topological spaces.

Definition 4. *A map f between topological spaces X and Y, is continuous if the pre-image f^{-1} of every open set in Y is an open set in X.*

We provide an example of a continuous map between topological spaces in figure 2.6 inspired by [Bod23].

Definition 5. *A map h between topological spaces X and Y, is homeomorphic if h is bijective and if h as well as h^{-1} is continuous.*

A useful notion of a homeomorphic map is a map which maps the natural numbers $n \in \mathbb{N}$ to $\mathbb{R}^n$. This defines an Euclidean space and if this map is not homeomorphic then $\mathbb{R}^n$ is Non-Euclidean.

Further information about topological spaces such as homotopy and homotopy equivalence which is not relevant to this thesis can be found in [ELZ00].

2.8.3 Complexes

One can capture the connectivity of a topological space through a cell complex which allows one to capture the homology of a topological space. The homology groups allow

us to define a finite descriptor of the topological space. We start with the simplicial complex; an example of a cell complex which is a triangulation of a topological space.

Informally, one can describe a simplex as an n^{th} dimensional generalisation of a triangle. Formally, a k-simplex is a k-dimensional polytope which is the convex hull of its $k+1$ affinely independent vertices. For example, the 0-simplex is a point, the 1-simplex is a line, the 2-simplex is a triangle and the 3-simplex is a tetrahedra. We visualise examples of k-simplices from the 0-simplex to the 3-simplex in figure 2.7.

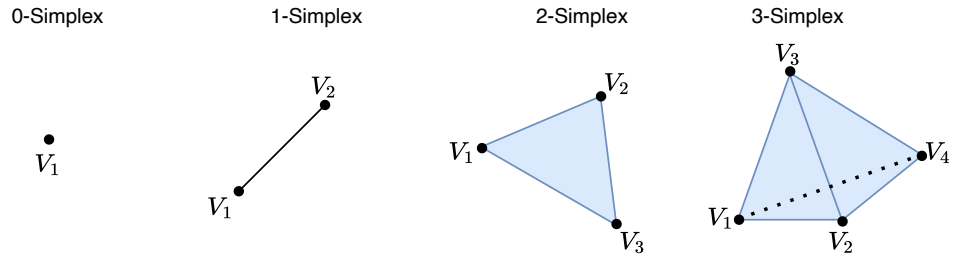


Figure 2.7: We show an example of the k-simplex from $k = 0$ which is a point to $k = 3$ which is the tetrahedra.

We will now define a simplicial complex

Definition 6. *A simplicial complex, $\mathcal{K}$ is a set of k-simplices which satisfy the following axioms:*

- *The face of any k-simplex in $\mathcal{K}$ is also in $\mathcal{K}$*
- *The intersection of any two k-simplices in $\mathcal{K}$ is a face of both simplices.*

Given a simplicial complex $\mathcal{K}$, we will now define the k^{th} dimensional simplicial chain C_k over $\mathcal{K}$.

Definition 7. *A simplicial k-chain is a linear combination of the k-simplices, $\sigma \in \mathcal{K}$ formally defined in the equation below.*

$$C_k(\mathcal{K}) = \left\{ \sum_i z_i \sigma_i \mid z_i \in \mathbb{Z} \right\} \tag{2.30}$$

Note, if $z_i \in \mathbb{R}$, then we have a vector space of k-chains. This is however not relevant for this thesis. We will provide a concrete example to demonstrate the simplicial chains C_k in a simplicial complex, $\mathcal{K}$.

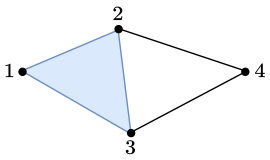


Figure 2.8: An example of a simplicial complex $\mathcal{K}$.

In figure 2.8, we show a simplicial complex, $\mathcal{K}$ and derive the simplicial chains C_k from $k = 0$ to $k = 2$.

$$\begin{aligned} C_0(\mathcal{K}) = \{&0, 1, 2, 3, 4, 1+2, 1+3, 1+4, 2+3, 2+4, 3+4, 1+2+3, 1+2+4, \\ &1+3+4, 2+3+4, 1+2+3+4\} \end{aligned} \tag{2.31}$$

$$\begin{aligned} C_1(\mathcal{K}) = \{&0, 12, 31, 23, 42, 34, 12+31, 12+23, 12+42, 12+34, 31+23, 31+42, 31+34, \\ &23+42, 23+34, 42+34, 12+31+23, 12+31+42, 12+31+34, 12+23+42, \\ &12+23+34, 31+23+42, 31+23+34, 23+42+34, 12+31+23+42, \\ &12+23+42+34, 12+31+42+34, 12+31+42+34, 12+31+23+34, \\ &31+23+42+34, 12+31+23+42+34\} \end{aligned} \tag{2.32}$$

$$C_2(\mathcal{K}) = \{0, 123\} \tag{2.33}$$

The next step to defining the homology group is to define the boundary homeomorphisms.

Definition 8. *Given a simplex, $\sigma \in \mathcal{K}, \sigma = [v_0, v_1, ...v_k]$, the boundary homeomorphism, $\partial_k : C_k(\mathcal{K}) \to C_{k-1}(\mathcal{K})$ is defined in the equation below.*

$$\partial_k(\sigma) = \sum_i (-1)^i [v_0 ... \hat{v}_i ... v_k] \tag{2.34}$$

$\hat{v}_i$ in equation 2.34 is the removal of v_i from the sequence.

We next will provide the following important theorem about boundary operators.

Theorem 1. *The boundary of a boundary is always 0.*

$$\partial_{k-1}\partial_k = 0 \tag{2.35}$$

We will now prove theorem 1.

Proof.

$$\partial_{k-1}\partial_k[v_0, v_1, ...v_k] = \partial_{k-1}\sum_i (-1)^i [v_0...\hat{v}_i...v_k] \tag{2.36}$$

$$\begin{aligned} &= \sum_{j<i} (-1)^i (-1)^j [v_0...\hat{v}_j...\hat{v}_i...v_k] \\ &+ \sum_{j>i} (-1)^i (-1)^{j-1} [v_0...\hat{v}_i...\hat{v}_j...v_k] \end{aligned} \tag{2.37}$$

$$= 0 \tag{2.38}$$

Equation 2.37 is equal to 0 because the second sum negates the first sum. □

The boundary homeomorphism therefore allows to connect the simplicial chains in a simplicial complex to construct a chain complex C as shown below.

$$... \to C_{k+1} \xrightarrow{\partial_{k+1}} C_k \xrightarrow{\partial_k} C_{k-1} \to ... \tag{2.39}$$

We will now define the k^{th} boundary group B_k and cycle group Z_k.

Definition 9. *The k^{th} boundary group is the image of the boundary homeomorphism ∂_{k+1} expressed below.*

$$B_k = im\ \partial_{k+1} \tag{2.40}$$

Definition 10. *The k^{th} cycle group is the kernel of the boundary homeomorphism ∂_k expressed below.*

$$Z_k = ker\ \partial_k \tag{2.41}$$

$$= \{c \in C_K | \partial_k c = \emptyset\} \tag{2.42}$$

An example of an element in C_1 which lies in the kernel of ∂_1 is $\{12 + 23 + 31\}$ because $\partial_1(12+23+31) = \partial_1(12)+\partial_1(23)+\partial_1(13) = (1-2)+(2-3)+(3-1) = 0$. Given, definition 9 and 10, theorem 1, allows us to realise that the image of ∂_{k+1} is a subspace of the kernel of ∂_k. This means the boundary of C_{k+1} lies in the cycle group Z_k and we therefore make the following implication in equation 2.43. This implication is also visualised in figure 2.9

$$B_k \subseteq Z_k \subseteq C_k \tag{2.43}$$

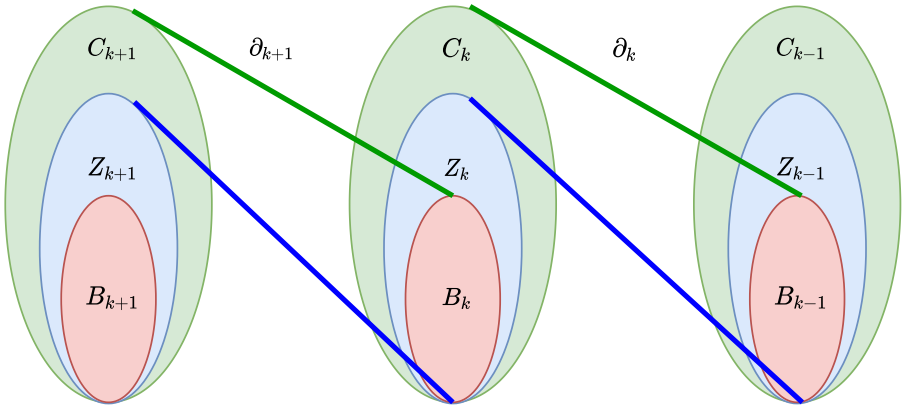


Figure 2.9: Visualisation of a chain complex with its boundary and cycle groups. The image (green line) and kernel (blue line) of the boundary operators are demonstrated.

.

We now have the ingredients to define the homology group. Intuitively, The homology groups H_n measures n-dimensional "holes" or cycles in the space. These groups provide a systematic algebraic framework for understanding and classifying the topological features associated with these holes, helping to distinguish between different shapes and structures in the space. H_0 notes the connected components in the topological space. H_1 defines the holes/tunnels in the space while H_2 captures the voids.

We define the homology group in the following equation.

$$H_k = Z_k/B_k = \ker \partial_k / \text{img } \partial_{k+1} \tag{2.44}$$

Equation 2.44 states the homology groups H_k are cycles *modulo* boundaries which means it's the group which are cycles but not boundaries in C_k.

As an exercise, we can calculate the homology group H_1 of figure 2.8. We already know

H_2 is empty because the image of ∂_3 is not defined. To calculate H_1, we have to take the image of ∂_2 and kernel of ∂_1 and then take the quotient between these two groups. We show the calculation below.

$$\text{img } \partial_2(C_2) = \{12 + 23 + 31\} \tag{2.45}$$

$$\ker \partial_1(C_1) = \{12 + 23 + 31, 23 + 42 + 34, 12 + 24 + 43 + 31\} \tag{2.46}$$

$$\ker \partial_1(C_1)/\text{img } \partial_2(C_2) = \{23 + 42 + 34, 12 + 24 + 43 + 31\} \tag{2.47}$$

We shown H_1 for the simplicial complex in figure 2.8 is $\{23 + 42 + 34, 12 + 24 + 34 + 13\}$. Here one can visualise that the triangle, $\{23+42+34\}$ and quadrilateral $\{12+24+43+31\}$ are cycles which contain the hole/tunnel $\{23+42+34\}$ which does not form the boundary of anything. Only $\{12 + 23 + 31\}$ is a cycle which forms the boundary of the filled-in 1-simplex $\{123\}$.

One can also describe the homology groups through the Betti number $\mathcal{B}_k$ which is calculated as the rank of the homology group H_k. This is expressed in the equation below.

$$\mathcal{B}_k = \text{Rank } H_k = \text{Rank } Z_k - \text{Rank } B_k \tag{2.48}$$

$\mathcal{B}_0, \mathcal{B}_1$ and $\mathcal{B}_2$ counts the number of connected components, holes and voids respectively in a simplicial complex. The Betti numbers for the simplicial complex in figure 2.8 are as follows: $\mathcal{B}_0 = 1, \mathcal{B}_1 = 1, \mathcal{B}_2 = 0$.

In our thesis, we use the *Betti error* to define the topological accuracy of a segmentation output. The Betti Error is a metric used to quantify the dissimilarity in topology between the predicted mask and the ground truth mask. This is determined by calculating the absolute difference between their corresponding Betti numbers.

2.8.4 Persistent Homology

Simplicial homology uses matrix reduction algorithms to describe the connectivity of $\mathbb{K}$ as a sequence of homology groups [ELZ00].

In this thesis, the Betti numbers are useful topological descriptors for the binary labels

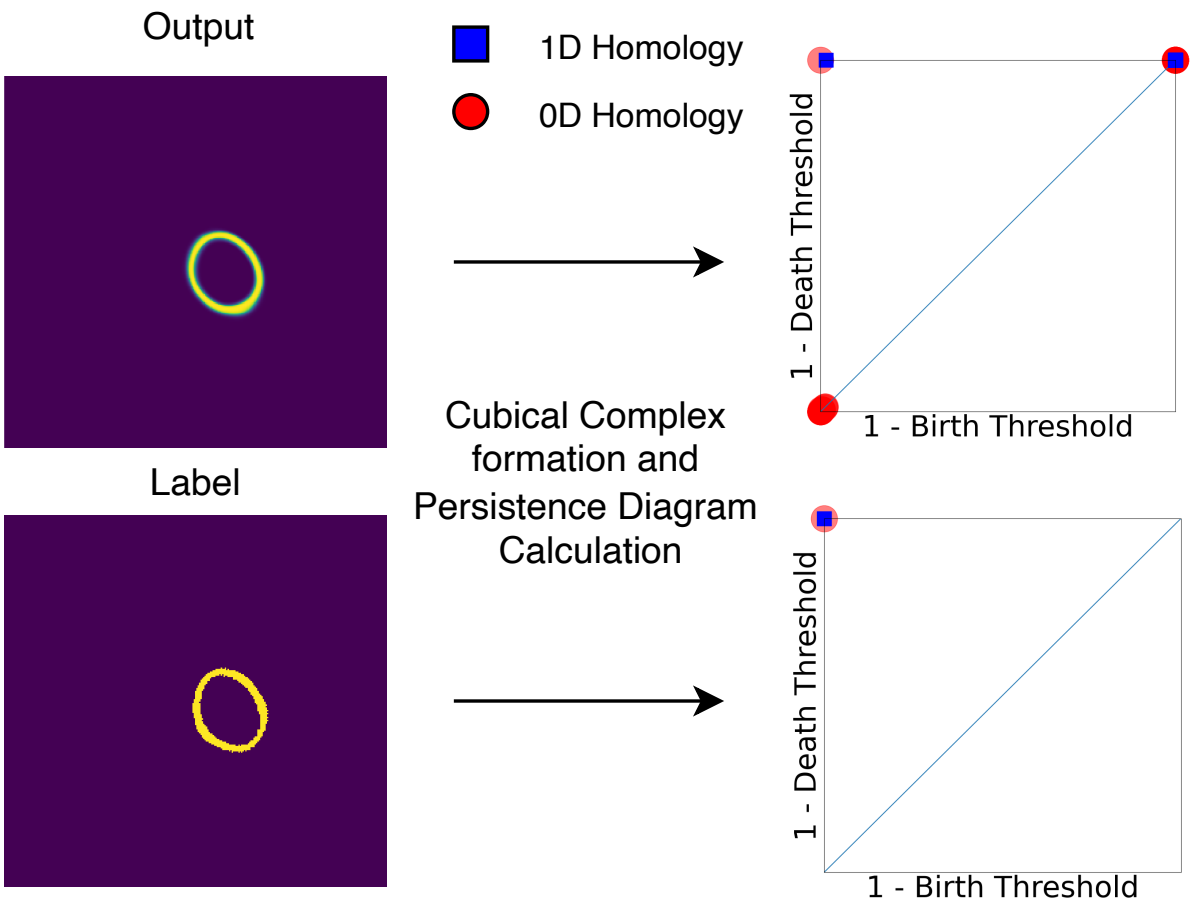


Figure 2.10: Example of persistence diagrams (right) calculated from the softmax model output and segmentation label for the heart myocardium. The x and y axis range from 0 to 1.

but not a continuous segmentation output in $\mathbb{R}^{x \times y \times z}$ because Betti numbers are calculated on a single scale. Therefore, a continuous measure of the change in topological features in each homology group at different scales provides a rich and differentiable descriptor of the topology of data. This is described as persistent homology and requires a filtration function f to track the homology groups over multiple scales, ϵ. A common filtration function is a distance function often used for point cloud data; one refers to this simplicial complex as a Vietoris-Rips complex [EH22]. Medical imaging is however structured as 2D and 3D grids and therefore the cubical complex C is naturally equipped to deal with such data. In the cubical complex, cubes and squares are equivalent to tetrahedra and triangles respectively in the simplicial complex The vertices in the cubical complex of a 2D or 3D image X would correspond to voxels and the edges or connectivity is determined by a grid which connects all voxels in an image.

We define a filtration based on the voxel values (pixel intensities) in X which represents vertices in the cubical complex. Therefore, a cubical complex is constructed at a threshold ϵ over the output defined as: $C^{\epsilon} = \{x \in X | x \geq \epsilon\}$. Given m threshold values between ϵ_1 and ϵ_m, one can construct m cubical complexes from each sub-level set. Thus, this will satisfy a nesting relationship between the cubical complexes of X shown in equation 2.49 making it possible to track changes in the homology groups as ϵ_i decreases.

$$\emptyset = C^{\epsilon_1} \subseteq C^{\epsilon_2} ... \subseteq C^{\epsilon_m} = X \tag{2.49}$$

One can now derive a persistence diagram for n dimensional topological features from tuples (b, d) with b denoting the threshold at which a topological feature is born and d is the threshold at which it dies. In figure 2.10 we provide an example of persistence diagrams for a segmentation output in the top row and the segmentation label in the bottom row. We overlap the persistence diagrams for 0 and 1-dimensional topological features. In the persistence diagram for the label in figure 2.10, ϵ_i decreases from 1 to 0. Here, the red circle with coordinates $(0, 1)$ signifies the birth of a single connected component (the ring) at $\epsilon = 1$ which dies at $\epsilon = 0$. The blue square denotes a tunnel born at $\epsilon = 1$ which also dies at $\epsilon = 0$. The persistence diagram for the output denotes similar topological features but with additional topological features representing noise found close to $(0, 0)$ and $(1, 1)$ due to the continuous output.

2.8.5 Riemannian Geometry

We will now move to the highest level of the hierarchy shown in figure 2.5, the Riemannian manifold. We start with the differentiable manifold. A differentiable manifold is a topological space X equipped with an atlas of coordinate charts such that the transition maps between charts are differentiable. A formal definition is provided in definition 11.

Definition 11. *Given a topological space M, there exists a collection of charts with its associated transition function $\{(U_\alpha, \phi_\alpha)\}$, where U_α is an open set in M and, $\phi_\alpha : U_\alpha \rightarrow \mathbb{R}^n$ is a homeomorphism onto an open set in $\mathbb{R}^n$. M becomes a differentiable manifold if for any two overlapping charts with their associated transition maps, $\{(U_\alpha, \phi_\alpha)\}$ and $\{(U_\beta, \phi_\beta)\}$, the*

transition function, $\phi_\beta \circ \phi_\alpha^{-1}$ is differentiable. If this transition function is infinitely differentiable then this becomes a smooth manifold.

On a differentiable manifold, each point x has an associated tangent space T_xM, which is a vector space that captures the notion of directional derivatives at that point. We define a tangent space and tangent bundle in definition 12 and 13 respectively.

Definition 12. *The tangent space T_xM at a point x on a differentiable manifold M is a vector space consisting of all possible tangent vectors at that point. A tangent vector at x can be thought of as the velocity vector of a curve passing through x.*

Definition 13. *The tangent space varies smoothly over the manifold, leading to the concept of the tangent bundle TB, which is the disjoint union of tangent spaces at all points on the manifold formally defined in the equation below.*

$$TB = \bigcup_{x \in M} T_xM \tag{2.50}$$

We can now define the Riemannian manifold below.

Definition 14. *The Riemannian manifold is a differentiable manifold equipped with the Riemannian metric. The Riemannian metric defines the set of inner products $g_x : \mathcal{T}_x\mathcal{M} \times \mathcal{T}_x\mathcal{M} \to \mathbb{R}$ of every point x on $\mathcal{M}$. The Riemannian metric at every point x must be positive definite.*

The properties of a Riemannian manifold allow us to measure lengths, distances and geodesics on the manifold. We first define the length of a path on a Riemannian manifold below.

Definition 15. *Given a smooth curve $\gamma : [a, b] \to M$ on a Riemannian manifold M with metric tensor g, the length of the curve is defined by the integral:*

$$L(\gamma) = \int_a^b \sqrt{g(\dot{\gamma}(t), \dot{\gamma}(t))}\, dt \tag{2.51}$$

Here:

- *$\gamma(t)$ represents the position of the curve at time t,*

- $\dot{\gamma}(t)$ *is the velocity vector (derivative) of the curve with respect to time,*
- $g(\dot{\gamma}(t), \dot{\gamma}(t))$ *is the inner product of the velocity vector with itself, as defined by the metric tensor* g*,*
- *The square root ensures that we obtain the actual length.*

The integral is taken over the parameter interval $[a, b]$*.*

Equation 2.51 generalises the notion of arc length from Euclidean spaces to curved Riemannian manifolds, and it gives the natural notion of distance along a curve on the manifold.

The geodesic distance is the shortest length between any two points on a Riemannian manifold. This can be expressed as taking the infimum over all lengths connecting two points on the manifold. The process of finding this locally shortest path involves minimizing the length between the points using the principles of the *calculus of variations*.

An important property of the Riemannian manifold is parallel transport which is defined in definition 16.

Definition 16. *Parallel transport* $P_{x \to y}$ *describes the translation of a vector field* V *along a differentiable manifold to a new vector field* V' *such that the covariant derivative always stays at 0.*

2.8.6 Hyperbolic Geometry

Introduction

In this thesis, hyperbolic geometry is used for imposing constraints in the latent space to improve robustness. We will provide a brief overview of important concepts to understand hyperbolic geometry. In Euclidean geometry, the parallel postulate states that, given a line and a point not on it, there is exactly one parallel line through that point. Hyperbolic geometry rejects this postulate, allowing for multiple parallel lines through a given point not on a line. The key property of hyperbolic geometry is that it is characterised by constant negative curvature.

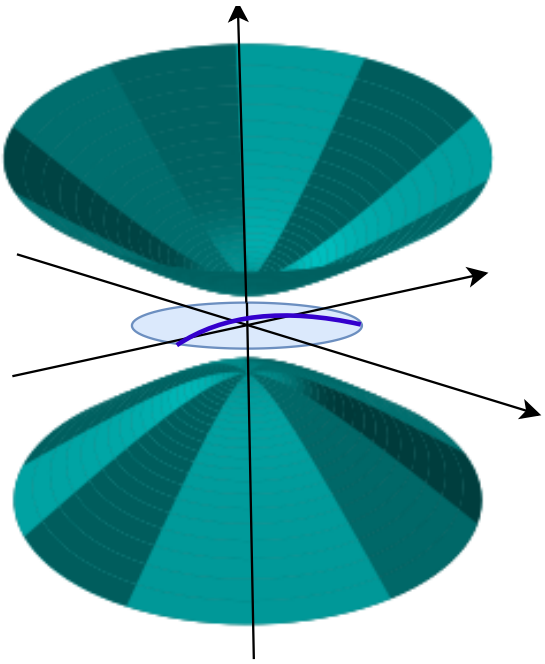


Figure 2.11: Here we provide a visualisation of the two-shift hyperboloid in green with the Poincaré disc visualised in blue with a purple geodesic.

Several models represent hyperbolic geometry, such as the Poincaré disk model and the Poincaré half-plane model. These models provide visualisations of hyperbolic spaces and help understand the properties of hyperbolic geometry. In this thesis, we will focus on the *Hyperboloid* and *Poincaré* models of hyperbolic space ($\mathcal{H}$) equipped with constant negative curvature $-1/K, (K > 0)$ which is inversely proportional to the radius of the Hyperbolic model. In this thesis we work only with the hyperboloid $\mathbb{H}^{d,K}$ and Poincaré $\mathbb{B}^{d,K}$ models of d dimensions with unit radius and hence a fixed negative curvature of -1 ($\mathbb{H}^{d,1}$, $\mathbb{B}^{d,1}$). We show a visualisation of the hyperboloid superimposed on the Poincaré disc in 3D space in figure 2.11.

Geodesic Distance

Hyperboloid We first focus on the two-sheet unit hyperboloid model equipped with a Riemannian metric ($g_x^{\mathbb{H},1}$) given by the Minkowski metric tensor $\langle ., . \rangle_S$ whereby $\langle x, x \rangle_S = -1$ and $x \in \mathbb{R}^{d+1}$ [CYRL19]. Therefore, the dot product between arbitrary points u and v on $\mathbb{H}^{d,1}$, is defined as $g_x^{\mathbb{H},1}\langle u, v \rangle := -u_0 v_0 + u_1 v_1 \ldots + u_d v_d; u_0 > 0$ [CYRL19]. Given the definition of the Riemannian metric tensor, we can introduce the notion of the geodesic distance defined as the shortest distance between $(u, v) \in \mathbb{H}^{d,1}$. The equation for the

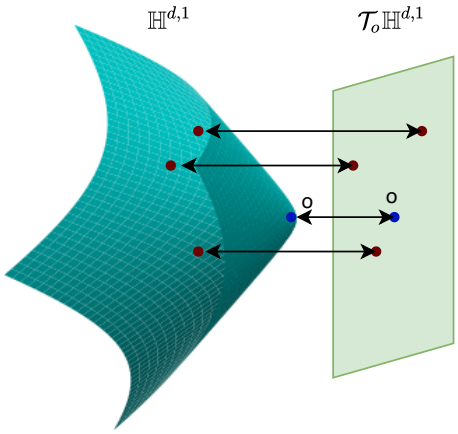


Figure 2.12: Here we show the logarithmic mapping (hyperboloid to tangent space) and exponential mapping(tangent space to hyperboloid). The tangent space is demonstrated on the origin of the hyperboloid.

geodesic distance in the hyperboloid is defined below.

$$d^{\mathbb{H},1}(u,v) = arcosh(g_x^{\mathbb{H},1}\langle u,v\rangle) \tag{2.52}$$

Poincaré Disc We next formally define the Poincaré unit ball, $\mathbb{B}^{d,1} = \{x \in \mathbb{R}^d | \|x\| < 1\}$. $\mathbb{B}^{d,1}$ is determined by the following Riemmanian metric tensor: $g_x^{\mathbb{B},1} = \left(\frac{2}{1-\|x\|^2}\right)^2$. Given this metric tensor, the derived geodesic distance equation for $(u,v) \in \mathbb{B}^{d,1}$ is shown below.

$$d^{\mathbb{B},1}(u,v) = \cosh^{-1}\Big(1 + 2\Big(\frac{\|u-v\|^2}{(1-\|u\|^2)(1-\|v\|^2)}\Big)\Big) \tag{2.53}$$

Exponential and Logarithmic Mapping

Now, we introduce the notion of mapping between hyperbolic and Euclidean space (tangential plane) in the definition below.

Definition 17. *Given $y \in \mathcal{T}_x\mathbb{H}^{d,1}$ and $v \in \mathbb{H}^{d,1}$, exponential and logarithmic mappings are denoted as $exp_x^{\mathbb{H},1}(y)$: $\mathcal{T}_x\mathbb{H}^{d,1} \to \mathbb{H}^{d,1}$ and $log_x^{\mathbb{H},1}(v)$': $\mathbb{H}^{d,1} \to \mathcal{T}_x\mathbb{H}^{d,1}$ respectively.*

Hyperboloid We first show here the general equations $\exp_x^K(y)$ and $log_x^K(v)$ with any negative curvature, $-1/K$ for $\mathbb{H}^{d,K}$ in Equation 2.54 and 2.55 respectively.

$$exp_x^{\mathbb{H},K}(y) = cosh\Big(\frac{\|y\|_S}{\sqrt{K}}\Big)\, x + \sqrt{K} sinh\Big(\frac{\|y\|_S}{\sqrt{K}}\Big)\frac{y}{\|y\|_S} \tag{2.54}$$

$$log_x^{\mathbb{H},K}(v) = d^{\mathbb{H},K}(x,v)\Big(\frac{v + \frac{1}{K}\langle x,v\rangle_S\, x}{\|v + \frac{1}{K}\langle x,v\rangle_S\, x\|_S}\Big) \tag{2.55}$$

In this work, we perform mappings with the tangential space at the origin o, which for $\mathbb{H}^{d,1}$ is defined on $\{1, 0_1, \ldots, 0_d\} \in \mathbb{H}^{d,1}$. In this case, the mappings between $y \in \mathcal{T}_o\mathbb{H}^{d,1}$ and $v \in \mathbb{H}^{d,1}$ such that $\langle y, o\rangle = 0$ are shown in equation 2.56 for the exponential mapping and equation 2.57 for the logarithmic mapping.

$$exp_o^{\mathbb{H},1}(y) = \Big(cosh(\|y_{1:d}\|_2), sinh(\|y_{1:d}\|_2)\frac{y_{1:d}}{\|y_{1:d}\|_2}\Big) \tag{2.56}$$

$$log_o^{\mathbb{H},1}(v) = \Big(0, arcosh(v_0)\frac{v_{1:d}}{\|v_{1:d}\|_2}\Big) \tag{2.57}$$

We show a visualisation of the exponential and logarithmic mapping between the hyperboloid and its tangent space centered on the origin in figure 2.12.

Poincaré A Riemannian metric tensor is conformal to another Riemannian metric if it defines the same angles. In the case of the Poincaré unit ball, there is a smooth conformal mapping, between the Euclidean metric tensor $g_x^{\mathbb{R}}$ and Poincaré metric tensor $g_x^{\mathbb{B},1}$. The Poincaré conformal factor $\lambda_x = \frac{2}{1-\|x\|^2}$ is defined such that $g_x^{\mathbb{B},1} = \lambda_x^2 g_x^{\mathbb{R}}$.

$$exp_x^{\mathbb{B},K}(y) = x \oplus^K \Big(tanh(\frac{\sqrt{K}\lambda_x^K\|y\|}{2})\frac{y}{\sqrt{K}\|y\|}\Big) \tag{2.58}$$

$$log_x^{\mathbb{B},K}(v) = \frac{2}{\sqrt{K}\lambda_x^K} arctanh(\sqrt{K}\|-x \oplus^K v\|)\frac{-x \oplus^K v}{\|-x \oplus^K v\|}) \tag{2.59}$$

Next, the equations for the mapping between $y \in \mathcal{T}_o\mathbb{B}^{d,1}$ and $v \in \mathbb{B}^{d,1}$, with the origin o defined as $\{0, 0_1, \ldots, 0_d\} \in \mathbb{B}^{d,1}$, is shown in equation 2.60 for the exponential mapping and equation 2.61 for the logarithmic mapping.

$$exp_o^{\mathbb{B},1}(y) = \Big(tanh(\|y\|_2)\frac{y}{\|y\|_2}\Big) \tag{2.60}$$

$$log_o^{\mathbb{B},1}(v) = \left(0, arctanh(||v||_2)\frac{v}{||v||_2}\right) \tag{2.61}$$

Parallel Transport

In this section, we reveal the equations of parallel transport for vector b on the tangential space of the origin to the tangential space of v. Formally, we show the general equations of parallel transport, $P^K_{o\to v}(b)$ for the hyperboloid $\mathbb{H}^{K,d}$ in equation 2.62 and Poincaré space $\mathbb{B}^{K,d}$ in equation 2.63 below.

Hyperboloid The general form of this equation, $P^K_{x\to v}(b)$ shown below in Equation 2.62 is derived in [CYRL19].

$$P^K_{o\to v}(b) = b - \frac{\langle log_o(v), b\rangle_S}{d^{\mathbb{H},1}(o,v)^2}(log_o(v) + log_v(o)) \tag{2.62}$$

Poincaré Equation 2.63 is derived in [GBH18].

$$P^K_{o\to v}(b) = \frac{\lambda^K_o}{\lambda^K_v} b \tag{2.63}$$

Projections

We need to apply projections to constrain points to the manifolds and their tangential space after optimising. In our work, all operations are performed in the tangential plane and therefore one needs to only apply projections after the exponential mapping from the tangential space to the manifold which we define below.

Equation 2.64, below shows the projection after exponential mapping y on $\mathcal{T}_x\mathbb{H}^{d,K}$ to the hyperboloid manifold $\mathbb{H}^{d,K}$.

$$Proj(\exp_x^{\mathbb{H},K}(y)) = (\sqrt{1 + ||v_{1:d}||_2^2}, v_{1:d}) \tag{2.64}$$

The projection when mapping a point y on $\mathcal{T}_x\mathbb{B}^{d,K}$ to Poincaré space $\mathbb{B}^{d,K}$ is achieved by normalising $\exp_x^{\mathbb{B},K}(y)$ if $||\exp_x^{\mathbb{B},K}(y)||_2^2 > 1/\sqrt{K}$ or > 1 in the case of the Poincaré unit

ball.

Mapping between the Poincaré disc and the Hyperboloid

Next, the hyperboloid and the Poincaré disc are isomorphic and therefore there exists a diffeomorphic mapping ψ between $u \in \mathbb{H}^{d,1}$ and $v \in \mathbb{B}^{d,1}$ shown in equation 2.65 (Hyperboloid to Poincaré disc) and equation 2.66 (Poincaré disc to Hyperboloid). below.

$$\psi_{\mathbb{H}^{d,1} \to \mathbb{B}^{d,1}}(u_0 ... u_d) = \frac{u_1 ... u_d}{u_0 + 1} \tag{2.65}$$

$$\psi_{\mathbb{B}^{d,1} \to \mathbb{H}^{d,1}}(v_1 ... v_d) = \frac{(1 + ||v||_2^2, 2v_1 ... 2v_d)}{1 - ||v||_2^2} \tag{2.66}$$

2.9 Categories and Sheaves

2.9.1 Category Theory

In chapter 9, the language of category theory is used to formally describe the relation between objects in a concise manner. We therefore give a brief introduction of concepts which are required to understand chapter 9. Informally, a category is a collection of objects where the relation between objects must satisfy a set of rules. We provide the formal definition for a category below.

Definition 18. *A category C consists of the following data:*

- ***Objects:*** *A collection of objects, denoted by $Obj(C)$.*
- ***Morphisms (Arrows):*** *For each pair of objects A, B in C there exists a set of morphisms. ($Hom_C(A, B)$) from A to B. An example of a morphism f is expressed as; $f : A \to B$.*

A morphism must satisfy the following three properties.

- *The first property, a morphism must satisfy is the composition rule. That is for any two morphisms $f : A \to B$ and $g : B \to C$, there exists another morphism, $h : A \to C$.*

- *The second property is that for any object there exists a unique identity morphism $id_A : A \to A$ such that $f \circ id_A = f$.*

- *The final property is composition is associative such that for any three morphisms f, g, h, $(f \circ g) \circ h = f \circ (g \circ h)$.*

There are many mathematical structures which satisfy the properties of a category. For example, the category of all open subsets of a topological space T satisfies a category whereby, all open sets in the topological space are objects and the morphisms between objects are inclusion relations such that $g : A \to B$ if $A \subseteq B$. Another example of a category is a partially ordered set P such for $p, q \in P$, the morphism $f : p \to q$ implies, $p \leq q$.

The next important tool in category theory is a functor. In category theory, a functor is a mapping between two categories, such that the structure of these categories is unchanged. This means that there is a mapping between objects and morphisms in one category to the objects and morphisms in another category which continuously satisfies the required morphism properties of composition and identity. A formal definition is provided below [Awo10].

Definition 19. *Given two categories, $\mathcal{A}$ and $\mathcal{B}$, a functor $F; \mathcal{A} \to \mathcal{B}$ must satisfy the following properties.*

- *Given an object $a \in \mathcal{A}$, $F(a) \in \mathcal{B}$.*

- *For every morphism $f : a \to b$ in $\mathcal{A}$, $F(f) : F(a) \to F(b)$ in $\mathcal{B}$.*

- *F maintains composition such that $F(f \circ g) = F(f) \circ F(g)$ for all morphisms $f : a \to b$ and $g : b \to c$ in $\mathcal{A}$.*

- *F maintains the identity morphism such that for $a \in \mathcal{A}$, the identity morphism $id_a : a \to a$ is mapped by F to the identity morphism $id_{F(a)} : F(a) \to F(a)$ in $\mathcal{B}$*

We also need to understand the contravariant functor which reverses the direction of the morphisms. We define it formally below [Awo10].

Definition 20. *Given two categories, $\mathcal{A}$ and $\mathcal{B}$ be categories. A contravariant functor $F^{op} : \mathcal{A} \to \mathcal{B}$, maps $\forall a \in \mathcal{A}$, to $F(a) \in \mathcal{B}$. However, here every morphism is reversed such that $\forall f : a \to b$ in $\mathcal{A}$, $F^{op}(f) : F^{op}(b) \to F^{op}(a)$ in $\mathcal{B}$. The properties of composition and the identity morphism still hold like the functor.*

We now will define the natural transformation in category theory which relates two functors on a category such that they commute. We provide the formal definition of a natural transformation below [Awo10].

Definition 21. *Given two categories, $\mathcal{A}$ and $\mathcal{B}$ and two functors $F, G : \mathcal{A} \to \mathcal{B}$, a natural transformation $\eta : F \Rightarrow G$ is a map between functors such that for $a \in \mathcal{A}$ and $b \in \mathcal{B}$, $\eta_a : F(a) \to G(a)$ and $\eta_b : F(b) \to G(b)$ commute as shown in commutative diagram below.*

$$\begin{array}{ccc} F(a) & \xrightarrow{\eta_a} & G(a) \\ \downarrow_{F(f)} & & \downarrow_{G(f)} \\ F(b) & \xrightarrow{\eta_b} & G(b) \end{array}$$

There are a few key properties which a natural transformation must satisfy.

- *The above diagram must commute for $\forall a \in \mathcal{A}$ and $\forall f : a \to b$. This is also known as the naturality square.*
- *The next property is vertical composition. Here, given two natural transformations $\eta : F \Rightarrow G$ and $\theta : G \Rightarrow H$, the vertical composition $\theta \circ \eta : F \Rightarrow H$ is defined in the equation below given a category $\mathcal{A}$.*

$$(\theta \circ \eta)_a = \theta_a \circ \eta_a, \forall a \in \mathcal{A} \tag{2.67}$$

- *The final property required is the identity natural transformation. Given the functor, $F : \mathcal{A} \to \mathcal{B}$, then there is an identity natural transformation $1_F : F \Rightarrow F$ shown in the equation below.*

$$1_{F_A} = id_{F(a)} \forall a \in \mathcal{A} \tag{2.68}$$

We now introduce the useful concept of a diagram in category theory. The informal notion of a diagram is related to the idea of smaller categories within a category which represents a collection of objects and morphisms that capture certain relationships or structures within that category. We provide the formal definition below with the use of functors [Awo10].

Definition 22. *Consider the category $\mathcal{C}$ and smaller category $\mathcal{J}$ such as an indexing category. A diagram D of shape $\mathcal{J}$ in $\mathcal{C}$ is defined below.*

$$D : \mathcal{J} \to \mathcal{C} \tag{2.69}$$

We can define the category of $\mathcal{J}$ shaped diagrams in $\mathcal{C}$ as $Func(\mathcal{J}, \mathcal{C})$.

We provide an example below.

Example 1. *Consider a category $\mathcal{C}$ with objects A, B, and C, and morphisms $f : A \to B$ and $g : B \to C$. The diagram D associated with this category might have an index category $\mathcal{J}$ with objects j_1, j_2 and j_3 and morphisms $a : j_1 \to j_2$ and $b : j_2 \to j_3$. The assignment function would map j_1 to A, j_2 to B, j_3 to C as well as a and b to morphisms, f and g respectively in $\mathcal{C}$.*

We finally introduce the notion of limits and co-limits required to understand chapter 9.

We first define the limit of a diagram D in category $\mathcal{C}$.

Definition 23. *Consider a diagram $D : \mathcal{J} \to \mathcal{C}$, where $\mathcal{J}$ is a smaller category and $\mathcal{C}$ is a category. The limit of the diagram D is an object $A \in \mathcal{C}$, denoted $A = \lim_{\leftarrow} D$, and morphisms $\pi_j : A \to D(j) \forall j \in \mathcal{J}$, which satisfies the following universal property [Awo10].*

1. *For any object $B \in \mathcal{C}$ and morphisms $g_j : B \to D(j), \forall j \in \mathcal{J}$, there exists a unique morphism $\varphi : B \to A$ such that the following diagram commutes, $\forall j \in \mathcal{J}$:*

$$\begin{array}{ccc} & & B \\ & \swarrow^{g_j} & \downarrow^{\varphi} \\ D(j) & \xleftarrow{\pi_j} & A \end{array}$$

Therefore, for any choice of morphisms $g_j : B \to D(j)$, there is a unique morphism φ from B to A making the diagram commute.

Next, we define the colimit of a diagram D in category $\mathcal{C}$.

Definition 24. *Consider a diagram $D : \mathcal{J} \to \mathcal{C}$, where $\mathcal{J}$ is a smaller category and $\mathcal{C}$ is a category. The colimit of the diagram D is an object A in $\mathcal{C}$, $A = \varinjlim D$, and morphisms $\iota_j : D(j) \to A$ for each object j in $\mathcal{J}$, satisfying the following property [Awo10].*

1. *For any object B in $\mathcal{C}$ and morphisms $f_j : D(j) \to B \forall j \in J$, there exists a unique morphism $\psi : A \to B$ such that the following diagram commutes, $\forall j \in \mathcal{J}$:*

$$\begin{array}{ccc} A & \xleftarrow{\iota_j} & D(j) \\ \downarrow{\scriptstyle \psi} & & \swarrow{\scriptstyle f_j} \\ B & & \end{array}$$

Therefore, given any choice of morphisms f_j mapping $D(j)$ to B, there exists a unique morphism ψ mapping A to B such that the diagram commutes.

There are multiple other concepts in category theory such as monoids we haven't defined. For a more comprehensive overview of category theory, please refer to [Awo10].

2.9.2 Sheaf Theory

We will now introduce sheaf theory as required in chapter 9. Sheaf theory is a branch of mathematics with applications in algebraic topology and algebraic geometry. Its goal is to provide a framework for glueing local data associated with a topological space to produce new global information. The first place to start is the pre-sheaf. A pre-sheaf is a way of attaching data such as vector spaces, rings etc. to all subsets of a topological space such that structure is preserved between the topological space and the data. We provide a visualisation of a pre-sheaf over some topological space in figure 2.13.

Definition 25. *Given a topological space X, a presheaf is a contravariant functor $\mathcal{F}^{op}$ from the category of all open sets in X to another category, often a category of vector spaces, sets or modules.*

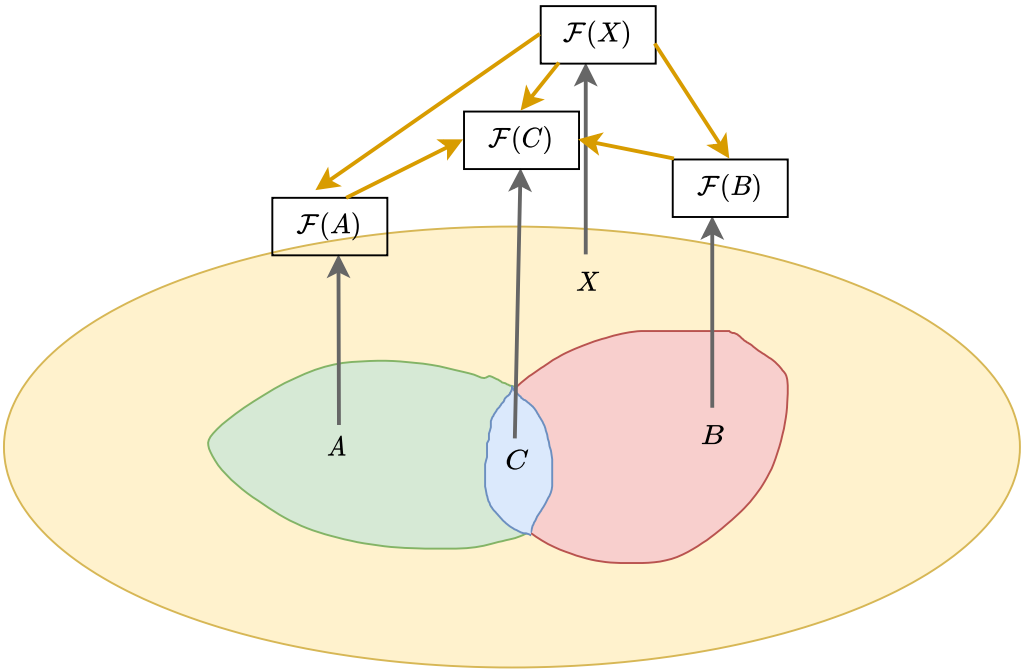


Figure 2.13: Here we show a pre-sheaf over the topological space X where open subsets A, B and its intersection C is mapped with the contravariant functor $\mathcal{F}^{op}$ to $\mathcal{F}(A), \mathcal{F}(B)$ and $\mathcal{F}(C)$ respectively such that inclusion maps in X are mapped with the contravariant functor $\mathcal{F}^{op}$ to restriction maps. This figure is adapted from [Bod23].

For example, given any open set $A \in X$, the presheaf over sets, assigns a set denoted as $F(A)$, and for any inclusion $V \subseteq U$, there is a restriction map $\rho_{A,B} : \mathcal{F}^{op}(A) \to \mathcal{F}^{op}(B)$ satisfying all the properties of a contra-variant functor.

We introduce new terminology called the sections which is defined as given $A \subseteq X$, elements in $\mathcal{F}(A)$ are called sections of $\mathcal{F}$ over A. We also introduce new notation for simplicity; given the subsets $B \subseteq A$ in a topological space X and any section $s \in \mathcal{F}(A)$, we denote, $S|_A := \rho_{A,B}(s)$ to be the restriction map, $\rho_{A,B}(s)$ applied to s.

Definition 26. *The sheaf is a pre-sheaf with two additional properties of locality and glueing which are formally defined below given a topological space with open cover $(A_i)_{i \in I}$.*

- *The notion of locality is defined given any sections $s, t \in \mathcal{F}(A_i)$, $s = t$ if $s|_{A_i} = t|_{A_i} \forall i \in I$.*
- *The notion of gluing is defined, given $s_i|_{u_i \cap u_j} = s_j|_{u_i \cap u_j} \forall i, j \in I$, then there is a section $s \in \mathcal{F}(U_i)$ where $s|_{U_i} = s_i, \forall i \in I$. We call s here the glueing of the sections.*

The glueing property constrained by locality implies the glueing, s is unique. It also implies that sections which agree on their overlap can be glued together. This property

of sheaves underlies the principle of compositionality that an object or data consists of the sum of its parts. An example of a pre-sheaf which is also a sheaf is the pre-sheaf of continuous function over the real line.

In chapter 9, it is important to introduce the pre-sheaf and sheaf over the basis of the topological space X.

Definition 27. *Given a topological basis $\mathcal{B}$ for X. A presheaf a is a contra-variant functor $\mathcal{F}^{op}$ which maps each open set in the basis to some set or algebraic structure. Formally, $\forall U \in \mathcal{B}$, there exists a set, $\mathcal{F}^{op}(U)$, and for any inclusion $V \subseteq U$, there is a restriction map, $\rho_{UV} : \mathcal{F}^{op}(U) \to \mathcal{F}^{op}(V)$. $\mathcal{F}^{op}$ is a contra-variant functor so it satisfies the properties of composition and the identity morphism.*

Definition 28. *A sheaf over the basis $\mathcal{B}$ over a topological space X, is a pre-sheaf which satisfies the properties of locality and gluing required of a sheaf.*

We now have the fundamentals to introduce cellular sheaves which form the core component of chapter 9. In order to understand cellular sheaves we need to define a poset and cell complex.

Definition 29. *A* poset *is a partially ordered set with a binary relation, $\leq$ between elements in a set which is reflexive, anti-symmetric, and transitive [Hu20].*

Example 2. *We define two elements in our poset as two n-tuples, $(a_1, a_2, \ldots, a_n)$ and $(b_1, b_2, \ldots, b_n)$, whereby each element belongs to the set of binary integers $\mathbb{Z}/2\mathbb{Z}$. We define the relation $(a_1, a_2, \ldots, a_n) \leq (b_1, b_2, \ldots, b_n)$ if, and only if, $a_i \leq b_i$ for all i in the set $\{1, 2, \ldots, n\}$. In this context, the ordered set $(\mathbb{Z}/2\mathbb{Z}^n, \leq)$ forms a partially ordered set (poset). Therefore, a 3 element poset, $a, b, c \in P$ must satisfy: 1. $a \leq a$, 2. $a \leq b$ and $b \leq c$ implies $a \leq c$ and 3. $b \leq a$ and $a \leq b$ implies $a = b$*

Definition 30. *A topological space X is a cell complex when it is divided into pieces $X_{p \in P_x}$ where P_x is the indexing set with the following characteristics:*

- *$\forall x \in X$ with a neighborhood U intersects only a finite number of X_p.*
- *Every point X_p is homeomorphic to $\mathbb{R}^k$ for some k.*

- *If $X_q \cap X_p$ is non-empty, then $X_p \subseteq X_q$ which defines X_p to be a face of X_q. Thus, this face relation establishes a poset structure on the indexing set P_x.*

We can now define a cellular sheaf.

Definition 31. *Given a cell complex with poset structure, X, P_x, a cellular sheaf $\mathcal{S}$ is a functor which maps every cell in a topological space X to some algebraic object such as a set or vectors space D defined, $\mathcal{F} : Cell(X) \to D$. This infers the following:*

- $\forall X_p \in X$, $\mathcal{F}(p)$ *exists.*
- *For every pairs of cells, $X_p \subset X_q$, there exists a restriction map $\rho_{\mathcal{F}_{p,q}} : \mathcal{F}(p) \to \mathcal{F}(q)$. We can understand this explicitly in terms of the relations defined in the poset structure P_x such that for $p, q \in P_x$ and $p \leq q$, then there is functor, $\mathcal{F}_{p \leq q} : \mathcal{F}(p) \to \mathcal{F}(q)$ which satisfies the rules of composition and identity morphisms required of a functor.*

Cellular sheaves assign data on their cells as opposed to open sets which makes it finite. However, one can assign a topology to a poset known as an Alexandrov topology which turns cellular sheaves into actual sheaves. We explore the cellular sheaf over the poset to assign vector spaces in more detail in chapter 9. Further information and details about cellular sheaves can be found in [Cur14].

2.10 Measure Theory

In chapter 9, we also use measure theory to demonstrate the advantage of using cellular sheaves over persistent homology from a geometric perspective for shape analysis. We therefore provide the fundamentals of measure theory.

We must first define the σ-algebra, denoted $\mathcal{S}$ on a set X. This is a collection of subsets of X which obey the following axioms.

- The whole set and empty set is in the σ-algebra; $\emptyset, X \in \mathcal{S}$.
- The complement of every subset in $\mathcal{S}$ also exists. That is if $A \in \mathcal{S}$, then $A^c \in \mathcal{S}$.

- $\mathcal{S}$ is closed under countable union. Therefore, if $A_1, A_2, \ldots A_n \in \mathcal{S}$, then $\bigcup_{i=1}^{i=n} A_i \in \mathcal{S}$.

A useful theorem to understand about σ-algebras is the following:

Theorem 2. *The intersection of any two σ-algebras is also a σ-algebra.*

We prove theorem 2 as follows.

Proof. Given, two σ-algebras $\mathcal{S}_1$ and $\mathcal{S}_2$ b on a set X. We show that $\mathcal{S}_1 \cap \mathcal{S}_2$ is also a σ-algebra.

- **Contains the whole and empty Set:** Since any σ-algebra contains the whole set X and empty set $\emptyset$, then it follows, $\emptyset, X \in \mathcal{S}_1 \cap \mathcal{S}_2$.
- **Closed under Complements:** Let $A \in \mathcal{S}_1 \cap \mathcal{S}_2$. This implies that $A \in \mathcal{S}_1$ and $A \in \mathcal{S}_2$. Since $\mathcal{S}_1$ and $\mathcal{S}_2$ are both σ-algebras, $A^c \in \mathcal{S}_1$ and $A^c \in \mathcal{S}_2$. Therefore, $A^c \in \mathcal{S}_1 \cap \mathcal{S}_2$.
- **Closed under Countable Unions:** Let $A_1, A_2, \ldots A_n \in \mathcal{S}_1 \cap \mathcal{S}_2$. This implies that $A_i \in \mathcal{S}_1$ and $A_i \in \mathcal{S}_2$ for all i. Since $\mathcal{S}_1$ and $\mathcal{S}_2$ are σ-algebras, $\bigcup_{i=1}^{i=n} A_i \in \mathcal{S}_1$ and $\bigcup_{i=1}^{i=n} A_i \in \mathcal{S}_2$. Therefore, $\bigcup_{i=1}^{i=n} A_i \in \mathcal{S}_1 \cap \mathcal{S}_2$.

□

Since $\mathcal{S}_1 \cap \mathcal{S}_2$ satisfies all the properties of a σ-algebra, we conclude that $\mathcal{S}_1 \cap \mathcal{S}_2$ is a σ-algebra.

The σ-algebra defined is for closed sets and therefore we now define the σ-algebra for open sets called the Borel σ-algebra.

Definition 32. *The Borel σ-algebra, denoted as $\mathcal{B}$, is the smallest σ-algebra containing all open sets.*

Borel σ-algebras are useful for defining measures over topological spaces and an example is the Borel σ-algebra over the real line denoted, $\mathcal{B}(\mathbb{R})$. Note a set in $\mathcal{B}$ is a Borel set.

We next define a measurable and measure space.

Definition 33. *A measurable space is defined as,* $(X, \mathcal{S})$*, where* X *is a set and* $\mathcal{S}$ *is a* σ*-algebra on* X*.*

Definition 34. *A measure space is a measurable space with a measure* μ *defined on* $(X, \mathcal{S})$ *to construct the triple,* $(X, \mathcal{S}, \mu)$*. A measure is a function* $\mu : \mathcal{F} \to [0, \infty]$ *with the following properties:*

- $\mu(\emptyset) = 0$ *(empty set has measure zero).*
- *It satisfies the property of sub-additivity (*σ*-additivity). That is, given disjoint sets,* $A_1, A_2, \ldots A_n \in \mathcal{S}$*, then the equation below is true.*

$$\mu\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{i=n} \mu(A_i) \tag{2.70}$$

We now will define a few examples of measures. First, we introduce the probability measure.

Definition 35. *A probability measure* $\mathbb{P}$ *on a measurable space* $(X, \mathcal{S})$ *is a measure* $\mathbb{P} : \mathcal{S} \to [0, 1]$ *such that* $\mathbb{P}(X) = 1$*.*

The counting measures is a useful and simple measure defined below.

Definition 36. *The counting measure on a measurable space* $(X, \mathcal{S})$ *is a measure* $\mu : \mathcal{S} \to [0, \infty)$ *defined by*

$$\mu(A) = \begin{cases} \textit{cardinality}(A), & \textit{if } A \textit{ is finite,} \\ \infty, & \textit{if } A \textit{ is infinite.} \end{cases}$$

We now define a common measure called the Borel measure.

Definition 37. *A Borel measure is a measure defined on the Borel* σ*-algebra* $\mathcal{B}$ *which satisfies the following two properties below. We define* $\mathcal{B}$ *over* $\mathbb{R}$ *to highlight the two properties.*

- ***Translation Invariance:*** *If $B \in \mathcal{B}(\mathbb{R})$, then $\forall r \in \mathbb{R}, B+r = \{x+r \mid x \in B\}$, $B+r \in \mathcal{B}(\mathbb{R})$, and $\mu(E) = \mu(E+a)$.*

- ***Countable Additivity:*** *If $\{B_n\}$ is a countable collection of pairwise disjoint Borel sets; $B_i \cap B_j = \emptyset$ for $i \neq j$, then the equation below is true.*

$$\mu\left(\bigcup_{n=1}^{n=\infty} B_n\right) = \sum_{n=1}^{n=\infty} \mu(E_n) \tag{2.71}$$

The Lebesgue measure is a way of assigning a “size" or “length" to sets and extends this notion to more complicated sets.

Definition 38. *The Lebesgue measure m on $\mathbb{R}$ is an example of a Borel measure. It is defined on the Borel σ-algebra and has the following properties:*

- $m((a,b]) = b - a \quad$ *for any interval* $(a,b]$

- *It also satisfies translation invariance and countable additivity for a Borel measure.*

- *It is monotone defined as $A \subseteq B \subseteq \mathbb{R} \rightarrow 0 \leq m(A) \leq m(b) \leq \infty$*

We now need to define what is required for a function to be measurable below.

Definition 39. *Given two measurable spaces $(X, \mathcal{S})$ and $(Y, \mathcal{T})$, a function $f : X \rightarrow Y$ is called measurable if, $\forall t \in \mathcal{T}$, the pre-image, $f^{-1}(t)$ belongs to $\mathcal{S}$ ($f^{-1}(t) \in \mathcal{S}$).*

An important theorem to be aware of when dealing with measurable functions is the monotone convergence theorem which is shown below.

Theorem 3. *Given an increasing sequence of non-negative measurable functions on $(X, \mathcal{F}, \mu)$; $f_1, f_2, \ldots f_n$ and $\lim_{n\rightarrow\infty}, f_n(x) = f(x)$ $\forall x$ then the equation below holds:*

$$\lim_{n\rightarrow\infty} \int_X f_n \, d\mu = \int_X f \, d\mu \tag{2.72}$$

If we now extend the sequence of non-negative measurable functions to integrable functions we have the dominated convergence theorem defined below.

Theorem 4. *If* $|f_n| \leq g, \forall n$, g *is integrable, and* f_n *converges pointwise to* f*, then equation 2.72 again holds.*

There are further concepts and theorems in measure theory such as product measures, Caratheodory's extension theorem and Fubini's Theorem which are however irrelevant to this thesis. A more comprehensive overview of measure theory with further details can be found in [Hal13].

Chapter 3

Datasets

In this thesis, we use several datasets which are used repeatedly in each chapter. We will therefore describe all the datasets used in this thesis. This chapter is adapted from my early-stage assessment.

3.1 External Datasets

3.1.1 Beyond the Cranial Vault (BTCV) dataset

We use the Beyond the Cranial Vault (BTCV) dataset consisting of 30 CT scans of the Abdomen with 13 labels acquired from a single domain (Vanderbilt University Medical Center) [LXI+15]. The CT scans were performed with contrast (portal venous phase) with volumes ranging from $12 \times 512 \times 85$ to $512 \times 512 \times 198$ and field of views ranging from $280 \times 280 \times 280$ mm^3 to $500 \times 500 \times 650$ mm^3. The in-plane resolution ranged from 0.54×0.54 mm^2 to 0.98×0.98 mm^2. The slice thickness varied from 2.5 mm to 5.0 mm. The segmentation labels used for this dataset include: (1) spleen, (2) right kidney, (3) left kidney, (4) gallbladder, (5) esophagus, (6) liver, (7) stomach, (8) aorta, (9) inferior vena cava, (10) portal vein and splenic vein, (11) pancreas, (12) right adrenal gland and (13) left adrenal gland.

This single domain dataset is used in chapter 5 and 6.

3.1.2 NCI-ISBI13 Challenge Dataset

The data from the NCI-ISBI13 Challenge is a prostate dataset acquired from two domains [BMH+15b, CVS+13, BMH+15a]. It consists of 60 T2 weighted MRI scans of which half come from Boston Medical Centre (BMC) acquired on a 1.5T scanner with an endorectal coil and the other half is acquired from Radboud University Nijmegen Medical Centre (RUNMC) on a 3T scanner with a surface coil [BMH+15b]. We summarise the acquisition parameters of the T2 images in both datasets in table 3.1.

Institution	Number of Cases	Field Strength (T)	Axial Resolution (mm)	Slice Thickness	Endorectal Coil	Sequence	Manufacturer
RUNMC	30	3	0.6-0.625	3.6-4	Surface	Turbo Spin-Echo	Siemens
BMC	30	1.5	0.4	3	Endorectal	Turbo Spin-Echo	Philips

Table 3.1: T2 weighted image acquisition parameters for the BMC and RUNMC dataset from the NCI-ISBI13 Challenge.

Segmentations of the Transitional zone and Peripheral zone provide two segmentation labels for this dataset. This dataset is used for single domain generalisability in chapter 5, 7, 8 and 6.

3.1.3 Segmentation Decathlon (Prostate)

We use the Prostate dataset from the segmentation decathlon which contains 32 T2-weighted and apparent diffusion coefficient (ADC) images from the Radboud University Nijmegen Medical Centre (RUNMC) [ARB+22]. The T2 weighted images have the same acquisition parameters as the RUNMC dataset in the NCI-ISBI13 Challenge shown in table 3.1. We use this dataset in chapter 7 and 9.

3.1.4 ProstateX challenge

This dataset is made up of 204 T2-weighted axial images, diffusion-weighted imaging (b-800), ADC maps and K-trans images from the ProstateX challenge which were also acquired from RUNMC [LDB+14]. We however only use the T2-weighted and ADC

images for this thesis. This is because diffusion-weighted and contrast imaging is dynamic and hence are 4D images which makes the input volume too large. Also, DCE is not available in many centres which makes it hard to test domain generalisability. ADC also captures the diffusion characteristics contained in diffusion-weighted imaging by calculating the apparent diffusion coefficients. Images were acquired with a Siemens 3T scanner and without an endorectal coil. We show the acquisition parameters for the T2 weighted and ADC images below.

T2			ADC		
Slice Thickness	Axial Resolution	Sequence	Slice Thickness	Axial Resolution	Sequence
3.6mm	0.5mm	turbo SE	3.6mm	2mm	single-shot EP

Table 3.2: T2 and diffusion-weighted image acquisition parameters for the ProstateX dataset. ADC maps in both datasets are calculated with b-values; 50, 400 and 800.

The ProstateX dataset also provides the location and grading of lesions in each image. This dataset consists of 330 pre-selected lesions with lesion grading. All lesions were graded by a radiologist with 20 years of experience according to the PIRADS score [BTC15]. PIRADS scores above 2 were considered suspicious and sent for biopsy under MRI guidance. 112 of the 330 lesions were biopsied and graded according to Gleason grading by a pathologist with more than 30 years of experience. A Gleason grade of 2 and above is considered clinically significant. The cohort's median age stood at 66 years, ranging from 48 to 83 years. The median PSA level was 13 ng/ml, ranging from 1 to 56 ng/ml. The distribution of patients across Gleason Grade Groups of those biopsed was as follows: 32% in Group 1, 36% in Group 2, 16% in Group 3, 9% in Group 4, and 7% in Group 5. The dataset is highly imbalanced with only 23 per cent of lesions classed as clinically significant (Gleason grade group 2 and above) if we consider those not biopsied to have a Gleason grade group of 1. Please refer to [LDB+17, LKB+17] for further information on this dataset. The segmentation labels of the prostate zones and tumours for this dataset were provided by [CSC+21, CCS+21]. This dataset is used in chapter 4 and 9.

3.1.5 NIH chest X-ray Dataset

The National Institute of Health (NIH) dataset consists 108,948 posterior-anterior chest radiographs (from 32,717 patients) acquired from the NIH medical centre. Labels are provided for each image which include whether this is a normal chest X-ray or the name of the pathology in the image such as pneumonia or lung cancer. Annotations are generated using Natural Language Processing (NLP) techniques applied to radiological reports. Images are provided in a 1024x1024 pixel matrix size. The segmentations of both lungs are provided by [TTXS19] for 100 images which are used in this thesis. This dataset is used in chapter 5.

3.1.6 JSRT chest X-ray Dataset

The Japanese Society of Radiological Technology (JSRT) dataset consists of 34.6×34.6 *cm* posterior-anterior chest x-rays acquired from 13 medical centres in Japan and 1 medical centre from the USA [SKI+00]. The dataset consists of 154 conventional chest radiographs, including 100 with malignant lung nodules and 54 with benign nodules. Additionally, there are 93 radiographs without nodules. The images were digitized using a laser digitizer with a 2048x2048 pixel matrix size and a 12-bit grey scale. Each pixel represents 0.175 mm. The data is in big-endian raw format without headers. Supplementary information such as patient age, gender, diagnosis (malignant or benign), X and Y coordinates of nodules, and simple diagrams of nodule locations are provided. The segmentations of both lungs are provided for the 154 radiographs with nodules. This dataset is used in chapter 5.

3.1.7 M & M cardiac dataset

This dataset is the Multi-centre, multi-vendor & multi-disease (M&Ms) cardiac imaging 3-class segmentation dataset [CGI+21] which is divided into 4 domains determined by the MRI scanner vendor. The images are acquired from 6 hospitals in Spain, Canada and Germany (1. Hospital Vall d'Hebron, 2. Clinica Sagrada Familia, 3. Universitatsklinikum Hamburg-Eppendorf, 4. Hospital Universitari Dexeus, 5. Clinica Creu

Blanca and 6. McGill University Health Centre). T2 weighted images are acquired from the following 4 vendors; Siemens, Philips, GE and Canon. We provide information on the Vendors below. Short axis views were obtained in both end-systole and diastole for each patient. There are 95 scans in domain A (Siemens), 125 scans in domain B (Philips) and 50 scans in domain C (GE) and D (Canon). We show the acquisition parameters in table 3.3.

Centre	Vendor	Model	Field Strength (T)	Axial Resolution (mm)	Slice Thickness	Number of Slices	Number of Time frames
1	Siemens	MAGNETOM Avanta	1.5	1.32	9.2	12	25
2	Philips	Achieva	1.5	1.2	9.9	10	30
3	Philips	Achieva	1.5	1.45	9.9	11	26
4	GE	Signa Excite	1.5	1.36	10	12	25
5	Canon	Vantage Orian	1.5	0.85	10	13	29
6	Siemens	MAGNETOM Skyra	3	0.98	9.7	12	29

Table 3.3: Vendor description for each site in the M&M dataset.

The images consisted of patients who were healthy as well as having a range of pathological conditions such as myocarditis and ischemic cardiomyopathy. Segmentation labels were provided for 3 anatomical structures; the left ventricle (LV) and right ventricle (RV) and the left ventricle myocardium (MYO). This dataset is used in chapter 7, 8 and 9.

3.1.8 MNIST

The MNIST dataset is a popular collection of 70,000 handwritten digit images (28×28 pixels) [Den12]. Each image is labelled with the corresponding digit (0-9). This dataset is used in chapter 10.

3.1.9 AFHQ

The Animal FacesHQ (AFHQ) dataset is a collection of animal faces that comprise 15,000 high-quality images, and each image has a resolution of 512×512 pixels [CUYH20]. The dataset is divided into three domains: cat, dog, and wildlife, with each domain containing 5,000 images. Images are aligned so that the eyes are positioned at the centre. This dataset is used in chapter 10.

3.1.10 STL10

The STL-10 dataset is an image dataset that originated from ImageNet [CNL11]. It consists of a total of 113,000 images, of which 100,000 are unlabeled. The remaining 13,000 images are labelled and distributed across 10 object classes, such as birds, boats and dogs. All images in the dataset are in colour and have a resolution of 96×96 pixels. This dataset is used in chapter 10.

3.1.11 MIMIC

MIMIC-CXR is a comprehensive dataset openly accessible to the public, consisting of de-identified chest radiographs sourced from patients admitted to the Beth Israel Deaconess Medical Center spanning the years 2011 to 2016 [JPB+19, JPG+19]. The dataset encompasses 371,920 chest X-rays linked to 227,943 imaging studies with some studies coming with both frontal and lateral radiographs. The images are annotated with 14 labels obtained through a natural language processing tool applied to the accompanying radiology reports. All images were processed with histogram equalisation after being normalised between 0 and 255. Images were converted from DICOM to JPG files with a quality factor of 95 using OpenCV. This dataset is used in chapter 10.

3.2 Internal Dataset

3.2.1 Dataset Description

We construct a dataset for tumour classification and segmentation internally. This dataset is a prostate MRI dataset.

The dataset comprises information on 409 prostate cancer patients who underwent radical prostatectomy surgery after pre-operative MRI. The dataset also includes a prospective outcome follow-up database with a median follow-up period of 5 years (ranging from 1 to 12 years), covering the years 2006 to 2018. The inclusion and exclusion criteria were applied to filter the patients.

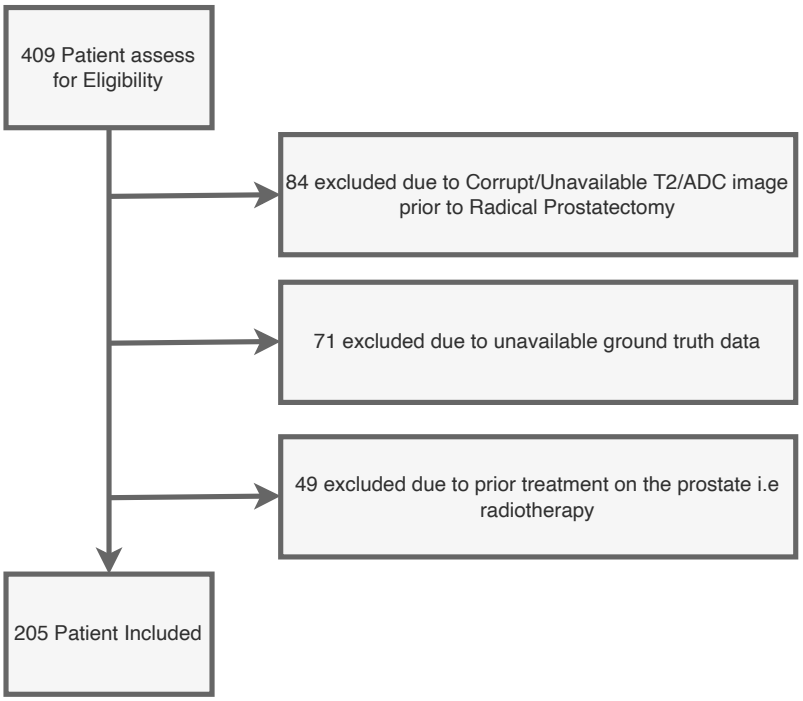


Figure 3.1: Exclusion criteria applied to 409 patients to obtain an image dataset from 205 patients.

The inclusion criteria for this dataset are shown below.

- Patients with pre-radical prostatectomy multi-parametric MRI scans containing uncorrupted T2-weighted or ADC images.
- Lesions with complete information on the Gleason Grade.

The exclusion criteria for this dataset are shown below.

- Patients who did not undergo MRI scans prior to radical prostatectomy.
- Corrupted or incomplete MRI scans.
- Unavailable ground-truth labels (whole mount histopathology).

We show in a flowchart in figure 3.1 the exclusion criteria applied to the 409 patients to finally include 205 patient's MRI images in the dataset.

Scans were acquired on a Siemens or Philips 1.5/3Tesla scanner with a surface coil. Specifically, 40 images were acquired on a 1.5T Philips scanner, 126 images were acquired on a 3T Siemens scanner and 39 images were acquired on a 1.5T Siemens scanner.

Segmentation was carried out by a senior radiologist with at least 5 years experience. Parameters for the T2/ADC acquisition are shown in the table below.

T2			ADC		
Slice Thickness	Axial Resolution	Sequence	Slice Thickness	Axial Resolution	Sequence
3-4mm	0.325-0.625mm	turbo SE	3.6-4mm	2-2.5mm	single-shot EP

Table 3.4: T2 and diffusion-weighted image acquisition parameters for the internal dataset. ADC maps are calculated with b-values; 50, 400 and 800.

The distribution of patients across Gleason Grade Groups was as follows: 11% in Group 1, 32% in Group 2, 28% in Group 3, 18% in Group 4, and 11% in Group 5. This dataset is used in chapter 4 and 9.

3.2.2 Anonymisation and Ethics

All images were pseudonymised for use. The process involves exporting MRI scans in a pseudonymised DICOM format using a Clinical Trial Processor (CTP). A secure Linux Virtual Machine (VM) hosts the CTP Server, which connects to the Live Carestream PACS for pseudonymisation and the Carestream Research PACS/VNA for storing pseudonymised images. During pseudonymisation, patient names are replaced with study numbers, linking them to their pseudonymised datasets. Access to personal/identifiable data is restricted to the clinical care team on NHS computers, while study researchers access pseudonymised data via university computers.

The study has received ethical approval from HRA and Health and Care Research Wales.

Chapter 4

Robust Disease Classification on MRI using Transformers with Discrete Representations

This chapter is adapted from the following two publications.

- **Santhirasekaram, A.**, Pinto, K., Winkler, M., Aboagye, E., Glocker, B. and Rockall, A., 2021. Multi-scale Hybrid Transformer Networks: Application to Prostate Disease Classification. In Multimodal Learning for Clinical Decision Support: 11th International Workshop, ML-CDS 2021, Held in Conjunction with MICCAI 2021, Strasbourg, France, October 1, 2021, Proceedings 11 (pp. 12-21). Springer International Publishing.
- **Santhirasekaram, A.**, Winkler M, Rockall A, Glocker B. Robust prostate disease classification using transformers with discrete representations. International Journal of Computer Assisted Radiology and Surgery. 2024 May 13:1-0.

4.1 Introduction

In this chapter, we will explore the robustness of vision transformers for prostate disease classification on MRI. Automated prostate disease classification on multi-parametric

MRI has recently shown promising results with the use of CNNs. The vision transformer is a convolutional-free architecture which only exploits the self-attention mechanism and has surpassed CNNs in some natural imaging classification tasks but only when pre-trained on large-scale imaging datasets. There is also limited work in using transformers in the task of prostate disease classification. This is because, in a small data regime such as prostate MRI, the use of transformer-only networks is limited and prone to over-fitting. Additionally, CNNs and vision transformers are not very robust to textural shifts in the input space. In MRI we often have to deal with textural shifts arising from varying acquisition protocols.

Here, we focus on the ability of models to generalise well to new magnet strengths for MRI. Transformers have had their greatest impact on language models which inherently deal with discrete input. Therefore, we propose to limit the input space of the vision transformer to a finite number of components. This is achieved using the VQ-VAE which is a CNN-based model to derive discrete representations for input into the transformer. We hypothesise this will improve the robustness of vision transformers for disease classification of lesions on prostate MRI by tackling textural shifts. Furthermore, this method exploits the local inductive bias of CNNs useful in low data regimes to extract CNN-based features which are discretised for input to the transformer, thus, combining the advantages of convolutions and MHSA. We analyse the robustness of our model by assessing domain shift from a 1.5T to a 3T scanner and vice versa which showed a significant performance improvement compared to models with continuous representations.

4.2 Related Work

4.2.1 Advancements on the Vision Transformer

The performance of the vision transformer is hugely dependent on pre-training on large datasets. Optimal performance was achieved in using the full JFT-300M for pre-training [DBK+20]. This is due to the lack of locality imposed within the vision transformer; a useful inductive bias induced in CNNs. Local pixels are highly correlated, and this is

a useful prior within architecture design, which in the case of the vision transformer requires large-scale pre-training. In fact, it has been shown that under certain fixed parameter choices, MHSA with N heads and learned relative positional encodings can learn a convolutional kernel of size $\sqrt{N} \times \sqrt{N}$ [CLJ20]. [dTL+21] exploited this setting to improve vision transformers through a soft convolutional inductive bias. They used gated positional self-attention (GPSA) to incorporate this soft convolutional inductive bias and initialise the layers with the locality of a convolution. During training, the GPSA layers have the option to escape locality imposed during initialisation to learn more global features [dTL+21]. Empirically, it was found that in the initial layers, the locality imposed is kept while escaping in the latter layers [dTL+21]. This highlights the usefulness of imposing locality in extracting visual features, particularly in earlier layers. However, the convolutional locality imposed and described in [CLJ20] and [dTL+21] is between patches and ignores local information within patches. This still prevents the learning of fine local features which may be important in medical imaging tasks where fine textural patterns may be predictive of the presence of pathology. Furthermore, self-attention between patches is quadratic in computational complexity which makes vision transformers more computationally expensive than CNNs.

There have been numerous advancements in improving the performance of the vision transformer in low-data regimes. The majority of the technical modifications involve incorporating convolutions into the vision transformer architecture. For example, the data efficient image transformer (DeiT) uses a pre-trained CNN as a teacher model with its output used as hard labels for the vision transformer loss function [TCD+21]. This method has been shown to impose a convolutional inductive bias into the vision transformer and outperformed vision transformers in the low data regime. Transformer in Transformer (TnT) exploited locality within a patch by applying self-attention between pixels in a patch before learning long-range dependencies between patches [HXW+21].

Other methods, directly incorporate convolutions into the vision transformer architecture. For example, the convolutional vision transformer (CvT) introduces a strided convolution operation to progressively increase the number of feature maps and down-sample before each transformer layer [WXC+21]. This is like the Pyramid Vision Transformer (PVT) which uses sub-sampling in the projections to maintain the hierarchical

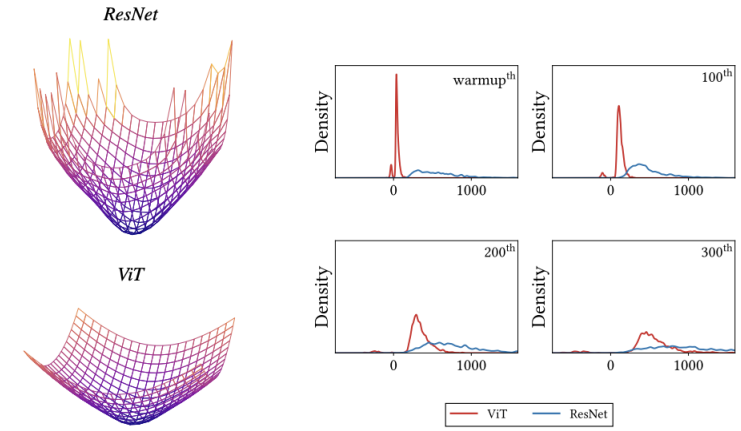
structure of CNNs and overcome the limitation of the vision transformer applying self-attention only on a single scale [WXL+21]. The CvT additionally replaces the linear projections of queries, keys and values with convolutional layers [WXC+21]. CvT outperformed the vision transformer when pre-trained on ImageNet22k, and fine-tuned on ImageNet1k. 'Convolutional neural networks meet vision transformers' (CMT) adopt a similar approach of alternating depth-wise convolutional and MHSA layers [GHW+22]. CMT additionally replaces the MLPs with convolutional layers [GHW+22]. 'LeVIT: a hybrid neural network for fast inference image classification', also uses 1×1 convolutions for Q, K, and V projections but applies progressive sub-sampling within the transformer layers [GENT+21]. LeVIT applies the transformer layers on top of a convolutional backbone as opposed to alternating transformer and convolutional layers [GENT+21].

The Swin transformer model imposes locality constraints by adopting a hierarchical approach [LLC+21]. This is performed by dividing the input image into non-overlapping windows and applying transformer-based processing at each level. Therefore, by integrating local and global processing, the Swin transformer achieves superior performance on tasks such as object detection and semantic segmentation [TYL+22].

4.2.2 Robustness of the Vision Transformer

There have been several works which demonstrated the increased robustness of the vision transformers compared to CNNs. For example, the vision transformer has shown to be robust to adversarial attacks [SSY+22, MQC+22, PC22] and perturbations in the input space including data corruptions and image occlusion [NRK+21]. It has been shown that the MHSA mechanism in the vision transformer leads to a smoother loss landscape [PK21] as shown in figure 4.1. This was demonstrated by analysing the Hessian eigenvalues which represent the local curvature of a loss function. The magnitude of the Hessian eigenvalues of the vision transformer is less than in the CNN which highlights the flatter loss landscape in the vision transformer shown in figure 4.1. It is well established that flatter loss landscapes lead to more robust models [LXT+18, KMN+16, FKMN20]. However, in the low data regime, there are more negative Hessian eigenvalues in the vision transformer leading to non-convex loss landscapes which

harm optimisation. The amount of negative eigenvalues however reduces with larger amounts of training data.



(a) Loss landscape visualizations (b) Hessian max eigenvalue spectra

Figure 4.1: a. Loss landscape visualizations demonstrating the flatter landscapes in the vision transformer compared to the ResNet. b. The visualisations are in keeping with the lower maximum Hessian eigenvalues in the vision transformer. This figure is taken from [PK21].

MHSA also demonstrates exceptional robustness specifically against high-frequency noise [SSY+22, PK21] compared to convolutions. This is because MHSA and convolutions display contrasting characteristics. MHSA aggregates feature maps by ensembling, whereas convolutions differentiate them [PK21]. This leads to a reduced variance in the feature maps with an increasing number of MHSA layers. This is in contrast to convolutions which appear to increase the variance as the number of consecutive CNN layers increases. Specifically, the variance appears to increase at the end of each stage of the ResNet before pooling operations. [PK22] identified that reducing feature map uncertainty by ensembling can significantly improve the optimisation process in deep learning models.

Furthermore, a Fourier analysis of the feature maps reveals that MHSA suppresses high-frequency signals, while convolutions amplify high-frequency elements [PK21]. This means that the MHSA function acts as low-pass filter, while convolutions serve as high-pass filters. Additionally, this makes convolutions susceptible to high-frequency noise, while MHSA remains unaffected [PK21]. One can therefore deduce based on these findings, that CNNs are biased towards textural features [GRM+18] whereas MHSA is shape biased [NRK+21]. This also highlights the complementary nature of MHSA and

convolutions which can be combined and engineered to form better performing neural architectures. For example, [PK21] advises incorporating MHSA after each stage of a CNN network before sub-sampling and increasing the number of MHSA layers as one goes deeper into the network. They called this network Alter-ResNet-50 as shown in figure 4.2. This approach demonstrated superior accuracy in comparison to ResNet-50 within a limited data context (CIFAR-100) as well as enhanced robustness indicated by reducing large eigenvalues.

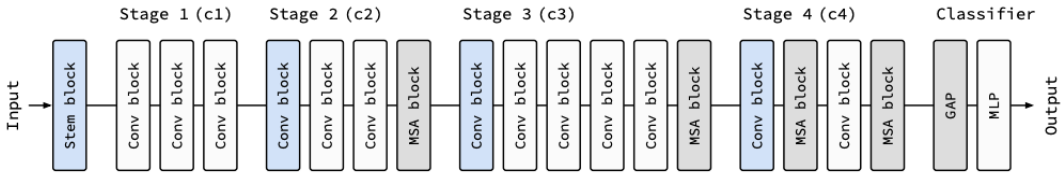


Figure 4.2: AlterNet architecture. The blue, white, and grey blocks indicate sub-sampling, conv, and MHSA block respectively. The network is a modification of the pre-activation ResNet-50 network. This figure is taken from [PK21].

In concurrent research [MJD+21] adopts an identical approach to us to address the issue of the vision transformer not generalising well to out-of-distribution, real-world data. They proposed a simple yet effective modification to the vision transformer's input layer by incorporating discrete tokens generated by a vector-quantised encoder. They demonstrate that this approach makes the vision transformer less sensitive to perturbations in the input space and also showed up to a 12% robustness improvement in terms of the relative corruption error across seven ImageNet benchmarks for four architecture variants.

4.2.3 The Vision Transformer in medical imaging

CNNs have dominated the field of deep learning in medical imaging. However, the vision transformers particularly in combination with CNNs have gained popularity in solving certain medical imaging tasks.

In order to train the vision transformer from scratch, most medical imaging tasks use a convolutional backbone to incorporate local inductive bias in the extraction of low-dimensional complex features as input into the vision transformer. At this latter stage

of the network, the vision transformer can build better globally meaningful features where a hard-coded local inductive bias is not useful. This approach has achieved good results in COVID-19 diagnosis classification and severity quantification [KPO+21]. The hybrid approach has also been applied to 3D medical imaging datasets with promising results such as in pre-clinical stage Alzheimer's disease detection [ASJ+21]. There are however notable applications of the vision transformer in medical imaging which do not use any convolutions such as in chest X-ray classification [NQ22] which relied on sophisticated pre-training and incorporated a spatial pyramid pooling module into MHSA [HTB+22].

There has recently been a large body of works which incorporated transformer layers into CNN-based segmentation networks such as the UNet and achieved improved performance in anatomical and tumour segmentation [CLY+21, HTN+22, GZM21, CWC+23, HNT+22]. These methods either incorporate transformer layers in the bottleneck of a CNN-based segmentation model such as in the TransUNet [CLY+21] or use a transformer-only encoder with down-sampling operations such as in UNETr [HTN+22] or SwinUNet [HNT+22, CWC+23]. All these networks however use a convolution-only decoder. The improved performance is likely secondary to the transformer learning better global representations as well as being more shape-biased which is useful for learning representations for segmentation [NRK+21].

4.3 Motivation

MRI acquisition and magnet strengths vary across different sites leading to textural shifts. Therefore, as MRI scanners evolve to produce images with higher quality and faster acquisition, we need deep learning models to generalise well when applied to data from new MRI scanners without retraining the model. For example, 7T scanners may be incorporated into clinical practice and therefore, we want deep learning models trained on 1.5 and 3T scanners to perform as well as when tested on the 7T scanner. Our method is aligned with methods which tackle model robustness by proposing an architectural design. We first draw inspiration from figure 4.2 to combine a convolutional backbone with the vision transformer to process downstream higher abstract

features. We secondly propose discrete representation learning for more robust image classification. Our intuition in this proposal lies in the way humans reason about the visual world through discrete or categorical representations [TT20]. This is because we hypothesise categorical representations allow us to learn robust generalisable representations by ignoring noise. For example, in MRI we as humans classify a lesion by stating whether that lesion is dark or bright on a particular sequence which is a binary representation (discrete). We ignore the shades of darkness on a particular sequence (continuous representation) which allows us to reliably classify lesions even if there were a textural shift in a new domain due to a different scanner type. Therefore, we hypothesise discrete representations of an MRI sequence will lead to more robust disease classification by a vision transformer. Specifically, we propose our method improves the robustness of disease classification on prostate MRI under an acquisition shift from a 1.5T to a 3T scanner and vice versa.

4.4 Methods

Our method is divided into two stages. The first stage aims to learn a discrete representation of both the T2-weighted and ADC images. The second stage then utilises the T2-weighted and ADC low dimensional discrete representations as input into a transformer based architecture to predict the disease class of the prostate lesion.

4.4.1 Stage 1

The first stage of our method is learning discrete representations of both the T2-weighted and ADC images. We use the VQ-VAE [VDOV+17]. In this, the discrete latent space is a categorical distribution defined as a codebook $\mathbb{D} \in \mathbb{R}^{K \times d}$ where K is the number of elements in the codebook and d is the dimensionality of each vector in the codebook. We denote the j^{th} element in $\mathbb{D}$ as l_j. Please refer to section 2.4 in the background for the quantisation process. We use equation 2.15 defined in section 2.4 as our loss function for a codebook embedded in Euclidean space.

In our ablation experiments, we use a codebook size of 1024 and note there is codebook

collapse where only 193 vectors are used during training. We also note the codebook vectors are concentrated in certain regions of Euclidean space. We show this by measuring the variance in the distance between a codebook vector and its nearest codebook vector. The mean distance between nearest neighbours in the codebook is 0.03 and the variance is 0.042 which highlights the spread of elements in the codebook is not uniformly spread after training. We therefore propose to tackle this issue by constraining the codebook to a unit hypersphere. In this method, we first initialise the codebook on the unit hypersphere and unit normalise the continuous embedding space, e. After each training update, euclidean optimisation moves the codebook vector along the tangential plane of a point on the unit hypersphere and hence moves off the surface of the hypersphere. Therefore, we can apply optimisation which is curvature aware and therefore will move points along the curved surface of a hypersphere which is also known as Riemannian optimisation [BG18]. One can also simply unit normalise each element of the codebook after each training step. We found no difference in our ablations between unit normalising the codebook and applying Riemmanian Adam [BG18]. We also wanted our codebook to be uniformly spread with maximal distance between elements in the codebook. This is to prevent clusters of elements on the hypersphere from forming and reduce overlap between codebook vectors which we want to semantically unique as possible. This is equivalent to enforcing orthogonality between all codebook elements as shown in the first term of equation 4.1. Equation 4.1 is the loss function for hyper-spherical vector quantisation which is a modification of equation 2.15 where we sample the codebook by cosine distance and add an orthogonality constraint. We just use the dot product to calculate the cosine distance because e_i and l_j have unit norm.

$$\mathcal{L}_{HypS} = \sum_{j=0}^{K}\sum_{n=0}^{K} l_j \cdot l_{j+n}^{T} + \log p(x \mid z) + \sum_{i=0}^{m} ||sg(e_i) \cdot l_k^T||_2 + \sum_{i=0}^{m} \beta ||e_i \cdot sg(l_k)^T||_2 \tag{4.1}$$

4.4.2 Stage 2

In the second stage of training, the T2 and ADC images are passed through their respective encoders before sampling their respective dictionary to form the discrete latent variables $\hat{z}_{T2} \in \mathbb{R}^{256\times16\times16\times12}$ and $\hat{z}_{ADC} \in \mathbb{R}^{256\times16\times16\times12}$ as shown in figure 4.3. $\hat{z}_{T2}$

and $\hat{z}_{ADC}$ then respectively passes through three pre-activation residual convolutional blocks with no weight sharing shown in figure 4.3. Next, the output of the pre-activation convolutional blocks are divided into m tokens similar to quantisation where $m = x \times y \times z$ to produce $\tilde{z}_{T2}$ and $\tilde{z}_{ADC}$. A T2 and ADC class token is concatenated to $\tilde{z}_{T2}$ and $\tilde{z}_{ADC}$ respectively which are then consumed by the T2 and ADC transformer respectively. The transformer is the same transformer architecture as the vision transformer [DBK+20]. However, here after each transformer layer, we apply a cross-attention layer for the transfer of semantic information between the T2 and ADC transformers.

The cross-attention mechanism we use after the transformer blocks is similar to what is proposed in the CrossVit architecture [CFP21] and highlighted in figure 4.3. Here we propose to first take the output class token from the T2 transformer which we expect to contain all the salient information representative of the T2 image and concatenate it with the tokens outputted from the ADC transformer excluding the ADC class token to form *m+1* tokens. We apply a linear projection of the T2 class token to form a single query. The keys and values and formed by linear projection of the ADC tokens. The keys and values are formed by linear projections of the ADC tokens before passing the query, keys and values through MHSA shown in equation 2.6. The process is repeated for the transfer of salient information contained in the ADC class token to the T2 tokens. Alternate layers of the transformer and cross attention layers are repeated 8 times which allows the distilling of increasingly abstract knowledge between the tokens of the T2 and ADC transformer.

Finally, the class tokens from the T2 and ADC transformers are concatenated before passing through a multi-layer perceptron (MLP) for class prediction highlighted in figure 4.3.

We only optimise the weights in the pre-activation residual convolutional blocks, transformer layers and cross-attention layers while the rest of the network is frozen in stage 2 of our framework (figure 4.3). We use the cross entropy loss function in equation 2.10 with equal weighting for each class to optimise the trainable weights in stage 2 of our model. Label smoothing is applied with an alpha value of 0.2. We also consider the case where we train the VQ-VAE jointly with the transformer component of our model by combining stages 1 and 2. We refer to this approach as a multi-task model to

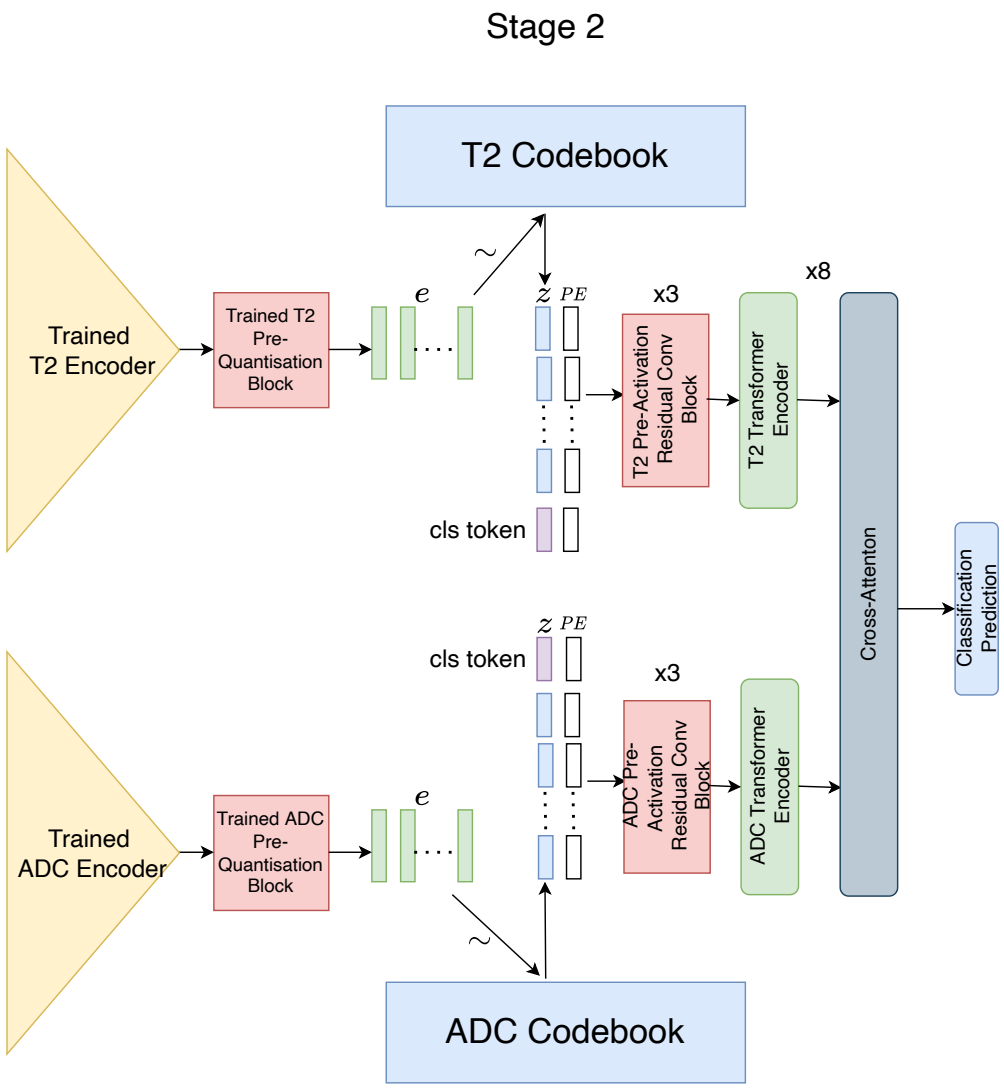


Figure 4.3: Second stage of training. We train a cross-attention architecture with input sampled for the T2 and ADC dictionary.

simultaneously reconstruct x and correctly predict the label y from $\hat{z}$.

4.4.3 Probabilistic Perspective

We can interpret our training scheme which maximises the evidence lower bound (ELBO) on the joint likelihood. First, we model the encoder and decoder of the VQ-VAE as $q_\theta(z \mid x)$ and $p_\gamma(x \mid z)$ respectively. For simplicity, we combine $\hat{z}_{ADC}$ and $\hat{z}_{T2}$ to form $\hat{z}$ in order to model the joint distribution of our model. Our transformer model is modelled with probability distribution $p_\psi(y \mid \hat{z})$. Therefore we can model the joint distribution with the factorization $p(x_{ADC}, x_{T2}, y, \hat{z}) = p_\theta(x_{T2}, x_{ADC}|\hat{z}).p_\psi(y|\hat{z})p(\hat{z})$. In equation 4.2 below, we minimise the KL divergence between the approximate posterior

$q_\theta(\hat{z} \mid x_{ADC}, x_{T2})$ and $p(\hat{z} \mid y, x_{ADC}, x_{T2})$. We assume a uniform prior for $p_\psi(y \mid \hat{z})$ and $p(z)$.

$$\mathcal{D}_{KL}(q_\theta(\hat{z} \mid x_{ADC}, x_{T2}) \parallel p(\hat{z} \mid y, x_{ADC}, x_{T2})) = -\sum_{\hat{z}} q_\theta(\hat{z}|x_{ADC}, x_{T2}) \log \frac{p(y, x_{ADC}, x_{T2}, \hat{z})}{p(y, x_{ADC}, x_{T2}) q_\theta(\hat{z}|x_{ADC}, x_{T2})} \quad (4.2)$$

The f-divergence is non-negative and therefore we derive the equation 4.4 shown below.

$$-\sum_{\hat{z}} q_\theta(\hat{z} \mid x_{ADC}, x_{T2}) \log \frac{p(y, x_{ADC}, x_{T2}, \hat{z})}{q_\theta(\hat{z} \mid x_{ADC}, x_{T2})} + \log\ p(x_{ADC}, x_{T2}, y)\Big[\sum_{\hat{z}} q_\theta(\hat{z} \mid x_{ADC}, x_{T2})\Big] \geq 0 \quad (4.3)$$

$$\log p(x_{ADC}, x_{T2}, y) \geq \sum_{\hat{z}} q_\theta(\hat{z} \mid x_{ADC}, x_{T2}) \log\ \frac{p(y, x_{ADC}, x_{T2}, \hat{z})}{q_\theta(\hat{z} \mid x_{ADC}, x_{T2})} \quad (4.4)$$

We rearrange and factorise equation 4.4 to form equation 4.5.

$$\log\ p(x_{ADC}, x_{T2}, y) \geq \mathbb{E}_{q_\theta(z|x_{T2}, x_{ADC})}\left[\log\ \frac{p_\gamma(x_{T2}, x_{ADC}|\hat{z}) p_\psi(y \mid \hat{z}) p(\hat{z})}{q_\theta(\hat{z} \mid x_{ADC}, x_{T2})}\right] \quad (4.5)$$

We can derive the ELBO shown below in equation 4.7.

$$\log\ p(x_{ADC}, x_{T2}, y) \geq \mathbb{E}_{q_\theta(z|x_{T2}, x_{ADC})} \log\ p_\gamma(x_{T2}, x_{ADC} \mid z) + \mathbb{E}_{q_\theta(z|x_{T2}, x_{ADC})}\left[\log\ \frac{p_\psi(y \mid \hat{z}) p(\hat{z})}{q_\theta(\hat{z} \mid x_{T2}, x_{ADC})}\right] \quad (4.6)$$

$$\geq \mathbb{E}_{q_\theta(z|x_{T2}, x_{ADC})} \log\ p_\gamma(x_{T2}, x_{ADC} \mid z) + D_{KL}\left(q_\theta(z \mid x_{T2}, x_{ADC}) \parallel p_\psi(y \mid \hat{z}) p(\hat{z})\right) \quad (4.7)$$

If we train the first stage and second stage together, we are maximising the ELBO with respect to the encoder, decoder and transformer parameters denoted, θ, γ and ψ respectively. In the two-stage training scheme, we are essentially bounding $p(x)$ with the ELBO where x is the ADC map or T2-weighted image. As mentioned previously, we ignore the KL divergence as it is a constant and use Jensen's inequality to derive the ELBO for stage 1 shown in equation 4.8.

$$\log p(x_{T2}, x_{ADC}) \geq \log p(x_{T2}, x_{ADC} \mid \hat{z}) p(\hat{z}) \quad (4.8)$$

In the second stage, we can rearrange equation 4.6 to construct equation 4.9. $p(y \mid z)$ is the vision transformer model expressed in the first term in equation 4.9. The second term is constant because θ and γ are fixed and we only maximise the ELBO with respect to ψ in equation 4.9.

$$\log p(x_{ADC}, x_{T2}, y) \geq \mathbb{E}_{q_\theta(z|x_{T2},x_{ADC})} \log p_\psi(y \mid z) \, \mathbb{E}_{q_\theta(z|x_{T2},x_{ADC})} \left[\log \frac{p_\gamma(x_{T2}, x_{ADC} \mid \hat{z}) p(\hat{z})}{q_\theta(\hat{z} \mid x_{ADC}, x_{T2})} \right] \quad (4.9)$$

4.5 Dataset Preparation

4.5.1 Pre-processing and Augmentation

All MRI images and their corresponding segmentations were respectively re-sampled with Cubic B-spline interpolation and Nearest-neighbour respectively to a resolution of $0.5mm \times 0.5mm \times 1.5mm$.
A patch of size of $128 \times 128 \times 12$ centred on the lesion is cropped. We normalise all images by re-scaling the intensities between 0 and 1.
We carry out various spatial transformations to augment the training dataset with a certain probability depending on the classes proportion (1 minus the class proportion). This includes vertical or horizontal flipping followed by random rotations between -90 and 90 degrees and random translations of 10mm.

4.5.2 Datasets

We combine our internal and external datasets to form a new dataset. The external dataset is made up of T2 weighted axial, diffusion-weighted imaging (b-800), ADC maps and K-trans images from the ProstateX challenge which were acquired on a 3 Tesla scanner from a single site [LDB+14] as described in chapter 3. This dataset consists of 330 pre-selected lesions with Gleason score labels.
The internal dataset consists of T2 weighted axial and ADC maps which were acquired on either 1.5 or 3 Tesla scanners as described in chapter 3. This dataset consists of 294 lesions from 205 patients and is also highly imbalanced with 89 percent of lesions

classed as clinically significant. Therefore, by mixing the two datasets, we have 624 lesions, of which 57 per cent is clinically significant. We further divide our mixed dataset into three risk groups based on the Gleason score; low risk - Gleason grade group 1 (46%), medium risk - Gleason grade group 2-3 (36%) and high risk - Gleason grade group 4-5 (18%).

In this work, we focus on training our model only on images acquired from a 1.5T scanner to test the ability of our model to classify prostate lesions well on a 3T scanner and repeat the experiment by switching the training and test set to assess domain shift from a 3T to 1.5T scanner. Therefore, we want to remove other possible causes of the reduced robustness of our model. This includes class imbalance and a prevalence shift from the training to the test set. Therefore, in this set of experiments we had to collate a new training and test set composed of our internal and external dataset to handle the severe class imbalance. This is because the external dataset consists only of images obtained from a 3T scanner and only 11% of the internal dataset is made up of clinically insignificant lesions. We therefore randomly centre crop patches of size, $128 \times 128 \times 8$ in the prostate which do not contain tumours based on the whole mount histopathology. This increased the number of patches with no/clinically insignificant tumours from 33 to 40 in the internal dataset which were scanned on a 1.5T scanner.

In the 1.5T dataset, we now have 40 lesions in each risk group all of which are derived from the internal dataset. Our 3T dataset consists of 40 randomly selected samples from each risk group acquired on a 3T scanner in the mixed internal and external dataset. In table 4.1 below, we show the proportion of internal and external data in each risk group for the 1.5T and 3T datasets.

Internal Data					
1.5T			3T		
Low Risk	Medium Risk	High Risk	Low Risk	Medium Risk	High Risk
40	40	40	0	14	20
External Data					
1.5T			3T		
Low Risk	Medium Risk	High Risk	Low Risk	Medium Risk	High Risk
0	0	0	40	26	20

Table 4.1: Train and test splits showing the number of scans from the internal and external dataset for each risk group.

We also, show the acquisition parameters for the 1.5T and 3T datasets in table 4.2. Note also the mild distribution shift in the axial resolution and slice thickness in the T2 weighted images from the 1.5T to 3T scanner.

1.5T					
T2			ADC		
Slice Thickness	Axial Resolution	Sequence	Slice Thickness	Axial Resolution	Sequence
3-4mm	0.325-0.625mm	turbo SE	3.6-4mm	2-2.5mm	single-shot EP
3T					
T2			ADC		
Slice Thickness	Axial Resolution	Sequence	Slice Thickness	Axial Resolution	Sequence
3-3.6mm	0.325-0.5mm	turbo SE	3.6-4mm	2-2.5mm	single-shot EP

Table 4.2: T2 and diffusion-weighted image acquisition parameters for the 1.5T and 3T datasets. ADC maps in both datasets are calculated with b-values; 50, 400 and 800.

We design a new experiment where there is no acquisition shift to act as our control. In the mixed dataset, we only have 40 scans in the low-risk group acquired on a 1.5T scanner which makes it difficult to design an experiment to assess our models where there is no acquisition shift for 1.5T scans as shown in table 4.1. The smallest risk group acquired in the mixed dataset on a 3T scanner is the high-risk group consisting of 69 lesions. Our control experiment therefore consists of training on 40 randomly selected samples from each risk group also acquired on a 3T scanner and testing on 29 randomly selected samples from each risk group acquired on a 3T scanner. We show the acquisition protocol of T2 weighted and diffusion-weighted images in the training and test set for the control experiment in table 4.3.

Training Dataset					
T2			ADC		
Slice Thickness	Axial Resolution	Sequence	Slice Thickness	Axial Resolution	Sequence
3.6-4mm	0.325-0.5mm	turbo SE	3.6-4mm	2-2.5mm	single-shot EP
Test Dataset					
T2			ADC		
Slice Thickness	Axial Resolution	Sequence	Slice Thickness	Axial Resolution	Sequence
3-3.6mm	0.325-0.5mm	turbo SE	3.6-4mm	2-2.5mm	single-shot EP

Table 4.3: T2 and diffusion-weighted image acquisition parameters for the training and test datasets for the control experiments. ADC maps in both datasets are calculated with b-values; 50, 400 and 800.

4.6 Experimental setup

4.6.1 Model

We use a hybrid 2D/3D VQ-VAE in order to handle the anisotropic nature of prostate MRI images. In every layer of the encoder and decoder, we use pre-activation convolutional blocks consisting of leaky ReLU activation and group normalisation (2 groups). Our VQ-VAE consists of 5 levels with the architecture shown below in table 4.4. The pre-quantisation and post-quantisation blocks are levels 5 and 1 of the encoder and decoder respectively. We explore different codebook sizes containing 1024, 512, 256 and 128 codebook vectors in our experiments.

	Encoder				Decoder			
Layer	**Convolutions**		**Downsampling/ Upsampling**	**Output ($c \times x \times y \times z$)**	**Convolutions**		**Downsampling/ Upsampling**	**Output ($c \times x \times y \times z$)**
Conv Block 1	$\begin{bmatrix} 3\times3\times1, 32 \\ 3\times3\times1, 32 \end{bmatrix}$	x 2	2D Max-Pooling	$32 \times 64 \times 64 \times 12$	$\begin{bmatrix} 3\times3\times3, 32 \\ 3\times3\times3, 256 \end{bmatrix}$	x 2	Bi-linear	$256 \times 32 \times 32 \times 12$
Conv Block 2	$\begin{bmatrix} 3\times3\times1, 64 \\ 3\times3\times1, 64 \end{bmatrix}$	x 2	2D Max-Pooling	$64 \times 32 \times 32 \times 12$	$\begin{bmatrix} 3\times3\times1, 128 \\ 3\times3\times1, 128 \end{bmatrix}$	x 2	Bi-linear	$64 \times 64 \times 64 \times 12$
Conv Block 3	$\begin{bmatrix} 3\times3\times1, 128 \\ 3\times3\times1, 128 \end{bmatrix}$	x 2	2D Max-Pooling	$128 \times 16 \times 16 \times 12$	$\begin{bmatrix} 3\times3\times1, 64 \\ 3\times3\times1, 64 \end{bmatrix}$	x 2	Bi-linear	$32 \times 256 \times 256 \times 12$
Conv Block 4	$\begin{bmatrix} 3\times3\times3, 256 \\ 3\times3\times3, 256 \end{bmatrix}$	x 2	None	$128 \times 16 \times 16 \times 12$	$\begin{bmatrix} 3\times3\times1, 32 \\ 3\times3\times1, 1 \end{bmatrix}$	x 2	None	$1 \times 256 \times 256 \times 12$

Table 4.4: VQ-VAE architecture.

The transformer is the same architecture as the vision transformer [DBK+20] consisting of 8 layers and 8 heads in MHSA. The MLP in our transformers model has an input of 1×256 with an expansion ratio of 2. We also use 8 heads in cross-attention.

4.6.2 Model Ablations

In the first set of experiments, we compare different versions of our method. We compare the hyper-spherical codebook 2-stage version of our model with an Euclidean codebook 2-stage model where the codebooks are embedded in Euclidean or hyper-spherical space. We also compare our model trained in a single stage where we combine stages 1 and 2. This is carried out for both the Euclidean and hyper-spherical codebook and due to the multi-task nature of this approach, we refer to these two models as 'Euclidean Codebook Multi-task Discrete' and 'Hyper-spherical Codebook Multi-task

Discrete'. We finally compare to a continuous version of our model where stage 1 and 2 is combined to form a multi-task model. We refer to this model as the 'Multi-task Continuous' model of which we explore both Euclidean and hyper-spherical representations. We compare these models in the setting of single domain generalisation where the training and test sets are from different magnet strength scanners shown in table 4.1.

4.6.3 Single Domain Generalisation

In the next set of experiments, we compare our method to the 3D ResNet-50, vision transformer and hybrid vision transformer aided with domain generalisation methods under an acquisition shift. We choose methods which focus on aggressive data augmentation, adversarial learning and self-supervised learning to build more robust representations. Specifically, the SDG methods used are BigAug [ZWY+20], ProstAdv [HZG+22] and Jigen [CDB+19]. ProstAdv is an adversarial technique which uses the decoupling direction and norm (DDN) method [RHO+19]. DDN produces gradient-oriented Adversarial examples that provoke misclassification with minimal L2 norm variations by decoupling the direction and adding adversarial perturbation to the image. The self-supervised method Jigen [CDB+19] is applied to the training set and used to initialise the weights of the classification model.

The vision transformer [DBK+20] consists of 12 layers and 8 heads. In the vision transformer, images are divided into patches of $8 \times 8 \times 2$ which are flattened before applying positional encodings.

The hybrid vision transformer consists of a modified ResNet-26 followed by a vision transformer made up of 12 layers and 8 heads. The exact architecture is shown in table 4.5 where the first convolutional block is a single convolutional layer and the blocks thereafter are pre-activation residual blocks. We show the convolutional kernel size and output in each block in table 4.5. 2D max pooling is applied after convolutional blocks 2 and 3. 3D max pooling is applied after convolutional block 4. We use grouped convolutions (2) for every convolutional layer throughout the network in order to divide the feature maps into T2-weighted and ADC image-specific feature maps. In

the single-scale hybrid vision transformer, the output of the last convolutional block are features of dimension, $256 \times 4 \times 4 \times 4$. The output is divided into 128 patches of size $128 \times 1 \times 1 \times 1$. We flatten the feature maps and apply positional encoding before input into the vision transformer. A class token of size, 1×128 is concatenated with the rest of the other feature maps to construct an embedding matrix of size, 257×128. The T2 weighted images and ADC maps are concatenated along the channel dimension for input into the model for all three models we compare to.

Layer	**Convolutions**	**Output** $(c \times x \times y \times z)$
Conv Block 1	$7 \times 7 \times 1$, Stride $= 2 \times 2 \times 1$, 32	$32 \times 64 \times 64 \times 8$
Conv Block 2	$\begin{bmatrix} 3 \times 3 \times 1, 32 \\ 3 \times 3 \times 1, 32 \end{bmatrix}$ x 2	$32 \times 32 \times 32 \times 8$
Conv Block 3	$\begin{bmatrix} 3 \times 3 \times 1, 64 \\ 3 \times 3 \times 1, 64 \end{bmatrix}$ x 2	$64 \times 16 \times 16 \times 8$
Conv Block 4	$\begin{bmatrix} 3 \times 3 \times 1, 128 \\ 3 \times 3 \times 1, 128 \end{bmatrix}$ x 2	$128 \times 8 \times 8 \times 8$
Conv Block 5	$\begin{bmatrix} 3 \times 3 \times 3, 256 \\ 3 \times 3 \times 3, 256 \end{bmatrix}$ x 2	$256 \times 4 \times 4 \times 4$
Vision Transformer		1×128
Fully Connected	FC(256, Number of Output Classes)	Number of Output Classes

Table 4.5: Hybrid Vision Transformer architecture.

4.6.4 Perturbation Experiments

In the perturbation experiments, we have the same experimental setup and training/test dataset as for the no-acquisition shift experiments where the training/test dataset is acquired from the same 3T scanner. However, in this experiment, we apply various perturbations to the test set. Here, we compare our method to the multi-task version of our method with continuous Euclidean representations.

We adjust noise levels between 1 and 30 % ((1, 5, 10, 15, 20, 25, 30 %)) for Gaussian, Poisson and Salt and Pepper (S&P) noise. Gaussian Blur is incorporated with a Gaussian kernel which has a window size of 7×7 and variance ranging from 0.1 to 2.0 (0.1, 1.0, 2.0). Random motion blur is applied by using the TorchIO deep learning library [PGSO21].

4.6.5 Training and Evaluation

The weights of the ResNet-50 are initialised with Kaiming initialisation [HZRS15] and trained for 100 epochs using Adam optimisation (weight decay = 0.01) with a base learning rate of 0.001 [KB14]. We train the vision transformer and hybrid transformer model from scratch using AdamW optimisation (weight decay = 0.05) with a cosine annealing learning rate scheduler (learning rate = 0.001) for 200 epochs with 10 warm-up epochs.

The convolutional weights in the VQ-VAE are initialised with Kaiming initialisation [HZRS15]. We use two different codebook initialisations for the Euclidean and hyper-spherical codebook as described above in section 4.4. The VQ-VAE is trained for 200 epochs using Adam optimisation (weight decay = 0.01) with a base learning rate of 0.0005 [KB14].

The transformer-based model in stage 2 is trained using AdamW optimisation (weight decay = 0.05) with a cosine annealing learning rate scheduler (learning rate = 0.0001) for 200 epochs with 20 warm-up epochs. We initialise weights in stage 2 of the model with truncated normal initialisation. We apply the same training scheme in our multi-task model where stage 1 and stage 2 are combined and trained together. The loss functions of stage 1 and stage 2 are combined to train our multi-task model.

Results are evaluated with the accuracy, specificity, precision, recall and AUC. We calculate the specificity, precision, recall and AUC for each class in this 3-class classification problem as a binary classification problem such that we calculate the scores for the group of interest against the other two groups combined. For example, the specificity, precision, recall and AUC for the low-risk group are defined as the low-risk group vs (medium-risk group + high-risk group).

We also calculate the relative corruption error in terms of the AUC compared to other baseline models. We use equation 2.18 in the background chapter (CHAPTER 2) to calculate the relative AUC error.

4.7 Results

4.7.1 Ablations

We first carry out ablation studies to find the optimal number of codebook vectors required to build a robust representation for an acquisition shift from a 1.5T to 3T scanner. We use Euclidean representations and assume an equal number of codebook vectors needed to reconstruct T2-weighted and ADC images. We train our model in two stages for these experiments.

Codebook Size	Accuracy	Specificity	Precision	Recall	AUC
No Acquisition Shift					
128	0.771 ± 0.079	0.768 ± 0.092	0.780 ± 0.071	0.759 ± 0.077	0.768 ± 0.063
256	**0.777 ± 0.097**	0.766 ± 0.080	0.785 ± 0.066	0.760 ± 0.070	0.778 ± 0.076
512	0.775 ± 0.081	0.760 ± 0.052	0.782 ± 0.072	**0.771 ± 0.073**	0.771 ± 0.083
1024	0.773 ± 0.073	**0.771 ± 0.063**	**0.789 ± 0.055**	0.770 ± 0.073	**0.772 ± 0.077**
Acquisition Shift					
128	0.703 ± 0.084	0.689 ± 0.083	0.696 ± 0.088	0.719 ± 0.088	0.708 ± 0.088
256	**0.724 ± 0.083**	**0.718 ± 0.078**	**0.720 ± 0.066**	**0.733 ± 0.059**	**0.728 ± 0.047**
512	0.718 ± 0.074	0.700 ± 0.039	0.716 ± 0.088	0.723 ± 0.080	0.703 ± 0.063
1024	0.709 ± 0.055	0.699 ± 0.049	0.707 ± 0.063	0.697 ± 0.092	0.700 ± 0.095

Table 4.6: Comparison of model performance on the test set using 5 evaluation metrics ± standard deviation for different Euclidean codebook sizes. The metrics were averaged across the three classes.

We find the optimal number of codebook vectors was 256 in the acquisition shift experiments shown in table 4.6 with an AUC of 0.728. In the experiments where there were no acquisition shifts, the ablation with the largest number of codebook vectors (1024) achieved the highest AUC score (0.772). In future experiments we will therefore now use 256 codebook vectors.

In our next set of ablations, we compare our 2-stage approach with the multi-task version of our method. We also compare our method using an Euclidean codebook to an alternate variation of our approach which uses a hyper-spherical codebook. Finally, we compare our approach to a continuous version of our model where the codebook is removed as described in the methods section (section 4.4. This creates 6 different versions of our model with the performance of each model shown in table 4.7 which is also visualised in the bar chart in figure 4.4. We carry out these experiments under both acquisition shifts (1.5T to 3T and vice versa) and no acquisition shift (control experiments). We average the metric scores across the three classes.

	Accuracy	Specificity	Precision	Recall	AUC
No Acquisition Shift					
Euclidean Codebook 2-stage	0.777 ± 0.097	0.766 ± 0.080	**0.785 ± 0.066**	0.760 ± 0.070	0.778 ± 0.076
Hyper-spherical Codebook 2-stage	0.781 ± 0.072	0.761 ± 0.062	0.779 ± 0.054	0.771 ± 0.081	0.775 ± 0.088
Euclidean Codebook Multi-task Discrete	0.782 ± 0.089	**0.769 ± 0.059**	0.781 ± 0.083	0.771 ± 0.070	0.777 ± 0.069
Euclidean Multi-task Continuous	**0.784 ± 0.066**	0.765 ± 0.048	0.783 ± 0.050	0.775 ± 0.071	**0.779 ± 0.063**
Hyper-spherical Codebook Multi-task Discrete	0.764 ± 0.071	0.743 ± 0.064	0.757 ± 0.059	0.768 ± 0.079	0.760 ± 0.057
Hyper-spherical Multi-task Continuous	0.770 ± 0.084	0.763 ± 0.073	0.771 ± 0.079	**0.776 ± 0.088**	0.774 ± 0.090
Acquisition Shift					
Euclidean Codebook 2-stage	0.724 ± 0.083	0.718 ± 0.078	0.720 ± 0.066	0.733 ± 0.059	0.728 ± 0.047
Hyper-spherical Codebook 2-stage	**0.731 ± 0.028**	**0.720 ± 0.028**	**0.727 ± 0.046**	**0.736 ± 0.049**	**0.739 ± 0.041**
Euclidean Codebook Multi-task Discrete	0.721 ± 0.065	0.718 ± 0.044	0.721 ± 0.085	0.730 ± 0.071	0.722 ± 0.061
Euclidean Multi-task Continuous	0.700 ± 0.083	0.691 ± 0.056	0.703 ± 0.077	0.708 ± 0.059	0.708 ± 0.044
Hyper-spherical Codebook Multi-task Discrete	0.728 ± 0.044	0.711 ± 0.051	0.729 ± 0.049	0.736 ± 0.070	0.729 ± 0.061
Hyper-spherical Multi-task Continuous	0.695 ± 0.056	0.688 ± 0.062	0.701 ± 0.069	0.684 ± 0.074	0.691 ± 0.066

Table 4.7: Comparison of model performance on the test set using 5 evaluation metrics ± standard deviation for 6 different variations of our model. The results for no acquisition shift where the training and test dataset are both acquired on a 3T scanner. In the acquisition shift results, we show the results averaged for 1.5T to 3T and 3T to 1.5T domain shift experiments. The metrics were averaged across the three classes. The metrics were averaged across the three classes.

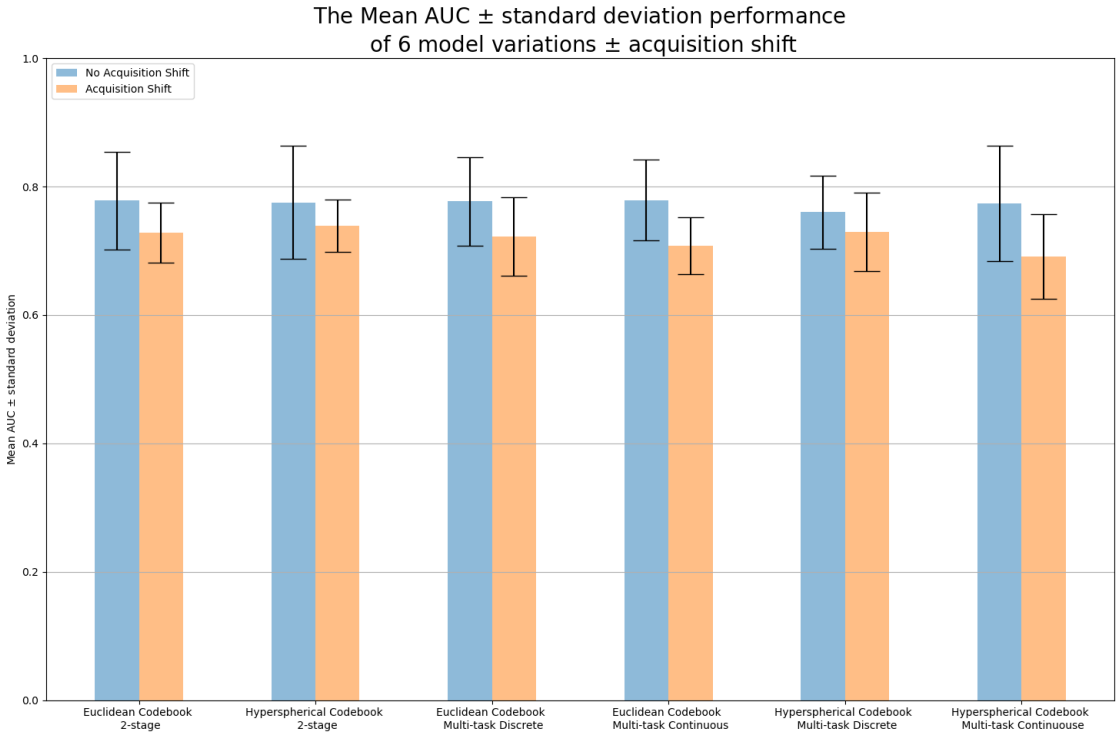


Figure 4.4: Bar Chart showing the AUC performance ± acquisition shift for 6 model types.

We find that under no acquisition shift, there appears to be no benefit from using our 2-stage approach or embedding the codebook on the hypersphere as shown in table 4.7. However, we note that our 2-step method which uses either an Euclidean or hyper-spherical codebook, improved model performance under an acquisition shift. Note the AUC improves from 0.722 to 0.728 by carrying out our methods in 2 steps

compared to our multi-task approach. Additionally, the AUC score improved by 0.01 after embedding the codebook on a hypersphere in the 2-stage training scheme (table 4.7. Finally, we highlight in table 4.7, that while the continuous multi-task version of our model performs favourably when the train and test set are from the same scanner, performance drops significantly compared to our discrete model versions under an acquisition shift.

4.7.2 Single Domain Generalisation

In this set of experiments, we compare the best version of our method in our ablation experiments for acquisition shift which was the 2-stage hyper-spherical codebook to BigAug [ZWY+20], AdvProst [HZG+22] and Jigen [CDB+19] used to improve the domain generalisability of deep learning models under an acquisition shift.

Discrete Representations					
	Accuracy	**Specificity**	**Precision**	**Recall**	**AUC**
Hyper-spherical Codebook 2-stage	**0.731 ± 0.028**	**0.720 ± 0.028**	**0.727 ± 0.046**	**0.736 ± 0.049**	**0.739 ± 0.041**
BigAug					
ResNet-50	0.720 ± 0.073	0.701 ± 0.078	0.719 ± 0.055	0.724 ± 0.047	0.724 ± 0.053
Hybrid vision transformer	0.726 ± 0.059	0.726 ± 0.033	0.720 ± 0.059	0.730 ± 0.047	0.731 ± 0.060
3D Vision transformer	0.646 ± 0.083	0.663 ± 0.092	0.622 ± 0.082	0.648 ± 0.090	0.641 ± 0.087
ProstAdv					
ResNet-50	0.717 ± 0.066	0.708 ± 0.069	0.721 ± 0.068	0.729 ± 0.070	0.730 ± 0.057
Hybrid vision transformer	0.722 ± 0.049	0.711 ± 0.045	0.726 ± 0.061	0.729 ± 0.047	0.729 ± 0.062
Vision transformer	0.620 ± 0.064	0.631 ± 0.087	0.618 ± 0.077	0.633 ± 0.083	0.625 ± 0.082
Jigen					
ResNet-50	0.690 ± 0.052	0.678 ± 0.073	0.691 ± 0.088	0.696 ± 0.070	0.699 ± 0.062
Hybrid vision transformer	0.701 ± 0.079	0.683 ± 0.099	0.702 ± 0.089	0.695 ± 0.082	0.704 ± 0.079
Vision transformer	0.600 ± 0.103	0.595 ± 0.117	0.606 ± 0.092	0.610 ± 0.096	0.608 ± 0.094

Table 4.8: Comparison of 3 different model architectures with 3 different SDG methods with our discrete representation approach. We evaluate performance with 5 evaluation metrics ± standard deviation. We show the results averaged for the 1.5T to 3T and 3T to 1.5T domain shift experiments. The metrics were averaged across the three classes.

The results in the table demonstrate 4.8 that under an acquisition shift our hyper-spherical codebook 2-stage approach outperforms all 3 of the different domain generalisation methods (augmentation, adversarial and self-supervised methods) applied to a convolutional architecture (ResNet-50), a hybrid convolutional/transformer model (Hybrid 3D vision transformer) and a transformer only model (3D vision transformer) for all evaluation metrics. For example, the AUC score for our method is 0.739 compared to the AUC score of 0.731 obtained by an aggressive augmentation-based method applied to our hybrid model which was the second-best method. We further demon-

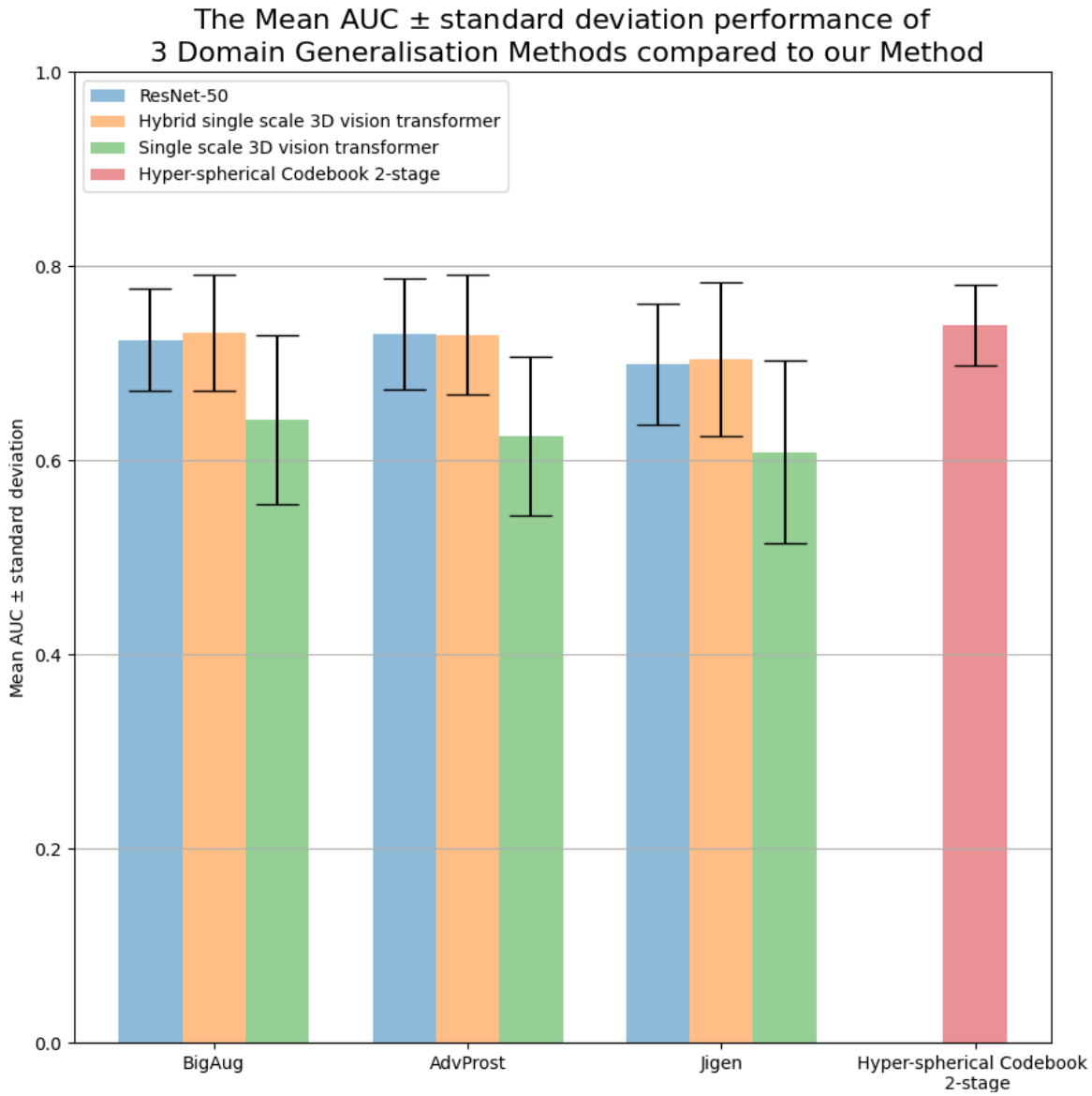


Figure 4.5: Bar Chart comparing AUC performance of 3 domain generalisation applied 3 different models compared to our model without any domain generalisation method applied.

strate the mean AUC scores and standard deviation obtained by different methods in a bar chart shown in figure 4.5.

We next analyse the 3 different model architectures with 3 different SDG methods compared with our discrete representation approach for each risk group shown in table 4.9 using the AUC scores.

Table 4.9 again shows that our method which simply uses discrete representation on a hypersphere outperforms the ResNet-50, Hybrid vision transformer and vision

Discrete Representations			
	Low Risk	**Medium Risk**	**High Risk**
Hyper-spherical Codebook 2-stage	**0.736 ± 0.036**	**0.740 ± 0.044**	**0.740 ± 0.041**
BigAug[ZWY+20]			
ResNet-50	0.723 ± 0.053	0.725 ± 0.056	0.724 ± 0.050
Hybrid vision transformer	0.729 ± 0.057	0.731 ± 0.058	0.734 ± 0.065
Vision transformer	0.640 ± 0.087	0.640 ± 0.090	0.644 ± 0.084
ProstAdv[HZG+22]			
ResNet-50	0.729 ± 0.060	0.731 ± 0.061	0.730 ± 0.050
Hybrid vision transformer	0.725 ± 0.055	0.730 ± 0.066	0.732 ± 0.065
Vision transformer	0.624 ± 0.081	0.627 ± 0.084	0.624 ± 0.081
Jigen[CDB+19]			
ResNet-50	0.687 ± 0.066	0.702 ± 0.060	0.708 ± 0.068
Hybrid vision transformer	0.702 ± 0.071	0.708 ± 0.082	0.702 ± 0.084
Vision transformer	0.604 ± 0.095	0.609 ± 0.096	0.611 ± 0.091

Table 4.9: AUC scores ± standard deviation of 3 different model architectures with 3 different SDG methods with our discrete representation approach. We show the results averaged for 1.5T to 3T and 3T to 1.5T domain shift experiments.

transformer in all 3 risk groups in terms of the AUC even when aided with 3 SDG methods. We note similar trends in the AUC scores for each risk group as for the AUC scores averaged across all risk groups in table 4.8.

4.7.3 Perturbation Experiments

	Baseline	**Gauss**	**Poisson**	**S&P**	**Blur**	**Motion**
	Low Risk Group					
ResNet-50	0.718 ± 0.074	0.665 ± 0.101	0.689 ± 0.092	0.680 ± 0.076	0.698 ± 0.086	0.664 ± 0.113
Hybrid Vision transformer	0.750 ± 0.046	0.708 ± 0.090	0.708 ± 0.083	0.698 ± 0.090	0.714 ± 0.084	0.685 ± 0.097
Vision transformer	0.666 ± 0.086	0.616 ± 0.068	0.632 ± 0.064	0.616 ± 0.090	0.630 ± 0.074	0.624 ± 0.074
Euclidean Multi-Task Continuous	**0.780 ± 0.061**	0.728 ± 0.059	0.720 ± 0.081	0.727 ± 0.083	0.729 ± 0.063	0.711 ± 0.083
Hyper-spherical Codebook 2-stage Discrete	0.777 ± 0.068	**0.764 ± 0.061**	**0.758 ± 0.083**	**0.754 ± 0.052**	**0.770 ± 0.071**	**0.740 ± 0.084**
	Medium Risk Group					
ResNet-50	0.732 ± 0.084	0.680 ± 0.114	0.694 ± 0.113	0.696 ± 0.083	0.699 ± 0.090	0.681 ± 0.113
Hybrid Vision transformer	0.773 ± 0.054	0.724 ± 0.090	0.706 ± 0.114	0.710 ± 0.084	0.727 ± 0.098	0.695 ± 0.104
Vision transformer	0.670 ± 0.083	0.620 ± 0.075	0.636 ± 0.054	0.621 ± 0.078	0.641 ± 0.065	0.609 ± 0.079
Euclidean Multi-Task Continuous	**0.784 ± 0.063**	0.735 ± 0.094	0.724 ± 0.081	0.736 ± 0.078	0.732 ± 0.064	0.721 ± 0.099
Hyper-spherical Codebook 2-stage Discrete	0.781 ± 0.075	**0.770 ± 0.061**	**0.761 ± 0.084**	**0.759 ± 0.054**	**0.768 ± 0.074**	**0.742 ± 0.085**
	High Risk Group					
ResNet-50	0.740 ± 0.090	0.694 ± 0.118	0.704 ± 0.114	0.708 ± 0.080	0.710 ± 0.093	0.688 ± 0.124
Hybrid Vision transformer	0.781 ± 0.068	0.729 ± 0.094	0.721 ± 0.118	0.728 ± 0.093	0.741 ± 0.134	0.694 ± 0.116
Vision transformer	0.671 ± 0.093	0.636 ± 0.071	0.633 ± 0.051	0.628 ± 0.081	0.646 ± 0.064	0.609 ± 0.083
Euclidean Multi-Task Continuous	**0.787 ± 0.061**	0.732 ± 0.091	0.734 ± 0.086	0.742 ± 0.089	0.747 ± 0.064	0.723 ± 0.094
Hyper-spherical Codebook 2-stage Discrete	0.785 ± 0.071	**0.770 ± 0.069**	**0.762 ± 0.081**	**0.759 ± 0.054**	**0.775 ± 0.077**	**0.747 ± 0.084**

Table 4.10: The mean AUC ± standard deviation for each risk group comparing 4 model types to our model under various perturbations in the input space. The baseline refers to no perturbations applied. The results shown are averaged across all perturbation parameters i.e. overall noise levels for Gaussian, Poisson and Salt and Pepper (S&P) noise.

The results in table 4.10 demonstrate that by using the hyper-spherical codebook 2-stage version of our method, the AUC score diminishes far less under various textural

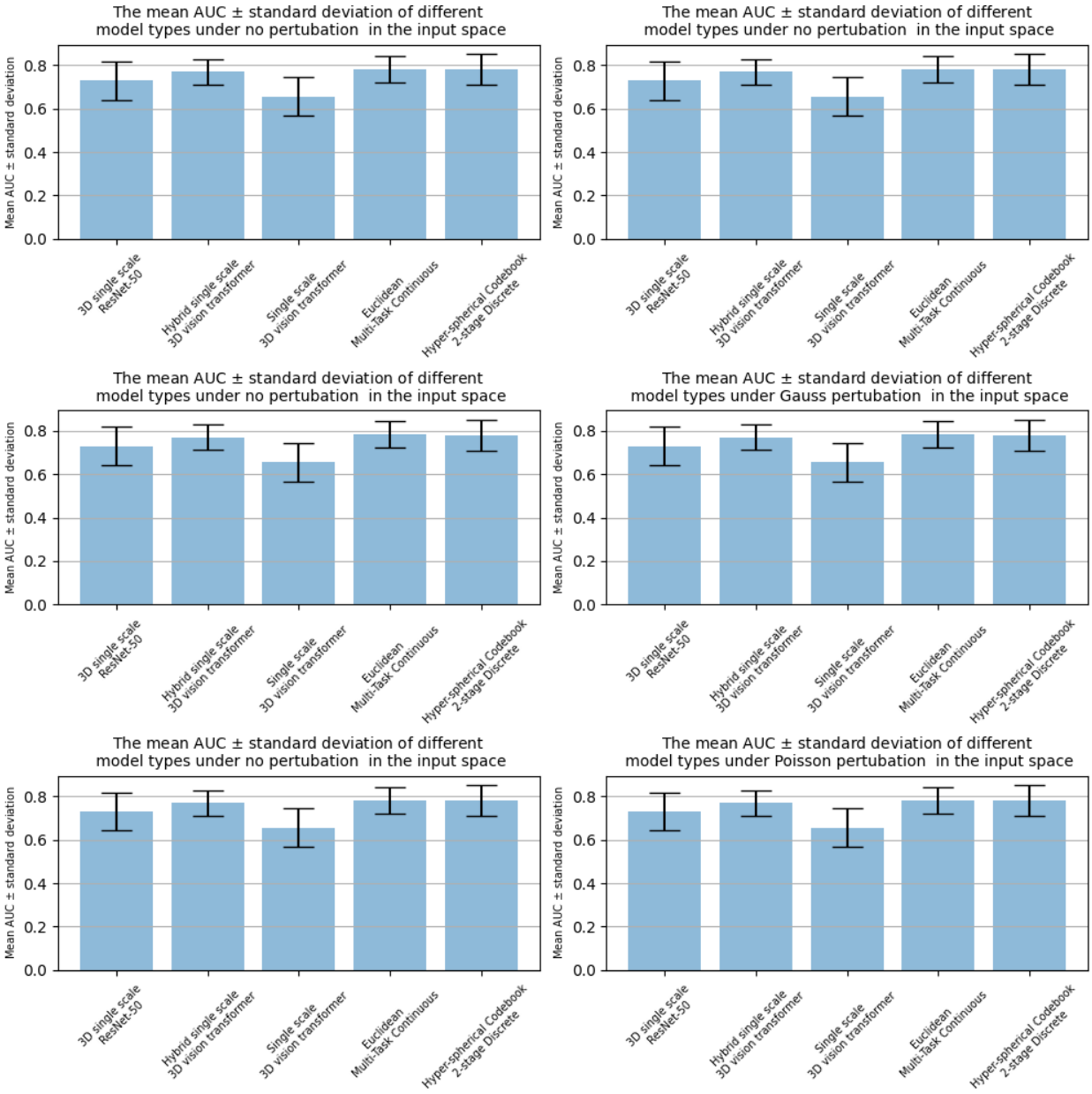


Figure 4.6: Bar Chart showing the AUC performance under various perturbations for 5 model types.

and spatial perturbations compared to the other models. This is true for all risk groups as highlighted in bold in table 4.10. For example, the AUC score for our approach only diminishes by 1.2 points on average across all risk groups under Gaussian noise averaged across all noise levels compared to by 5.1 points in the Euclidean multi-task continuous model. However, while our method compares favourably under various perturbations, the Euclidean multi-task continuous model produced the highest AUC scores for all risk groups with no perturbations (baseline). We visually demonstrate the mean AUC score and standard deviation averaged across all risk groups under various

perturbations in a bar chart shown in Figure 4.6.

	Low Risk	Medium Risk	High Risk
ResNet-50	51.0	50.0	57.1
Hybrid Vision transformer	41.8	34.6	38.3
Vision transformer	46.7	47.1	55.1
Euclidean Multi-Task Continuous	34.7	38.6	43.6

Table 4.11: The relative corruption error (%) based on AUC of our hyper-spherical codebook 2-stage discrete model relative to the 4 other models in table 4.10.

Finally in table 4.11, we show that the relative corruption error in terms of the AUC of our hyper-spherical codebook 2-stage discrete model relative to the 4 other models in table 4.10 are significantly less than 1 for all 3 risks groups. This shows our model demonstrates superior robustness performance average across all types of input perturbations compared to the 4 model architectures in table 4.11 for all risk groups. Significantly, we show the relative corruption error of the Euclidean continuous version of our model relative to the hyper-spherical codebook 2-stage model to be 0.347, 0.386 and 0.436 for the low, medium and high-risk groups respectively. This demonstrates the benefit of using discrete representations uniformly distributed on a hyper-sphere for robust classification with a vision transformer.

4.8 Conclusion

In conclusion, we show that we can improve the robustness of the vision transformer under an acquisition shift from 1.5T to 3T and vice versa by using a discrete input obtained from the vector quantised high abstraction CNN features in a hyper-spherical latent space. We apply our method in the task of prostate disease classification and outperform various CNN, transformer and hybrid-based models. We also show our approach competes well with augmentation-based, adversarial and self-supervised methods used for domain generalisation.

In future work, we will explore other manifold constraints such as the hyperbolic manifold to embed our discrete latent space. Hence, we will investigate whether curvature plays an important role in building robust discrete representations for classification.

Chapter 5

Vector Quantisation for Robust Medical Segmentation

This chapter is adapted from the following publication.

- **Santhirasekaram, A.**, Kori, A., Winkler, M., Rockall, A. and Glocker, B., 2022, September. Vector Quantisation for Robust Segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 663-672). Cham: Springer Nature Switzerland.

5.1 Introduction

This chapter first introduces the notion of discrete representation learning for robust segmentation. We propose and justify that learning a discrete representation in a low-dimensional embedding space improves the robustness of a segmentation model. This is achieved with a dictionary learning method called vector quantisation as in the previous chapter. We use a set of experiments designed to analyse robustness in both the latent and output space under a domain shift and noise perturbations in the input space. We adapt the popular UNet architecture, inserting a quantisation block in the bottleneck. We demonstrate improved segmentation accuracy and better robustness on three segmentation tasks.

Segmentation of medical images provides value in both aiding diagnosis and treatment planning [LWW+20]. Deep learning, notably CNNs has significantly improved segmentation performance and is now the most widely used method for automated segmentation [LWW+20]. However, it is well established in the literature that these models are not significantly robust to perturbations in the input space whether that be noise or a domain shift [CW17, MDFFF17]. This is particularly relevant in the medical domain whereby images are acquired from many sources with varying protocols and hence demonstrate different noise characteristics [Che21].

We note that current methods have not explored how to improve the design of a segmentation model so that it is inherently more robust to input perturbations [Che21]. We take inspiration from the VQ-VAE and hypothesise that deep segmentation models are more robust and accurate when mapping the input data to a discrete latent space. We focus on anatomical segmentation which benefits most from a finite latent space, because the spatial variability of human anatomy is limited.

5.2 Methods

5.2.1 Robustness and Network Assumptions

Given an input x, we first define a function $f(x)$ to represent the transformed input due to perturbation. This is a generic function in order to account for various types of perturbations ranging from a re-normalisation function to a non-linear mapping. We therefore now denote the perturbation to be $\delta(x) = f(x) - x$ which can represent noise or a domain shift. This work aims to find a way to learn a model Φ with weights w to be robust against $\delta(x)$ and construct an uncorrupted segmentation y from the perturbed input $f(x)$.

Assumption 1. *Assuming a small value for $\delta(x)$, we can then approximate $\Phi(x + \delta(x))$ with a first order Taylor expansion as follows: $\Phi(x + \delta(x)) = \Phi(x) + \delta(x)^T \nabla_x \Phi$. Therefore, the training framework should optimize for $argmin_w[\Phi(x + \delta(x)) - \Phi(x)]$ to be robust.*

Assumption 2. *In this work we assume that the segmentation network can be decomposed into an encoder Φ_e and decoder Φ_d such that $\Phi = \Phi_d \circ \Phi_e$, where $\Phi_e : \mathcal{X} \to \mathcal{E}$ maps from image*

space to a lower dimensional embedding space and $\Phi_d : \mathcal{E} \rightarrow \mathcal{Y}$ *maps the embedding space back to segmentation space.* Φ_d *can also be a function of the embedding and the outputs from each scale of the encoder denoted,* $\mathcal{S}$ *such that* $\Phi_d : \mathcal{E}, \mathcal{S} \rightarrow \mathcal{Y}$.

5.2.2 Quantisation for Robustness

We define a quantisation block inserted into the bottleneck of a segmentation network to provide a means of robustness to perturbations in the input space. Formally, with the quantisation block, our segmentation network Φ now decomposes as $\Phi_d \circ \Phi_q \circ \Phi_e$, where Φ_e, Φ_d, Φ_q corresponds to the encoder, decoder, and quantisation blocks. Φ_q maps the embedding vectors (e) from the continuous embedding space output of $\Phi_e(x)$ to quantised vectors (z). The goal of the quantisation block is to remove unnecessary information in the latent space by collapsing a continuous latent space into a set of discrete vectors.

The quantisation process initially requires us to define a codebook $c \in \mathbb{R}^{KxD}$. K is the size of the codebook and D is the dimensionality of each codebook vector $l_i \in \mathbb{R}^{\mathbb{D}}$. We then define a discrete uniform prior and map each embedding vector $e_j \in e$ to the nearest codebook vector denoted, l_k which forms $z_j \in z$ as follows [VDOV+17]:

$$\mathbb{P}(z_j = l_i \mid x) = \left\{ \begin{array}{ll} 1 & \text{if} \quad i = \arg\min_i \|\Phi_e(x)_j - l_i\|_2 \\ 0 & \text{otherwise} \end{array} \right\} \tag{5.1}$$

In other words, the above equation essentially states $\forall z_j \in z, z_j = l_k$, where $k = \arg\min_i \|\Phi_e(x)_j - l_i\|_2$.

Similar to the VQ-VAE described in chapter 4, backpropagation through the non-differentiable quantisation block requires straight-through gradient approximation whereby one copies the gradients from z to the encoder output (e) which is used to update the codebook. This allows the entire model to be trained end-to-end with the following loss function[VDOV+17]:

$$\mathcal{L}_{total} = \mathcal{L}_{Dice}(\hat{y}, y) + \mathcal{L}_{CE}(\hat{y}, y) + \|sg(\Phi_e(x)) - l\|_2 + \beta\|\Phi_e(x) - sg(l)\|_2 \tag{5.2}$$

The first two terms in equation 5.2 is the sum of the Dice and cross-entropy loss between the predicted segmentation $\hat{y}$ and label y. Similar to equation 2.15 in chapter 2, the third and fourth constitute the quantisation loss function where the third term updates the codebook by moving the codebook vectors l_i towards the output of the encoder and the fourth term in equation 5.2 is defined as a commitment loss weighted by β [VDOV+17]. A stop gradient (sg) is applied to constrain the update to the appropriate operand as in equation 2.15.

Based on assumption 1, we get the following equation.

$$\Phi_q(\Phi_e(x+\epsilon))) = \Phi_q(\Phi_e(x) + \epsilon^T \nabla_x \Phi_e(x)) \tag{5.3}$$

We claim, quantisation pushes $\epsilon^T \nabla_w \Phi_e(x)$ to 0 and thereby enforces $\Phi_q(\Phi_e(x+\epsilon))) = \Phi_q(\Phi_e(x))$. For this claim to hold, we make the following assumption:

Assumption 3. *We assume if $\|\Phi_e(x) - l_i\|_2 > 0$; then x is absolutely perturbed by $\delta(x)$. This means a codebook c with dimensionality D contains the minimal number of codebook vectors K to fully capture all possible semantics in the latent space i.e., complete. We also assume c is uniformly distributed on the surface of a D-dimensional hypersphere. Therefore, the space on the hypersphere which lie between c represents only perturbations of c. We denote the entire surface of the hypersphere as Z and $\Phi_e(x)$ only generates e which only lies on Z.*

Finally, if the decoder Φ_d is only a function of the quantised vector representation z, then given our assumption 3, $\Phi(x + \delta(x)) = \Phi(x)$. However, if Φ_d is a function of z and the output of each scale from the encoder s like in the UNet, then the effect on the output of the model by $\delta(x)$ is only reduced. Yet, this may be beneficial in practice where the codebook is not complete. One can imagine, having the effect of learning to interpolate between a function of the quantised latent space and the perturbed output from each scale of the encoder.

5.2.3 Perturbation Bounds

Proposition 1. *Given a uniformly distributed codebook, we propose to empirically derive a limit for the shift in latent space which represents the boundary between perturbation and a*

true semantic shift for the data distribution which we sample, given assumption 3. This can be defined as the maximum perturbation allowed around a single codebook vector denoted, r and calculated empirically as the average distance between a codebook vector l_i and its nearest neighbour l_{i+1} across the whole of c as follows:

$$r = \frac{\sum_{i=0}^{i=K-1} \frac{1}{2}(\|l_i - l_{i+1}\|_2)}{K-1} \tag{5.4}$$

Calculating the variance r allows one to somewhat estimate numerically the uniformity (completeness) of the codebook and hence assess how trusty-worthy r is to represent the boundary between a perturbation and a true semantic shift. If r is not uniformly distributed then one can just calculate the distance between a codebook vector and its nearest neighbour r_i. This allows us to at least calculate the maximum latent shift allowed for that codebook vector despite it not representing the true boundary between perturbation and a semantic shift.

Lemma 1. *For simplicity observe a single vector from the output of $\Phi_e(x)$ and $\Phi_e(x + \delta(x))$ denoted, e_j and $e_j + \delta$ respectively. We can combine equation 5.4 and the first order Taylor expansion of $\Phi_e(x + \delta(x))$ to theoretically express r in terms of $\delta(x)$ as follows:*

$$r > \|\delta(x)^T \nabla_x e_j\|_2 \tag{5.5}$$

Therefore to affect an output of the quantisation block Φ_q, a perturbation $\delta(x)$ should lead to a shift in the embedding space, e which is greater than r.

We now derive equation 5.5 in lemma 1 below.

Proof. We first perform a first-order Taylor expansion of $\Phi_e(x + \delta(x))$ shown below

$$e_i - (e_i + \Delta) = \delta(x)^T \nabla_x e_i \tag{5.6}$$

Next, we can then deduce the following to conclude our derivation.

$$\|e_i - (e_i + \Delta)\|_2 = \|\delta(x)^T \nabla_x e_i\|_2 \tag{5.7}$$

$$\therefore r > \|\delta(x)^T \nabla_x e_j\|_2 \tag{5.8}$$

$$x, \delta(x) \in \mathcal{R}^N \quad e_i, e_i + \delta \in \mathcal{R}^D \quad \nabla_x e_i \in \mathcal{R}^{N \times D}$$

□

5.2.4 Implementation Details and Data

We demonstrate through a set of experiments that quantising the latent space is robust to different noise characteristics for both 2D and 3D medical image segmentation tasks. We also assess how quantisation improves the robustness of a segmentation model when testing on a new domain in both the 2D and 3D settings.

Architecture:

We consider the UNet as our benchmark segmentation architecture and for the proposed architecture, *VQ-UNet*, we add a vector quantisation block in the bottleneck layer of the baseline UNet. Our codebook size is 1024 each of dimension 256. We consider both 2D and 3D UNets with 5 levels. In the encoder, we double the number of feature channels from 32 and 16 in the first level to 512 and 256 in the bottleneck, respectively for the 2D and 3D models. Each scale of the encoder and decoder consists of a single pre-activation residual block [HZRS16b], with group normalisation ([WH18]) and swish activation ([RZL17]).

Training:

We fine-tuned the hyper-parameter β in the loss function defined in equation 5.2 to be 0.25. The loss function for training the UNet is the sum of the first two terms of equation 5.2 (Dice/cross entropy). We train with a batch size of 10 and 2 for the 2D and 3D tasks, respectively. We apply the same spatial augmentation strategy for all models

which include vertical or horizontal flipping followed by random rotations between -90 and 90 degrees. Adam optimisation is used with a base learning rate of 0.0001 and weight decay of 0.05 [KB14]. We train all models for a maximum of 500 epochs on three NVIDIA RTX 2080 GPUs.

Datasets:

We use the following 3 datasets for our experimental study:

Abdomen: We use the BTCV dataset consisting of 30 CT scans with 13 labels acquired from a single domain (Vanderbilt University Medical Center) [LXI+15]. All images were normalised to 0-1 and resampled to 1.5×1.5×2mm. We randomly crop 96×96×96 patches for training. We use the 3D UNet for this dataset. During testing, the volume is divided into $96 \times 96 \times 96$ patches and we evaluate each patch in the test image.

Prostate: The prostate dataset originates from the NCI-ISBI13 Challenge [BMH+15b] with data coming from two domains. All images were re-sampled to 0.5×0.5×1.5mm and z-score normalized. We centre crop to 192×192×64. We use the 3D UNet for this dataset.

Chest-X-ray: We use the NIH chest X-ray dataset [WPL+17] with annotations provided by [TTXS19] and the JSRT dataset [SKI+00] for domain shift analysis. Images were resized to 512×512 pixels and normalised to 0-1. We use the 2D UNet for this dataset.

5.3 Experiments

5.3.1 Codebook Study

NIH	JSRT	Abdomen	BMC	RUNMC
0.001 ± 0.011	0.002 ± 0.014	0.011 ± 0.018	0.015 ± 0.021	0.012 ± 0.019

Table 5.1: Mean $r \pm 1$ standard deviation for all 5 datasets.

We first analyse whether assumption 3 holds by calculating r and its standard deviation based on equation 5.4. We note that there is a very large standard deviation around r

ranging from 0.011 to 0.021 for all 5 datasets (Table 5.1). This suggests the 5 codebooks are not uniformly distributed i.e., incomplete or concentrated in certain regions of space. Hence, we cannot reliably assume that a shift in the latent space greater than r represents the boundary between a meaningful semantic shift and a perturbation. Therefore, r is obsolete, and we can only denote r_i; the distance from each codebook vector (l_i) to its nearest neighbour. r_i allows us to at least calculate the maximally allowed perturbation in the latent space for each learnt codebook vector. This is important to bear in mind when we analyse the results of our experiments.

5.3.2 Domain Shift Study

We will first evaluate how the segmentation performance of the VQ-UNet differs from the UNet on a single domain and across domains for the chest X-ray and prostate datasets. We use two evaluation metrics; the Dice score and 95% Hausdorff distance in mm (HD95). We randomly split a single domain in the prostate dataset into 24 for training and 6 for validation and using the best-trained model based on the Dice score for testing on the second domain (30). We randomly select 20 and 30 samples for validation, in the NIH and JSRT chest X-ray datasets respectively to find the best model to test on the second domain.

	Chest X-ray							
	JSRT		**NIH**		**JSRT→NIH**		**NIH→JSRT**	
	Dice	**HD95**	**Dice**	**HD95**	**Dice**	**HD95**	**Dice**	**HD95**
UNet	0.93±0.03	7.31±1.12	0.96±0.02	6.80±0.94	0.95±0.02	7.12±0.97	0.82±0.05	8.27±1.10
VQ-UNet	**0.94±0.02**	**7.21**±1.03	**0.97±0.01**	**6.01**±0.81	**0.96±0.01**	**6.51±0.86**	**0.85±0.07**	**7.79±0.89**
UNet	0.93±0.04	7.31±0.91	0.96±0.02	6.80±0.86	0.95±0.02	7.12±1.00	0.82±0.06	8.27±1.03
VQ-UNet	**0.94±0.03**	**7.21±0.97**	**0.97±0.01**	**6.01±0.89**	**0.96±0.01**	**6.51±0.96**	**0.85±0.05**	**7.79±1.07**
	Prostate							
	BMC		**RUNMC**		**BMC→RUNMC**		**RUNMC→BMC**	
	Dice	**HD95**	**Dice**	**HD95**	**Dice**	**HD95**	**Dice**	**HD95**
UNet	0.80±0.08	8.42±1.24	**0.82±0.07**	7.84±0.99	0.55±0.13	33.3±4.76	0.62±0.10	25.7±5.98
VQ-UNet	**0.82±0.08**	**7.82±0.91**	0.82±0.09	**7.11±0.90**	**0.59±0.18**	**31.5±7.01**	**0.71±0.15**	**21.4±5.88**
UNet	0.93±0.03	7.31±1.09	0.96±0.02	6.80±0.88	0.95±0.02	7.12±1.05	0.82±0.08	8.27±1.14
VQ-UNet	**0.94±0.03**	**7.21±0.98**	**0.97±0.01**	**6.01±0.83**	**0.96±0.02**	**6.51±0.79**	**0.85±0.07**	**7.79±0.86**

Table 5.2: Mean Dice and HD95 ± standard deviation on the validation sets from the same domain and the test set which is derived from another domain. The arrow represents the domain shift.

Overall, the VQ-UNet improved the segmentation performance of both the in-domain validation set and the test set from a different domain. This is demonstrated for both the prostate and chest X-ray datasets (Table 5.2). We note the UNet Dice score reduces to

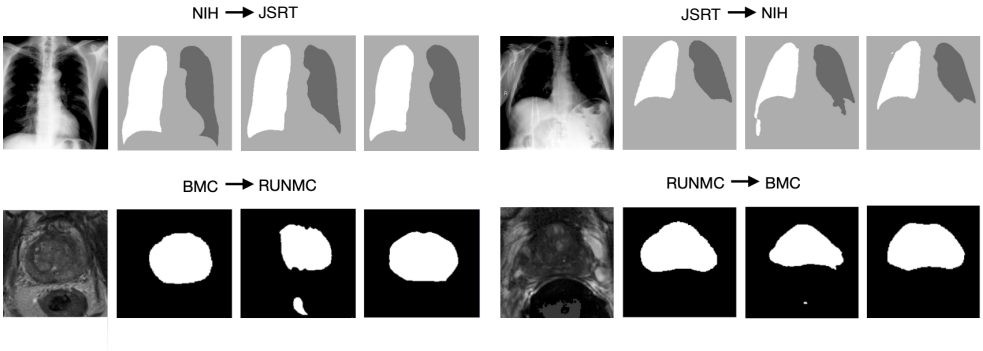


Figure 5.1: Sampled image input and Segmentation output for 2 domain shifts in chest X-ray (top row) and prostate (bottom row).

0.82 from 0.93 compared to 0.85 from 0.94 for the VQ-UNet when shifting the domain from JSRT to NIH (Table 5.2). For the prostate study, there is a significant domain shift and we note a significant drop in performance in terms of the Dice score and HD95 distances when testing on a different domain for both the UNet and VQ-UNet (Table 5.2). However, the VQ-UNet appears to be more robust to this domain shift. This is particularly noted when testing on the BMC dataset after training the VQ-UNet on RUNMC (Table 5.2). The drop in performance albeit improved compared to the UNet, is due to an incomplete codebook. It is highly likely the data from the test set maps to e which is a distance greater than r_i from the ground truth codebook vector (l_i). This suggests e is a perturbed version of a discrete point which is not in our incomplete codebook. Nonetheless, in Fig 5.1 we note the smoother, anatomically more plausible segmentation map of the VQ-UNet compared to the UNet.

5.3.3 Perturbation Study

	Abdominal CT			**Chest X-ray**			**Prostate**		
	Gauss. Noise	**S &P Noise**	**Poisson Noise**	**Gauss. Noise**	**S &P Noise**	**Poisson Noise**	**Gauss. Noise**	**S &P Noise**	**Poisson Noise**
UNet	0.46	0.44	0.46	0.51	0.43	0.47	0.56	0.51	0.51
VQ-UNet	**3e-4**	**5e-5**	**2e-4**	**2e-4**	**1e-4**	**3e-4**	**1e-4**	**6e-5**	**8e-5**

Table 5.3: Average latent space variance in the UNet and VQ-UNet for all three datasets.

We compare how much the latent space changes in both models with different perturbations in the input space for three datasets (abdomen, NIH and BMC). There are a myriad of perturbations one can apply in the input space, so we choose three different types of noise perturbations (Gaussian, salt and pepper, and Poisson noise) under 5

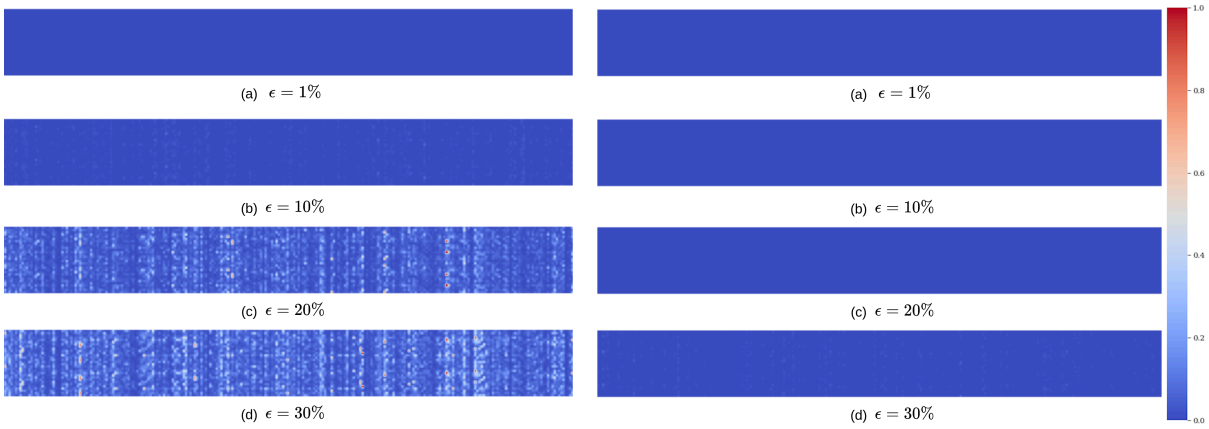


Figure 5.2: Variance heatmap of UNet(left) and VQ-UNet(right) latent space under 4 Gaussian noise levels for the NIH dataset. The X-axis indicates a unique subset of features from a latent space, the Y-axis corresponds to 100 randomly sampled test set images, and the value at each location indicates the variance of a specific feature for a given image across 100 test time augmentations with Gaussian noise.

noise levels ranging from 0% to 30% to justify our claim of robustness.

To evaluate the effect of noise on the latent space of the trained models, we sample 100 different noise vectors for each image at each noise level and observe the variance in the latent space on the in-domain validation set. Table 5.3 describes the average variance of latent space features in both models across all noise levels for each type of noise. It can seen that the latent space features in VQ-UNet are not significantly changed (close to 0 variance) under various types of noise. This is also noted for all three datasets. The results are visualised in Fig. 5.2 whereby the latent space of the VQ-UNet does not significantly change compared to the UNet under the addition of up to 30% Gaussian noise in the NIH dataset. This means that based on equation 5.5, ϵ of up to 30% is leading to a shift in the latent space of the VQ-UNet less than r_i.

In our analysis of the output space, table 5.4 indicates the effect of Gaussian perturbation on Dice scores on the in-domain validation set. It demonstrates the Dice scores are more stable in the VQ-UNet compared to the UNet for all three datasets up to 30% noise. We note identical findings for salt & pepper noise and Poisson noise in table 5.5

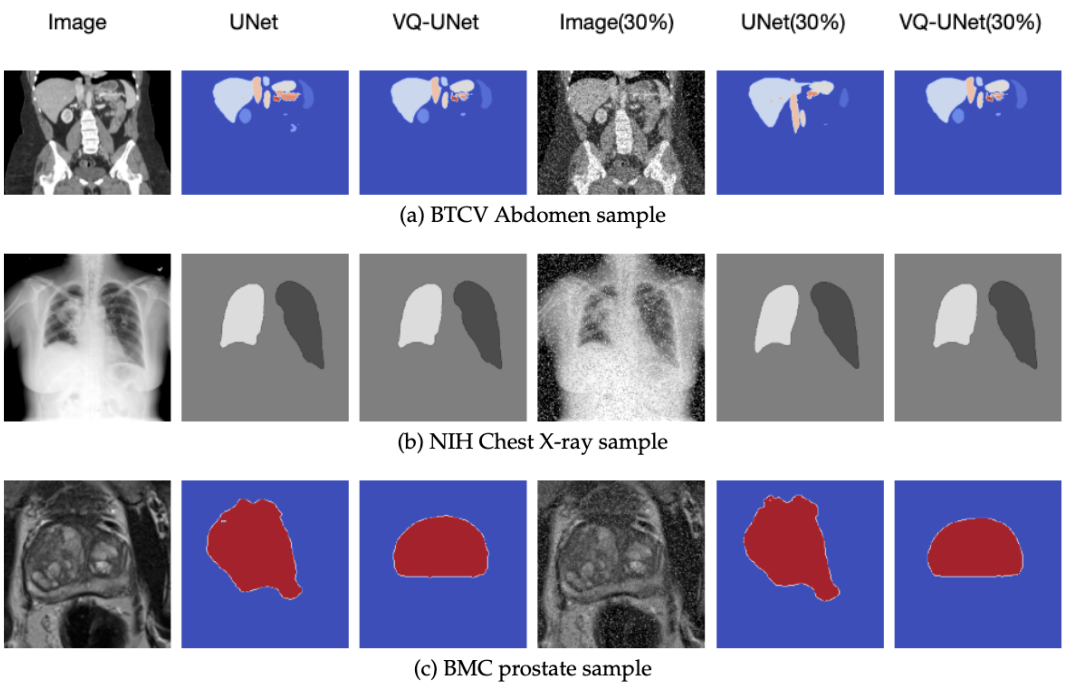


Figure 5.3: Sampled abdomen CT (Top row), chest X-ray (Middle Row) and prostate MRI (Bottom row) input images and Segmentation outputs for the UNet and VQ-UNet under 0% (1^{st} 3 columns) and 30% (2^{nd} 3 columns) s&p noise.

and 5.6 respectively where again the Dice score remains stable up to 30% noise. We highlight this result further in Figure 5.3 which demonstrates that the segmentation maps produced by the VQ-UNet under the addition of 30% salt & pepper noise noise do not change visually compared to the UNet.

Overall, the perturbation experiments show that quantisation helps in mitigating the effect of noise perturbation on the latent space, thereby generating non-corrupted segmentation maps. This is in contrast to the prostate domain shift experiments whereby the domain shift generates a shift in the latent space larger than r_i for each codebook vector or maps to perturbations from discrete points not present in the codebook.

We next calculate the relative corruption error in terms of the Dice score of the VQ-UNet compared to the UNet for all 3 three datasets under all possible perturbations. The relative corruption error in terms of degradation of the Dice score is 41.3%, 68.4% and 29.6% for the chest x-ray, abdomen and prostate datasets respectively which are all significantly less than 100%. This further highlights the superior robustness of the VQ-UNet compared to the UNet under numerous types of input perturbations.

	Dice @0% ϵ	Dice @1% ϵ	Dice @10% ϵ	Dice @20% ϵ	Dice @30% ϵ
Chest X-ray NIH dataset					
UNet	0.95 ± 0.02	0.96 ± 0.02	0.95 ± 0.03	0.95 ± 0.03	0.95 ± 0.03
VQ-UNet	**0.97 ± 0.01**	**0.97 ± 0.01**	**0.97 ± 0.01**	**0.96 ± 0.02**	**0.96 ± 0.02**
Abdominal CT					
UNet	0.77 ± 0.07	0.76 ± 0.07	0.77 ± 0.09	0.76 ± 0.10	0.75 ± 0.11
VQ-UNet	**0.80 ± 0.05**	**0.79 ± 0.09**	**0.80 ± 0.08**	**0.80 ± 0.08**	**0.79 ± 0.07**
Prostate BMC dataset					
UNet	0.80 ± 0.09	0.81 ± 0.09	0.80 ± 0.10	0.78 ± 0.11	0.77 ± 0.11
VQ-UNet	**0.82 ± 0.06**	**0.82 ± 0.06**	**0.82 ± 0.05**	**0.82 ± 0.07**	**0.80 ± 0.07**

Table 5.4: Gaussian noise perturbation results on all 3 datasets.

	Dice @0% ϵ	Dice @1% ϵ	Dice @10%ϵ	Dice @20% ϵ	Dice @30% ϵ
Chest X-ray NIH dataset					
UNet	0.96 ± 0.02	0.95 ± 0.02	0.92 ± 0.05	0.89 ± 0.08	0.86 ± 0.08
VQ-UNet	**0.97 ± 0.01**	**0.97 ± 0.01**	**0.95 ± 0.01**	**0.95 ± 0.02**	**0.95 ± 0.02**
Abdominal CT					
UNet	0.77 ± 0.07	0.77 ± 0.07	0.76 ± 0.07	0.74 ± 0.09	0.72 ± 0.11
VQ-UNet	**0.80 ± 0.05**	**0.79 ± 0.04**	**0.78 ± 0.08**	**0.78 ± 0.08**	**0.78 ± 0.10**
Prostate BMC dataset					
UNet	0.80 ± 0.06	0.80 ± 0.05	0.76 ± 0.08	0.75 ± 0.08	0.70 ± 0.11
VQ-UNet	**0.82 ± 0.06**	**0.82 ± 0.05**	**0.81 ± 0.07**	**0.82 ± 0.05**	**0.80 ± 0.06**

Table 5.5: Salt and Pepper perturbation results on all 3 datasets.

	Dice @0% ϵ	Dice @1% ϵ	Dice @10% ϵ	Dice @20% ϵ	Dice @30% ϵ
Chest X-ray NIH dataset					
UNet	0.96 ± 0.02	0.96 ± 0.02	0.94 ± 0.04	0.94 ± 0.04	0.92 ± 0.05
VQ-UNet	**0.97 ± 0.01**	**0.97 ± 0.01**	**0.96 ± 0.01**	**0.96 ± 0.02**	**0.95 ± 0.02**
Abdominal CT					
UNet	0.77 ± 0.07	0.76 ± 0.06	0.76 ± 0.08	0.75 ± 0.08	0.75 ± 0.07
VQ-UNet	**0.80 ± 0.05**	**0.80 ± 0.06**	**0.79 ± 0.05**	**0.79 ± 0.06**	**0.78 ± 0.07**
Prostate BMC dataset					
UNet	0.80 ± 0.04	0.80 ± 0.05	0.81 ± 0.04	0.78 ± 0.07	0.77 ± 0.09
VQ-UNet	**0.82 ± 0.05**	**0.81 ± 0.03**	**0.82 ± 0.06**	**0.82 ± 0.06**	**0.80 ± 0.07**

Table 5.6: Poisson noise perturbation results on all 3 datasets.

5.4 Conclusion

We propose and justify that given a segmentation architecture which maps the input space to a low dimensional embedding space, learning a discrete latent space via quantisation improves the robustness of the segmentation model. We highlight quantisation to be especially useful in the task of anatomical segmentation where the output space is structured and hence the codebook metaphorically behaves like an atlas in latent space. This however also possibly limits quantisation in highly variable segmentation tasks such as tumour segmentation.

We also note in our experiments that the codebook vectors are not uniformly distributed and often are concentrated in certain regions in space. This is a problem we tackle in the next chapter where we apply geometric constraints on the distribution of the codebook.

Chapter 6

Hierarchical Compositionality in Hyperbolic Space for Robust Medical Image Segmentation

This chapter is adapted from the following publication.

- **Santhirasekaram, A.**, Winkler, M., Rockall, A. and Glocker, B., 2023, October. Hierarchical Compositionality in Hyperbolic Space for Robust Medical Image Segmentation. In MICCAI Workshop on Domain Adaptation and Representation Transfer (pp. 52-62). Cham: Springer Nature Switzerland.

6.1 Introduction

In the previous chapter, we assumed the codebook was uniformly distributed. We however showed empirically this not to be true and found the codebook to either be incomplete or concentrated in certain regions of space. We also do not enforce any structural prior on the sampling of the codebook which prevents sampling the same codebook elements repeatedly for each segmentation class. In this chapter, we combat this issue by enforcing structural constraints on the codebook embeddings.

In this work, the base components are learnt by discretising the latent space via quantisation to form a dictionary of components for each class [VDOV+17]. We then enforce a hierarchical prior in the composition of the base components to construct the segmentation output. In order to learn the composition of the sampled-based components hierarchically, we consider the natural geometry in which to build our hierarchy. We therefore embed our base components in hyperbolic space. It has recently been shown that spatial representations in the brain exhibit hyperbolic geometry [ZRLS22]. We hypothesise hyperbolic geometry will provide the natural structure to efficiently embed the base components without distortion. We also hypothesise by further constraining the discretised latent space with a hierarchical prior in the composition of the base components to construct the segmentation output, one can increase the robustness of the segmentation model. Also, by uniformly distributing the codebook elements which are bottom of the hierarchy on the edge of the hyperbolic manifold, one increases the average distance between codebook elements. This means the required perturbation in the input space to shift from one codebook element to its nearest codebook element increases. The advantage of the hyperbolic manifold over the hyper-spherical manifold is that hyperbolic space is infinite whereas the hyper-sphere is finite which means as the number of components uniformly distributed on the manifold increases, then the distance between components decreases on the hyper-sphere which is not true for hyperbolic space. The advantage of hyperbolic space over Euclidean space is that space grows quadratically from the centre in Euclidean space whereas it grows exponentially from the centre in hyperbolic space. This makes hyperbolic space better equipped to deal with hierarchical data such as trees.

6.2 Related Work

The idea of compositionality has been incorporated into neural networks to improve the robustness of image classification under partial occlusion [KHLY20]. Recently, compositional representational learning by learning a discrete latent space has been shown to improve the robustness of segmentation models [LTS+22].

In our work, hyperbolic embeddings provide the natural basis for hierarchically com-

posing a discretised latent space. Hyperbolic embeddings have become increasingly popular for dealing with hierarchical data. For example, [NK17] suggests that the Poincaré model is better equipped to learn word embeddings to simultaneously capture a hierarchy and similarity between embeddings, which led to promising results in link prediction. Natural hierarchies explored in natural language processing often extend to the visual domain. For instance, visual concepts like various animal species lend themselves to hierarchical grouping. This notion was exploited by [KMU+20] who proposed learning hyperbolic image embeddings and demonstrated a significant boost in accuracy for few-shot image classification compared to Euclidean embeddings. Similarly, [LCP+20] proposed a hyperbolic visual embedding network for zero-shot recognition. [GAKRM21] proposed a prototype-based network on hyperbolic manifolds using Hyperbolic Busemann learning. Here, prototypes are positioned on the ideal boundary of the Poincaré ball without prior label knowledge and proximities to ideal prototypes are computed with penalised Busemann loss.

Recently, there have been works which exploit hyperbolic geometry for image segmentation. Here, hyperbolic image segmentation was developed to classify output pixels in hyperbolic space by learning hyperbolic planes for improved class separation [ASA+22]. [HGW+21] proposed using hyperbolic representations to better capture implicit hierarchical relationships between sub-volumes in a 3D image. The proposed approach involves a 3D hyperbolic variational auto-encoder with a gyroplane convolutional layer to map from the embedding space to the image space. An additional self-supervised loss, beyond the standard VAE loss, is introduced to infer approximate hierarchies and encourage related sub-volumes to be closer in the embedding space. Their hypothesis was validated on synthetic and medical datasets.

6.3 Methods

6.3.1 Base Component Learning

Similar to the previous chapter, we assume that a segmentation network can be decomposed into an encoder Φ_e to map the input space X to a lower dimensional embedding

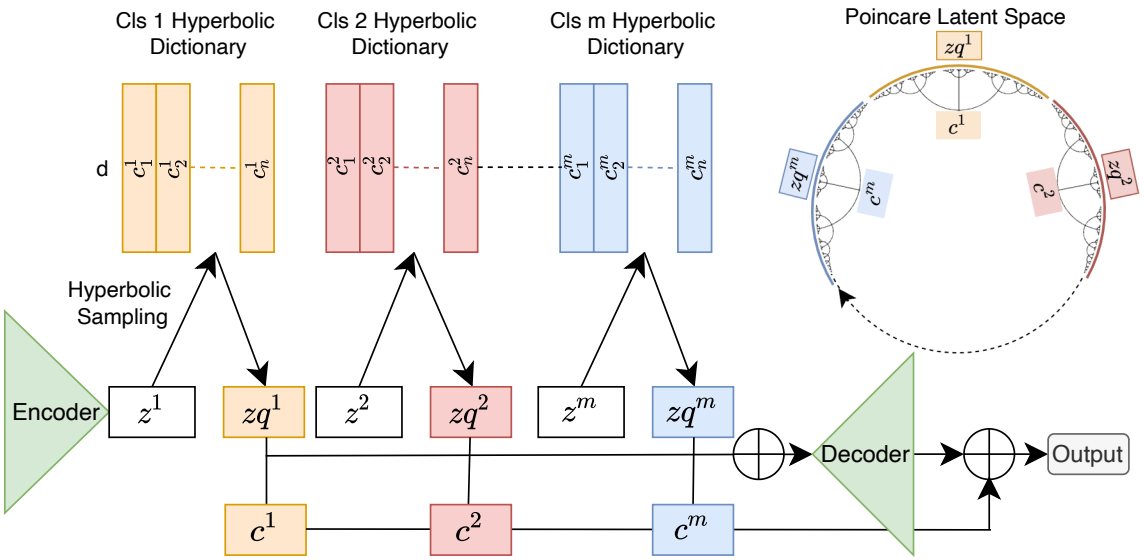


Figure 6.1: Our method (HypC) learns base dictionaries embedded in hyperbolic space for each class. The embedding features, z are split into m (number of classes) groups with each group sampling its respective class hyperbolic dictionary, $\mathbb{D}^i$ to form zq^i which is composed to form class segmentation outputs, c^i.

space denoted, $\mathcal{E}$ and a decoder to map $\mathcal{E}$ to the output space Y.

We propose to discretise $\mathcal{E}$ into a set of N base components to form a d dimensional dictionary $\mathbb{D}$ using quantisation [VDOV+17]. However, to induce compositionality into our base components, we enforce $\mathbb{D}$ to be divided into m classes such that $\mathbb{D}^i$ contains the base components required to compose each class segmentation output as shown in figure 6.1. This is further enforced by using only grouped convolutions equivalent to the number of classes in the encoder and decoder. The embedding features, z are divided into m groups with each group of n features, z^i sampled from its respective class dictionary $\mathbb{D}^i$ to produce discrete features zq^i. zq^i represents the components required to compose lower dimensional class segmentation outputs c^i with d dimensionality. This allows us to directly enforce a hierarchical relationship between zq^i and c^i in d dimensional hyperbolic space. The composition of the features in zq^i to form c^i shown in figure 6.1 is performed with a 1×1 convolutional layer.

In contrast with the previous chapter, we propose to discretise the embedding space with Gumbel quantisation [JGP16]. A random variable, G is defined to have the standard Gumbel distribution if it satisfies, $G = -\log(-\log(U))$ given U is sampled uniformly in the closed interval, $[0, 1]$. The Gumbel softmax trick can be used to sample a categorical distribution which in our case is the dictionary $\mathbb{D}^i$ in a fully differentiable

manner [JGP16]. Equation 6.1, shows a categorical distribution for each discrete feature $zq_l^i \in zq^i$ over the components in $\mathbb{D}^i$. This essentially tells us the probability of sampling each of the n components in $\mathbb{D}^i$ to form zq_l^i denoted, $p(zq_l^i = \mathbb{D}^i) \in \mathbb{R}^n$. In equation 6.1, the probability distribution is parameterised by $\alpha_l^i \in \mathbb{R}^n$ which one can note are unnormalised probabilities proportional to $\log(P(zq_l^i = \mathbb{D}^i))$. α_l^i is determined via a parameterised linear transformation, F of z_l^i to match the number dictionary elements (n) followed by a ReLU activation function. The parameters are shared across features in z^i. However, to compute the gradient with respect to the parameters of the deterministic function F, the reparameterisation trick is applied by adding noise $g \in \mathbb{R}^n$ sampled from standard Gumbel distribution to α_l^i. A normalised probability distribution is obtained by applying the softmax function in equation 6.1.

$$p(zq_l^i = \mathbb{D}^i) = \frac{\exp(\alpha_l^i + g)/\tau}{\sum_{j=1}^{j=n} \exp(\alpha_{l,j}^i + g)/\tau} \tag{6.1}$$

The temperature scale parameter τ determines the uncertainty in sampling $\mathbb{D}^i$ whereby as τ approaches 0, the distribution $p(zq_l^i = \mathbb{D}^i)$ becomes a one-hot vector. We determine $\tau = 0.5$ for optimal performance through our ablation experiments. During the forward pass for training and inference, we perform a soft sampling (weighted by $p(zq_l^i = \mathbb{D}^i)$) of $\mathbb{D}^i$ to construct the discrete feature zq_l^i formally defined as, $zq_l^i = p(zq_l^i = \mathbb{D}^i)^\top \mathbb{D}^i$. We found this to achieve superior results compared to performing a hard sampling of $\mathbb{D}^i$ via the $\arg\max_j$ function.

6.3.2 Hyperbolic embeddings

We initialise $\mathbb{D}^i$ on the Poincaré unit ball ($\mathbb{B}^{d,1}$) but perform soft Gumbel quantisation on the tangent plane of the origin via a logarithmic mapping of $\mathbb{D}^i$ formally defined as: $zq_l^i = p(zq_l^i = \mathbb{D}^i)^\top \log_o^{\mathbb{B},1}(\mathbb{D}^i)$.

Equation 2.56 is then used to apply an exponential mapping ($\exp_o^{\mathbb{B},1}$) of $\mathbb{D}^i$ to move back into Poincaré space.

An exponential mapping (equation 2.56) of zq and c to the Poincaré unit ball will exploit the natural geometry of hyperbolic manifolds to enforce a hierarchical composition of

zq to form c as shown in the Poincaré latent space in figure 6.1. This is achieved by defining a loss function shown in equation 6.2. Equation 6.2 shows the p features in zq^i are being pulled closer together in Poincaré distance (equation 2.53) to its composition c^i while being pushed apart from the compositions not composed from zq^i defined as $\bar{c}^i$. zq is then mapped back to Euclidean space with a logarithmic mapping and inputted into the decoder to construct the high-dimensional segmentation output shown in figure 6.1. We predict this hyperbolic hierarchical organisation of the latent space serves to learn more robust embedding features for constructing the segmentation output.

$$\mathcal{L}_{Poincare} = \frac{\sum_{i=1}^{m}\sum_{l=1}^{p} e^{d^{\mathbb{B},1}(zq_l^i, c^i)}}{\sum_{i=1}^{m}\sum_{l=1}^{p} e^{d^{\mathbb{B},1}(zq_{l'}^i, \bar{c}^i)} + \sum_{i=1}^{m}\sum_{i'=1}^{m-1} e^{d^{\mathbb{B},1}(c^i, \bar{c}^i)}} \tag{6.2}$$

We apply a Dice loss between the output y' and the label y as well as between the lower dimensional class segmentation outputs c and a down-sampled label of y to match the dimensionality of c denoted, y_2. The total loss during training is calculated as:
$\mathcal{L}_{Total} = \mathcal{L}_{Dice}(y', y) + \mathcal{L}_{Dice}(c, y_2) + \mathcal{L}_{Poincare}$.

6.3.3 Datasets and Training

Abdomen: This abdominal dataset is the BTCV dataset [LXI+15] as described in chapter 3. During training and inference, we randomly crop $96 \times 96 \times 96$ patches which are normalised between 0 and 1. During testing, the volume is divided into $96 \times 96 \times 96$ patches and we evaluate each patch in the test image.

Prostate: The prostate MRI dataset is from the NCI-ISBI13 Challenge [BMH+15b, CVS+13, BMH+15a] as previously described in chapter 3. All images are normalised between 0 and 1 and centre cropped to $256 \times 256 \times 24$.

Training and Evaluation: The models in all of our experiments are trained with Adam optimisation [KB14] at a learning rate of 0.0001 and weight decay (0.05). The parameters in $\mathbb{D}^i$ are mapped to Euclidean space using a logarithmic mapping (equation 2.61) for optimisation. All models were trained for a maximum of 500 epochs on three NVIDIA RTX 2080 GPUs. We perform simple data augmentation with random flipping (horizontal and vertical) and rotation. We evaluate model performance with the Dice

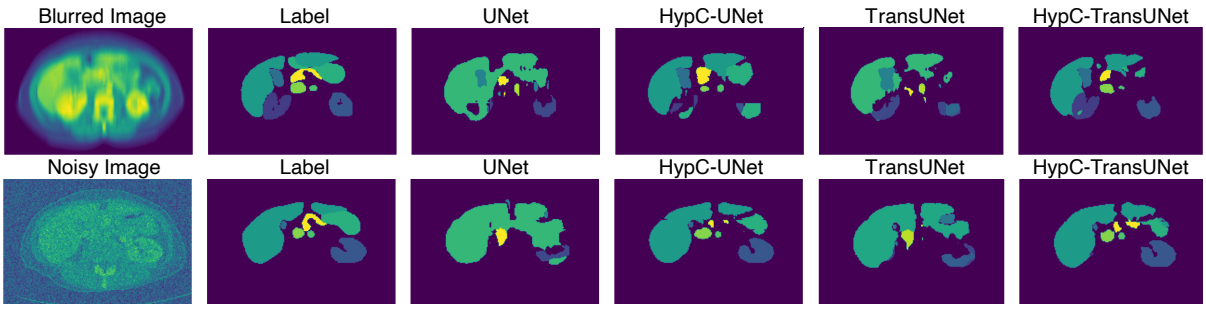


Figure 6.2: The figure shows the segmentations of an abdominal CT slice with 20% Gaussian noise addition (bottom row) and Gaussian blur (top row) produced by the UNet and TransUNet before and after incorporating our method.

score and maximum Hausdorff distance (mm). In the input perturbation experiment, we calculate the relative corruption error in terms of the Dice score for our hierarchical compositional model with different curvature constraints compared to the baseline model. We perform ablations to learn dictionaries, $\mathbb{D}^i$ which are as sparse as possible without affecting segmentation performance to improve model robustness.

6.4 Experiments

6.4.1 Perturbation experiments

In the first experiment, we observe the effect of incorporating a hyperbolic hierarchical compositional structure (HypC) in the embedding space of a segmentation model under various types of perturbations. We specifically analyse the robustness of the 3D UNet [RFB15, MNA16] and TransUNet [CLY+21], two popular segmentation architectures. In both models, 3D pre-activation residual blocks [HZRS16b] make up each level of a 5-level, convolutional encoder and decoder. In the TransUNet [CLY+21], the transformer block consisting of 12 layers and 8 heads is incorporated after the HypC block.

We also evaluate the importance of equipping the hierarchical structure of our discrete latent space with hyperbolic geometry. We therefore evaluate empirically the results in the perturbation experiments where the codebooks are in hyperbolic space compared with embedding our class codebooks in Euclidean space (0 curvature) or on a hyper-sphere (+1 curvature). The loss function to train the Euclidean and hyper-spherical

	Baseline	Gauss	S&P	Blur	Motion	Contrast	Intensity
	Dice						
UNet	0.77±0.08	0.71±0.10	0.69±0.06	0.72±0.11	0.63±0.13	0.62±0.14	0.68±0.11
VQ-UNet	**0.80±0.05**	**0.78±0.07**	**0.75±0.05**	0.75±0.10	0.65±0.15	**0.64±0.15**	**0.72±0.09**
EucC-UNet	0.78±0.08	0.73±0.18	0.71±0.09	0.75±0.14	0.64±0.17	**0.64±0.18**	0.71±0.18
SphC-UNet	0.77±0.05	0.74±0.11	0.70±0.13	0.76±0.13	0.63±0.18	0.63±0.19	0.70±0.19
HypC-UNet	0.78±0.07	0.76±0.04	0.73±0.04	**0.76±0.10**	**0.67±0.04**	0.63±0.07	0.71±0.11
TransUNet	0.82±0.08	0.78±0.12	0.76±0.09	0.79±0.08	0.70±0.11	0.74±0.17	0.73±0.11
VQ-TransUNet	0.82±0.09	0.79±0.09	0.78±0.12	0.80±0.10	0.72±0.13	0.75±0.13	0.75±0.12
EucC-TransUNet	0.82±0.12	0.80±0.14	0.78±0.08	**0.81±0.18**	0.73±0.10	**0.77±0.12**	0.74±0.11
SphC-TransUNet	**0.83±0.16**	0.79±0.14	0.78±0.11	0.80±0.18	0.72±0.08	0.75±0.15	**0.76±0.10**
HypC-TransUNet	0.82±0.11	**0.81±0.09**	**0.79±0.10**	**0.81±0.03**	**0.74±0.15**	0.76±0.11	0.75±0.13
	Hausdorff Distance (mm)						
UNet	24.32±12.13	38.55±15.43	40.39±19.09	30.39±16.18	45.48±12.17	44.28±20.04	37.39±18.27
VQ-UNet	23.86±11.13	35.72±15.98	**36.49±13.76**	29.09±14.11	42.78±16.20	42.94±13.82	35.73±11.03
EucC-UNet	23.57±11.42	35.65±20.41	37.19±19.55	26.61±13.51	42.18±17.29	40.49±21.03	35.65±18.29
SphC-UNet	23.42±18.39	35.58±18.16	39.73±13.29	26.48±19.04	40.59±18.32	41.63±24.92	34.86±13.62
HypC-UNet	**22.98±10.39**	**34.63±13.19**	38.71±19.39	**25.48±10.19**	**39.35±18.13**	**39.25±16.71**	**34.48±13.95**
TransUNet	23.62±8.71	33.69±15.14	35.58±16.38	28.47±10.38	34.25±8.58	33.41±15.62	30.94±9.35
VQ-TransUNet	22.47±9.38	30.58±14.99	32.84±10.58	27.19±8.37	32.91±9.73	32.84±13.85	28.90±8.61
EucC-TransUNet	22.55±5.59	30.84±19.58	34.72±16.28	26.16±11.85	**30.23±8.29**	29.28±13.31	28.68±9.36
SphC-TransUNet	22.82±11.22	31.42±11.49	34.05±14.88	27.94±16.39	32.08±10.80	29.44±15.56	**26.60±9.98**
HypC-TransUNet	**21.99±4.33**	**30.46±8.73**	**33.27±16.85**	**25.53±13.06**	32.19±12.64	**27.14±10.88**	27.00±15.17

Table 6.1: The mean Dice score and Hausdorff distance ± standard deviation under various perturbations in the input space for the Abdominal dataset of the UNet and TransUNet with hierarchical compositionality imposed with three different types of curvature constraints. We look at Euclidean (EucC: 0 curvature), hyper-spherical (SphC: +1 curvature) and hyperbolic (HypC: -1 curvature) geometry for hierarchical compositionality. We also show the results of the UNet and TransUNet with no constraints imposed.

versions of our model is similar to our hyperbolic model, except we replace the Poincaré distance with the Euclidean and Cosine distance for the Euclidean and hyper-spherical models respectively. We finally compare this method to the VQ-UNet in chapter 5 as well as to applying vector quantisation to the latent space of the TransUNet which we denote as the VQ-TransUNet.

We apply our method to the abdomen dataset which is split into 24 for training and 6 for testing.

Noise levels were varied between 1 to 30 % (1, 5, 10, 15, 20, 25, 30 %) for the Gaussian and Salt and Pepper (S&P) noise experiments. We use a Gaussian Blur incorporated with a Gaussian kernel which has a window size of 7×7 and a variance ranging from 0.1 to 2.0. Random motion blur is applied by using the TorchIO deep learning library [PGSO21]. The gamma values ranged from 0.5 to 4.5 for the contrast variations. Finally, for the intensity perturbations, intensity values are scaled by a factor between 0.8 and 1.2. The Dice scores and Betti errors are averaged across all parameter values used in the perturbations.

Table 6.1 highlights the improved absolute Dice scores and maximum Hausdorff distances of both the UNet and TransUNet under all types of perturbation effects after imposing HypC into the embedding space. We specifically noticed improved robustness of the two models under both textural and spatial distortions such as noise and motion perturbation. In table 6.1, one also notes that constraining the base components on a hyperbolic manifold which in this case is the Poincaré unit ball, is better for imposing hierarchical compositionality in the embedding space compared to on a hyper-sphere (positive curvature) or in Euclidean space (0 curvature). In our comparison with the VQ-UNet, we note the Dice score of the VQ-UNet is 0.80 under no perturbation which is 0.02 points higher than our method applied to the UNet under hyperbolic constraints. In addition, the Dice scores of the VQ-UNet is higher than our method applied to the UNet under all perturbations except for the blur and motion perturbation. In particular, under Gaussian and S&P noise, the VQ-UNet outperforms the HypC-UNet by 0.02 Dice points. In regards to the TransUNet, we show in table 6.1, that imposing hierarchical compositionality in the latent space under different curvature constraints appears to improve the Dice score compared to vector quantisation (VQ-TransUNet). In terms of the maximum Hausdorff distance, table 6.1 clearly demonstrates that the Hypc-UNet and Hypc-TransUNet outperform the VQ-UNet and VQ-TransUNet respectively. We highlight our findings visually in figure 6.2. Here, one can appreciate the smoother boundaries and reduced number of unconnected components after incorporating our method into the UNet and TransUNet under Gaussian noise and blur addition.

	VQ-UNet	**EucC-UNet**	**SphC-UNet**	**HypC-UNet**
UNet	89.4	87.7	80.7	73.6
	VQ-TransUNet	**EucC-TransUNet**	**SphC-TransUNet**	**HypC-TransUNet**
TransUNet	78.6	69.0	90.4	61.9

Table 6.2: Relative corruption error (%) in terms of the Dice score of our hierarchical compositional model under different curvature constraints as well as the VQ-UNet and VQ-TransUNet compared to the baseline UNet and TransUNet.

In table 6.2, we note the relative corruption error in terms of the Dice score decreases using a vector quantised and hierarchical compositional latent space in the latent space of the UNet and TransUNet. We also observe the relative corruption of the HypC-UNet (73.6%) and HypC-TransUNet (61.9%) is less compared to their Euclidean and hyper-spherical counterparts which demonstrate the increased robustness of imposing hierarchical compositionality in hyperbolic space. We finally observe in table 6.2 that

	Baseline	CutOut	BigAug	AdvBias	RandConv	Jigen	VQ-UNet	Ours(HypC-UNet)
	Transitional Zone							
Dice	0.62± 0.14	0.63± 0.15	0.66± 0.16	0.65± 0.11	0.67± 0.18	0.63± 0.15	0.66± 0.11	**0.68± 0.09**
HD	28.18± 6.74	26.19± 6.93	21.88± 5.96	24.78± 7.69	22.38± 7.31	27.52± 6.84	22.13± 5.89	**19.59± 5.81**
	Peripheral Zone							
Dice	0.36± 0.14	0.40± 0.11	**0.54± 0.18**	0.45± 0.16	0.47± 0.22	0.41± 0.15	0.48± 0.13	0.52± 0.15
HD	56.41± 22.19	49.03± 19.53	33.29± 12.34	36.64± 19.09	42.51± 17.10	46.21± 14.33	32.14± 13.61	**31.14± 16.07**

Table 6.3: Dice score and maximum Hausdorff distance(HD) ± standard deviation comparing our method to various other SDG methods. This table shows the segmentation results for the prostate zones averaged across both single domain shift experiments (*BMC* → *RUNMC* and *RUNMC* → *BMC*). In these experiments all methods are augmented with random rotations and flips in both the horizontal and vertical axis. BigAug however uses a more aggressive augmentation pipeline including a combination of Gaussian noise, blur, contrast changes and random affine transformations as designed exactly by [ZWY+20].

the HypC-UNet and HypC-TransUNet have a lower relative corruption error by 15.8% and 16.7% compared to the VQ-UNet and VQ-TransUNet respectively. We therefore show a hierarchical discrete latent space embedded in hyperbolic space is more robust compared to a vector quantised latent space for segmentation.

6.4.2 Single Domain Generalisation

We use the prostate dataset in our SDG experiments where we train on the RUNMC dataset to test on the BMC dataset and vice versa. To accommodate the anisotropic nature of the prostate MRI images, we use a hybrid 2D/3D UNet as our baseline model. This model has 4 levels, each level consisting of 2, 2D pre-activation residual blocks [HZRS16b] followed by a 2D strided convolutional layer for the encoder and a 2D up-sampling block for the decoder. The bottleneck level consists of a 3D pre-activation residual block. The hyperbolic hierarchical compositional structure is built into the bottleneck of the baseline model. We compare to the SDG methods; RandConv [XLY+20], CutOut [DT17], BigAug [ZWY+20], AdvBias [CQQ+20] and Jigen [CDB+19] applied to the baseline. We also compare to the VQ-UNet by incorporating the vector quantisation block in the bottleneck of our hybrid 2D/3D UNet.

In table 6.3, we demonstrate the superior performance of our approach in both metric scores compared to most other SDG methods. We achieve similar performance to

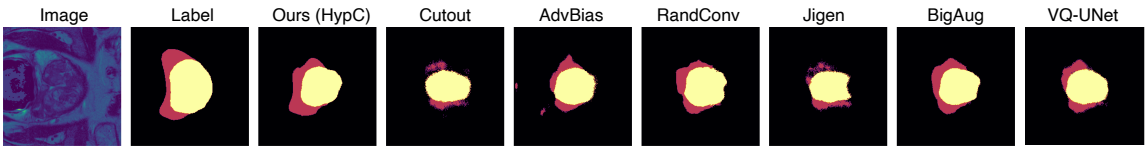


Figure 6.3: Segmentation of a prostate slice from the BMC dataset when training on the RUNMC dataset with different SDG methods and our method.

BigAug [ZWY+20] highlighting how incorporating a compositional structure under the natural geometric constraints improves the generalisability of a segmentation model even without an aggressive augmentation pipeline. This is demonstrated with a visual example in figure 6.3 which shows the more anatomically plausible segmentation maps produced by our method.

Ablation Experiments

We aim to learn dictionaries $\mathbb{D}^i$ which are as sparse as possible without affecting segmentation performance to learn a more compact representation to improve model robustness. In all models, we use grouped convolutions and group normalisation with leaky ReLU (0.2) activation in all the pre-activation residual and single-layer convolutional blocks. The number of groups used in the grouped convolutions and normalisation is equal to the number of classes (14 and 3 for the abdomen and prostate datasets respectively). Therefore, one must make sure the number of channels at each level of the encoder and decoder is divisible by this number.

	Prostate			**Abdomen**		
	Euclidean	**Hyper-sphere**	**Hyperbolic**	**Euclidean**	**Hyper-sphere**	**Hyperbolic**
Size	64	64	32	96	48	48
Channels	384	192	192	448	224	224

Table 6.4: Hyper-parameters used in our experiments. The 'size' is the number of components in the dictionary. 'Channels' is the number of embedding features in the model bottleneck.

In the ablation experiments shown in table 6.4, we note that without affecting segmentation performance, fewer embedding features and base components are required in $\mathbb{D}^i$ to impose hierarchical composition on a hyperbolic manifold compared to the other two spaces. This demonstrates a hyperbolic manifold as hypothesised is better equipped

to embed hierarchical data without distortion and overlap between base components which leads to more compact representations.

6.5 Conclusion

We propose to integrate hierarchical compositionality equipped with the natural geometry of a hyperbolic manifold into any existing segmentation model for improved robustness. Specifically, our approach improves model robustness to textural and spatial perturbations as well as domain shifts. We highlight in our results, that it is better to hierarchically compose and sample a set of learnt base components embedded in hyperbolic space compared to in spherical or Euclidean space. The main limitation of this work is that we assume a fixed unit negative curvature for all class codebooks to impose our compositionality constraint. Therefore, discovering the optimal curvatures to naturally embed each class codebook will likely improve the performance of our method. In future work, we aim to explore learnable curvature for compositionality in the embedding space to improve the performance of our method.

Chapter 7

A Geometric Approach to Robust Medical Segmentation

- **Santhirasekaram, A.**, Winkler, M., Rockall, A.G. and Glocker, B., 2023, April. Shape Equivariant Learning for Robust MRI Segmentation. In Medical Imaging with Deep Learning, short paper track.
- **Santhirasekaram, A.**, Winkler, M., Rockall, A.G. and Glocker, B., 2024. A Geometric Approach to Robust Medical Image Segmentation. *Under review*.

7.1 Introduction

In the previous chapters, we showed how constraining the latent space to a finite number of components to form a dictionary can improve the robustness of medical imaging segmentation models. We further improve robustness by showing the curvature in which one embeds these discrete components plays an important role in sampling and composing the elements from the dictionary to construct the segmentation output. However, in chapter 5 and 6, the components in the dictionary are not texture invariant or shape equivariant as such constraints have not been imposed. In this chapter, we explore these different geometric constraints to improve the robustness of any deep learning-based segmentation model.

A symmetry arises in a transformation applied to an object that retains some of its attributes. Symmetries naturally occur in the field of segmentation which includes, translation, rotation, and reflection along the body's symmetry axis. Generally, patients are positioned within the scanner in a predictable manner (fetal imaging being an exceptional deviation), and any divergence from the average patient alignment is usually slight (often up to 20 degrees). However, with a limited dataset, the observed patient orientations may not adequately reflect the diverse range of poses encountered in routine clinical settings.

An early demonstration of geometric inductive bias is the convolutional layer in a CNN. CNNs, crucial for computer vision tasks, owe their success largely to the translation-equivariant convolution operator which arises from using the same convolution kernel across the input signal [LBBH98]. CNNs can therefore learn feature detection at any location, leveraging inherent translational symmetries in many tasks. Group equivariant CNNs have been introduced to enforce equivariance beyond translational equivariance [CW16a] to include rotation and scale equivariance. It has been shown equivariant models will produce more compact models [Bek19].

Magnetic resonance imaging involves a complex acquisition process which differs across subjects and domains. This can lead to varying textural profiles and artefacts which affects the performance of deep learning-based segmentation models. We hypothesise in this work given a segmentation output only contains shape information, we can improve the robustness of segmentation models to textural shifts if we constrain our latent space to a finite number of shape-only components using vector quantisation. We achieve this by firstly enforcing texture invariance in the latent space. Next, we want each element in our shape dictionary to be unique and not contain linear transformations (i.e. rotations) of one another which will lead to a more compact shape codebook. We achieve this by enforcing equivariance to certain groups in the latent space.

7.2 Group Convolutions Preliminaries

CNNs involve a convolution kernel κ defined on $\mathbb{R}^d$ which modulates an input signal f on the same space. This application yields a function over a d dimensional grid $\mathbb{Z}^d$. Essentially, we first transform the convolution kernel κ for each group element $x \in \mathbb{Z}^d$, creating a set of kernels $\mathcal{L}_x(\kappa)$. Then, we perform the convolution operation on f using these transformed kernels $\mathcal{L}_x(\kappa)$. By maintaining consistent kernel weights across the translation group, we enable learned features to automatically adapt to different spatial positions [LB+95].

We will now go through the processes to derive a group equivariant convolution. There are two types of group CNNs, regular G-CNNs and Steerable G-CNNs. We first construct the recipe for the regular G-CNNs.

We will first introduce the **lifting convolution**. To maintain information about the position of features in the input, an equivariant convolution operation is achieved by elevating a function from the input space to a group's homogeneous space. For example, if we are specifically interested in image data that exists on a 2D grid $\mathbb{Z}^2$, so we consider the group of interest H in a semi-direct product with our data's domain; that is, $G = \mathbb{Z}^2 \circ H$. In group convolutions, every transformation in G left-acts on a given kernel, thereby generating a signal in the higher-dimensional space G instead of $\mathbb{Z}^2$. Given the group element $h \in H$ and the position $\tilde{x}$ in the input domain, we formally define the lifting convolution below.

$$(f *_{\text{lifting}} \kappa)(g) = \sum_{\tilde{x} \in \mathbb{Z}^2} f(\tilde{x}) \kappa_h(\tilde{x} - x) d\tilde{x} \tag{7.1}$$

In the above equation, $g = (x, h)$ and κ_h is the kernel acted upon by the group element $h \in H$ such that, $\mathcal{L}_h[\kappa](x) := \kappa(h^{-1}x)$ [CW16a]. The output dimension is the spatial dimension of x plus the group dimension h [CW16a].

We now have the domain transferred to the group space and therefore convolutional operations are now performed over the group space such that, $\kappa : G \to G$. Given, the Haar measures $d\tilde{g}$ and $d\tilde{h}$ over the group G and H respectively, we define the group

convolutional below [CW16a].

$$(f *_{\text{group}} \kappa)(g) = \sum_{G} f(\tilde{g})\kappa(g^{-1} \cdot \tilde{g})d\tilde{g} \tag{7.2}$$

$$= \sum_{g \in G} f(\tilde{g})\mathcal{L}_g \kappa(\tilde{g})d\tilde{g} \tag{7.3}$$

$$= \sum_{\tilde{x} \in \mathbb{Z}^2} f(\tilde{x},\tilde{h})\mathcal{L}_x \mathcal{L}_h \kappa(\tilde{x},\tilde{h})\frac{1}{|h|}d\tilde{x}d\tilde{h} \tag{7.4}$$

$$= \sum_{\tilde{x} \in \mathbb{Z}^2} f(\tilde{x},\tilde{h})\kappa(h^{-1}(\tilde{x} - x), h^{-1} \cdot \tilde{h})\frac{1}{|h|}d\tilde{x}d\tilde{h} \tag{7.5}$$

Equation 7.2 defines a function over G for all elements $g \in G$. We then express equation 7.2 in terms of (x, h) instead of g shown in equation 7.5. Regular group convolutions allow one to force exact lifting and group equivariant convolutions for finite groups such as the D4 group (dihedral group). This is because the sum in equation 7.2 is tractable. For continuous groups, we can either discretise the group H or approximate the group convolution through random sampling [CW16b]. We specifically uniformly sample the group elements in the group which are equidistant to each other. Discretisation ensures the network's exact equivariance is only to a discrete subgroup, which results in the network having a bias towards a specific subset of transformations. On the other hand, random sampling produces an unbiased estimate of the continuous group. Also, as one can imagine, a larger sample of the group will produce a better approximation of that group. However, this will increase the number of convolutional operations.

The representations that encapsulate both the group action and the effect of one group element on another are usually reducible [LW20]. This means that if the requirement to describe the group action is removed, then they can be simplified. Essentially, we can create decomposable unitary representation matrices composed of smaller block matrices and zero blocks shown in equation 7.6 [LW20]. The smaller blocks $\rho_i(g)$ are irreducible, meaning they cannot be further broken down into smaller units and zeros [LW20].

You can also create new representations by combining different irreducible representations such as $\rho_1(g)$ and $\rho_2(g)$ with direct sums i.e. equation 7.6 can also be expressed as

$\rho_0(g) \oplus \rho_1(g) \oplus \ldots \oplus \rho_k(g)$.

$$\rho(g) = \mathbf{S}^{-1} \begin{pmatrix} \rho_0(g) & 0 & \cdots & 0 \\ 0 & \rho_1(g) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \rho_k(g) \end{pmatrix} \mathbf{S} \tag{7.6}$$

These irreps are essential components in constructing more complex representations [LW20]. If the group is abelian, then all the irreducible representations are one-dimensional, whereas non-abelian groups are represented as square matrices. In essence, irreducible representations offer a versatile and compact way to represent and understand group structures, providing essential tools for analysis and interpretation.

The count of irreducible representations depends on the nature of the group: finite groups have a finite number of these, while locally compact groups have infinite. These irreducible representations function like orthonormal basis functions in Hilbert spaces [LW20]. According to the Peter-Weyl theorem, they can be used to form a complete basis-set for square-integrable L^2 functions on the group [LW20]. This transformation is essential since the irreducible representations, which typically output matrices representing g need to be converted to scalars to form a basis for an integrable function [LW20]. Integrable functions are derived from our space to our group using a lifting operation. Through this, irreducible representations enable us to express the functions as a direct sum of the coefficients of the irreducible representations. Importantly, the irreducible representations have been comprehensively defined for all groups, allowing us to refer to tables for their values instead of constructing them manually [LW20].

Given a representation of the group denoted, ρ and the equivariant constraint on the kernel denoted in equation 7.7, one can solve the kernel constraint for such representations by simplifying it through the use of irreducible representations (irreps) [LW20].

$$\kappa(gx) = \rho_{\text{out}}(g)\kappa(x)\rho_{\text{in}}(g^{-1}) \forall g \in G, x \in \mathbb{R}^2 \tag{7.7}$$

This approach hinges on the understanding that any representation of a finite or compact group can be transformed into a collection of irreps through a change of basis. These irreps correspond to invariant subspaces within the representation space and are

affected by the action of ρ. By using a change of basis denoted by Q, the representation ρ can be expressed as follows:

$$\rho = Q^{-1} \sum_{i \in I} \psi_i Q \tag{7.8}$$

Here, ψ_i are the irreducible representations of the group G, and the index set I describes the specific types and quantities of irreps contained within ρ. [LW20]

Given, the equivariant constraint imposed on the kernel we can decompose ρ_{in} and ρ_{out} into the following:

$$\kappa(gx) = Q_{\text{out}}^{-1} \left(\sum_{i \in I_{\text{out}}} \psi_i(g) \right) Q_{\text{out}} \kappa(x) Q_{\text{in}}^{-1} \left(\sum_{j \in I_{\text{in}}} \psi_j^{-1}(g) \right) Q_{\text{in}}, \forall g \in G, x \in \mathbb{R}^n \tag{7.9}$$

The constraint breaks down into independent constraints through left and right multiplication shown below [LW20, WC19].

$$\kappa_{ij}(gx) = \psi_i(g) \kappa_{ij}(x) \psi_j^{-1}(g), \quad \forall g \in G, x \in \mathbb{R}^2, \quad i \in I_{\text{out}}, j \in I_{\text{in}} \tag{7.10}$$

We will first solve the irreducible representation constraint shown in equation 7.10 for the SO(2) group. Except for the trivial representation denoted by ψ_0, all the remaining irreps are characterized as 2-dimensional rotation matrices with frequencies that belong to the set of positive natural numbers shown below [LW20, WC19].

$$\psi_{\text{SO(2)}}^0(r\theta) = 1 \tag{7.11}$$

$$\psi_{\text{SO(2)}}^k(r\theta) = \begin{bmatrix} \cos(k\theta) & -\sin(k\theta) \\ \sin(k\theta) & \cos(k\theta) \end{bmatrix} = \psi(k\theta), \quad k \in \mathbb{N} \tag{7.12}$$

Note, the action of any element from the SO(2) group is norm-preserving which means, $g\|x\| = \|x\|$, $\forall g \in G, x \in \mathbb{R}^2$. The kernel constraint also means the radial part of the kernel is free while the angular component is fixed. It is also important to realise that every irreducible representation in the SO(2) group correspond to a single angular frequency which means one can expand the kernel as a Fourier series with real-valued, radially dependent coefficients [LW20, WC19]. We can then plug in the Fourier expansion of the kernel into equation 7.7 and project onto individual harmonics to constrain the Fourier

ψ_m \ ψ_n	ψ_0	$\psi_n, n \in \mathbb{N}^+$
ψ_0	1	$\cos(n\phi)\ \sin(n\phi), \quad 9\sin(n\phi)\cos(n\phi)$
ψ_m, $m \in \mathbb{N}$	$\begin{bmatrix}\cos(m\phi)\\ \sin(m\phi)\end{bmatrix}'$ $\begin{bmatrix}9\sin(m\phi)\\ \cos(m\phi)\end{bmatrix}$	$\begin{bmatrix}\cos(m-n)\phi & 9\sin(m-n)\phi\\ \sin(m-n)\phi & \cos(m-n)\phi\end{bmatrix}', \begin{bmatrix}9\sin(m-n)\phi & 9\cos(m-n)\phi\\ \cos(m-n)\phi & 9\sin(m-n)\phi\end{bmatrix}'$ $\begin{bmatrix}\cos(m+n)\phi & \sin(m+n)\phi\\ \sin(m+n)\phi & 9\cos(m+n)\phi\end{bmatrix}', \begin{bmatrix}9\sin(m+n)\phi & \cos(m+n)\phi\\ \cos(m+n)\phi & \sin(m+n)\phi\end{bmatrix}$

Table 7.1: The angular component basis of SO(2)-steerable kernels that satisfy the irreducible representation kernel constraint. These are constructed for various combinations of input and output field irreducible representations ψ_n.

coefficients which force most coefficients towards 0 [LW20, WC19]. In the context of G's irreducible representations, there are varying dimensional characteristics to consider. They could be 1-dimensional or 2-dimensional. Consequently, there are different possible mappings, such as those between 2-dimensional irreducible representations, those mapping 2-dimensional to 1-dimensional irreducible representations, or vice versa and those between 1-dimensional irreducible representations. Hence, one must expand the kernel as a Fourier series for each representation mapping, which one then solves to provide solutions to the kernel constraint shown in equation 7.10 [LW20, WC19]. We consider only positive radial parts. The solutions are shown below in table 7.1. A detailed derivation of the solutions by using a Fourier expansion of the kernel is shown in [WC19].

In table 7.1, note each base element is a harmonic corresponding to a singular angular frequency. By expanding the kernel into the Fourier series, we are essentially expressing the kernel as a linear combination of circular harmonics as proposed in [WGTB17]. However, they utilise complex representations and perform actions on complex feature maps. For the O(2) group, the derivations would remain largely unaffected since the real and complex irreps align, only differing by a change of basis. On the other hand, the derivations are simpler in the case of SO(2) as their irreps in complex space are only 1-dimensional. Translating the solution spaces of complex G-steerable kernels back into real values requires careful consideration to avoid an under-parameterised implementation in real space. Additionally, in harmonic networks, while the original complex solution was complete, by shifting the network's operation to the real field, we also incorporate negative frequencies into the feature fields. This permits us to employ an expanded set of 24 steerable kernels without any additional overhead on

the system.

In the creation of SO(2) equivariant networks within a computerized environment, the sampling of both the kernels and feature maps on a discrete sampling grid is necessary and can potentially lead to aliasing issues [LW20, WC19]. This demands careful handling, especially with high-frequency filters. Low spatial resolution could exacerbate these issues, particularly with smaller radii where fewer pixels per solid angle are covered near the origin. To mitigate aliasing, a radially dependent angular frequency cutoff is introduced [WC19]. Furthermore, to counterbalance the aliasing effects stemming from the radial part of the kernel basis, a smooth Gaussian radial profile is chosen [WC19].

7.3 Related work

The exploration of neural networks with equivariance to specific symmetry groups has rapidly evolved [BLV+18, CW16a, CW16b, DDFK16, HPCW18, HTP+18, KSP+18, MVKT17, RSP17, WHS18]. These models can be understood at a broader level by looking at two distinct aspects: first, the symmetry group to which they maintain equivariance, and second, the kind of geometric characteristics they utilize [CGW18].

Equivariant neural networks have shown improved performances in various computer vision tasks. For example, [WC22] demonstrated by using regular group convolutions which included the dihedral group, one can significantly improve performance for lung nodule detection on volumetric CT imaging. Specifically, their 3D G-CNN outperformed a similar baseline architecture in false positive reduction for pulmonary nodule detection, showing better sensitivity to malignant nodules and faster convergence. Harmonic networks [WGTB17] which use steerable convolutions demonstrated improved parameter efficiency while achieving SOTA classification in boundary detection performance.

Previously, rotation-equivariant UNets were limited to 2D data [CTT+19, LWV+18, PDO+22] demonstrating improved sample efficiency. [MGTC21] describes a specific 6 six-dimensional diffusion MRI data application using SE3 equivariant CNN filters

for multiple sclerosis segmentation, capturing equivariance in both voxel space and k-space which leads to improved performance. More recently, an end-to-end SE(3) equivariant segmentation model was proposed. This SE(3)-equivariant volumetric segmentation network is immune to unseen data poses during training, eliminating the need for 3D rotation-based data augmentation. They also revealed better segmentation results on MRI for both brain tumours and healthy brain structures, with limited training data and increased parameter efficiency.

Recent research reveals that human recognition relies on shape, whereas ImageNet [DDS+09] pre-trained CNNs depend on texture. To combat this, [GRM+18] created Stylized ImageNet (SIN) using the AdaIN [HB17] style transfer algorithm, which replaces natural texture with various random ones. Since a model trained on SIN can't predict results using local texture, it must consider the input's shape and structure. Experiments in [GRM+18] show that a CNN trained on SIN behaves more like human recognition, focusing on shape. This is particularly relevant to segmentation tasks where only shape information is required to perform the task. [KB20] proposed a method to learn texture invariant features to improve the domain generalisability of segmentation models. They diversify synthetic image textures using a style transfer algorithm, preventing segmentation models from overfitting to one specific texture. Then, by fine-tuning the model with self-supervised training on the target texture, they achieve SOTA performance. RandConv [XLY+20] which is perhaps the most related work, attempts to learn texture invariant features by using a randomised convolutional input layer.

7.4 Methods

7.4.1 Method 1: Contrastive Learning

Texture Invariance

Firstly, to achieve texture invariance we exploit the multiple acquisitions which are used in MRI which leads to differing textural and contrast properties among different

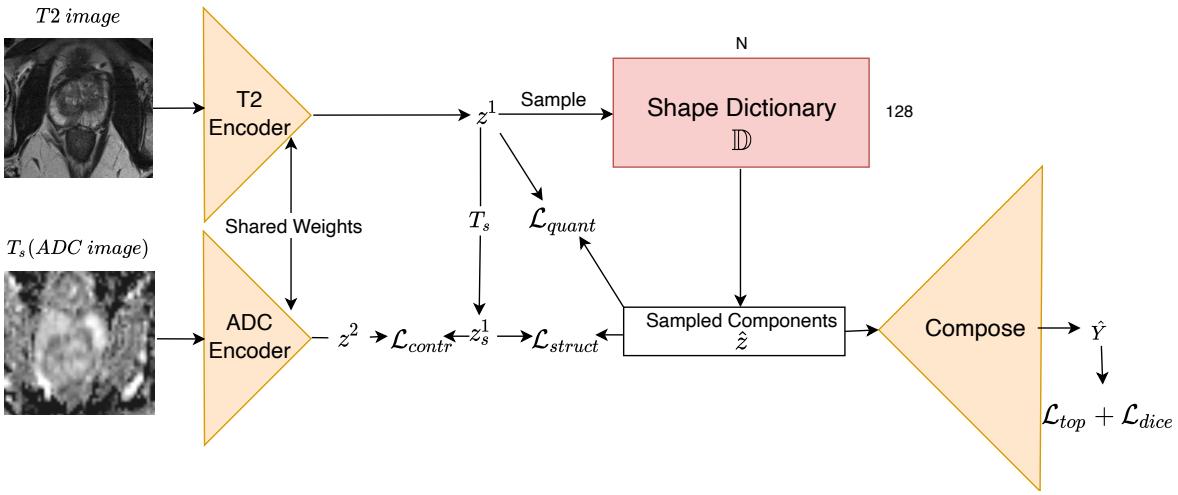


Figure 7.1: Overview of our method demonstrating using the ADC map to learn to shape equivariant features which are quantised to construct a shape dictionary, $\mathbb{D}$. We assume the dictionary is complete and sufficient to capture the entire distribution of segmentation outputs after the composition of the discrete shape space using the decoder.

sequences. Specifically, we propose a method to learn texture invariant features based on the principle that T2-weighted images and ADC maps calculated from diffusion-weighted imaging contain similar spatial information and only differ in their textural profiles. Therefore, we impose a contrastive loss such that T2-weighted and ADC imaging are mapped to the same representation. Formally, we start with the image input which is the T2 weighted image $x \in \mathbb{R}^{1\times256\times256\times24}$ and apply an intensity transformation, T_i which is equivalent to acquiring the ADC map.

In cases where there is only a single sequence available we will use a randomised convolutional layer in the first layer of our network to enforce texture invariance as implemented by [XLY+20]. The output of random filters retains relative similarities between input patches, as proven in [XLY+20]. Thus, locations with similar local textures in the input tend to remain similar in the output, meaning shapes in the output closely resemble those in the input, especially when filter sizes are small compared to typical shape sizes. Essentially, a convolution filter's size dictates the smallest shape it retains. For instance, 1x1 random convolutions act as random colour mapping, preserving single-pixel shapes, while larger filters disrupt shapes smaller than the filter, viewing them as local textures at that scale. In this work, we randomly sample uniform kernel sizes of 1,3 or 5 and we randomly sample weights for a randomised convolution layer from a Gaussian distribution ($\sim \mathcal{N}\left(0, \frac{1}{3k^2}\right)$). We mix the output of

the randomised convolutional layer denoted I_r with the original image, I as advised in [XLY+20]. Specifically, we randomly sample the mixing parameter, α from a uniform distribution($\alpha \sim U(0,1)$) to linear mix with the original image ($\alpha I + (1-\alpha)I_r$).

Shape Equivariance

We also want to impose shape equivariance in our latent before and after quantisation to construct a discrete shape equivariant latent space which we claim improves the robustness of a segmentation model.

We apply a spatial transformation, T_s to the ADC map which involves rotations. Specifically, we apply transformations from the dihedral group. In our work, we will be dealing with 3D imaging and hence we will be using 3D CNN filters. In a three-dimensional CNN, filters take the shape of cubes or cuboids rather than squares. In 3D medical imaging methods like CT and MRI, the pixel spacing along the x and y axes may differ from the z axis. Consequently, a filter-sized k × k × k might represent a space resembling a cuboid instead of a perfect cube. Therefore, in this work, we consider symmetries of the cuboid. We use rotations which maintain orientation and reflections as well as rotations which do not maintain orientation. In figure 7.2, we show the Cayley diagrams for groups D4 and D4h. In a Cayley diagram, every node represents a symmetry transformation, denoted as $h \in H$. In figure 7.2, each colour signifies the application of a specific generator transformation. By sequentially applying these generators, or tracing the diagram's edges, one can achieve any transformation within the group. We work with dihedral groups Dn which are orientation preserving of varying orders, which include D4, D6, D8, D12 and D16. For example, D4 consists of 90-degree rotations in the z plane and 180 degree rotations in the y plane shown in figure 7.2 by the red arrow and blue line respectively. The order of this group is 8 so we create 8 transformations per sample during training. D12 consists of 30 degree rotations in the z plane and 180 degree rotations in the y plane. The order of this group is therefore 24. The non-orientation preserving dihedral groups of varying orders used in our work consists of D4h, D6h, D8h, D12h and D16h. This group has the additional transformation of a reflection in the Z-axis (green line in Figure 7.2) so the order of the D4h and D12h groups is 16 and 48 respectively.

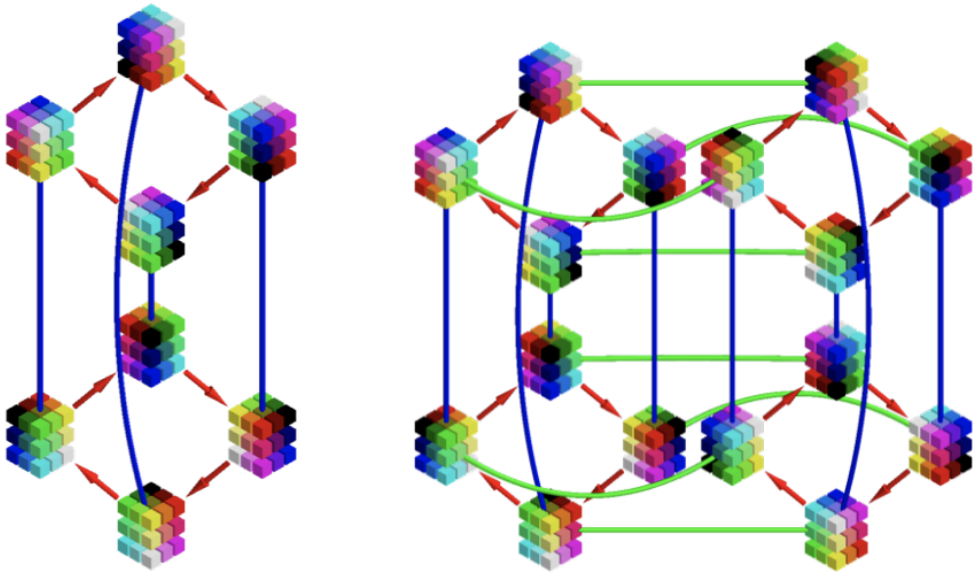


Figure 7.2: Cayley Diagram for D4(Right) and D4h(Left). This image is taken from [WC22].

Equivariance for Robustness to Domain Shifts

We next make the following assumption of a domain shift which occurs due to a change in the acquisition of the image:

Assumption 4. *Given convolutional encoder ϕ_e which is texture biased and maps image X to representation z, we assume the mutual geometric information between X and z is corrupted by an acquisition shift in X denoted X_a. In particular, we claim the equivariant relationship between X_a and z is corrupted*

The above assumption essentially claims, that shape and texture information is entangled in the neurons of the encoder and therefore a shift in textural information will affect the geometric/shape information extracted by the encoder leading to a semantic shift which is not desired [IKE+20]. This is despite the shape information remaining invariant in an acquisition shift or in other words a textural shift. In fact, local imperceptible textural change in the input space leads to significant semantic shifts which form adversarial examples [IST+19]. The semantics are often largely captured through geometric/shape information [GRM+18]. Therefore, as well as minimising the textural information in the latent space as we proposed previously, we also aim to maximise the mutual geometric/shape information between the input space and latent space. Given the output is a segmentation map with no textural information, we are also thereby

maximising the mutual information between the latent space and output space. This leads us to state our first lemma below.

Lemma 2. *By increasing the density of the equivariant constraint on the mapping between X and z we are also increasing the mutual information between Y and z ($I(Y;z)$)*

We next make the following assumptions.

Assumption 5. *We assume the decoder, ϕ_d which maps the latent space z to the output Y demonstrates improved performance that surpasses random guessing of each pixel in the output space.*

Assumption 6. *We also assume instances within our dataset are uniformly distributed across c categories.*

We impose a dense constraint on z by constructing n representation of X which are well defined where n is the order of the group. The equivariant constraint is shown in equation 2.21 in section 2.7 of chapter 2. The transformations are sampled from the matrix representations of all the group elements, $T_s \in \rho(G)$ to construct representations $z_0, z_1 ... z_n$. In other words $z_0, z_1 ... z_n$ acts as self-supervised labels for X.

We will now first form our first theorem given an acquisition shift $x_a \in X_a$. We state a theorem and prove it based on similar work by [MZJ+23] who also showed imposing equivariance constraints in the latent space improves robustness to adversarial attacks.

Theorem 5. *The prediction accuracy given the dense equivariant constraints is bounded as follows:*

$$P(Y|z_0, X_a) \in [b_0, c_0] \tag{7.13}$$

$$P(Y|z_0, z_1 X_a) \in [b_1, c_1] \tag{7.14}$$

$$P(Y|z_o, z_1, z_2, X_a) \in [b_2, c_2] \tag{7.15}$$

$$\vdots \tag{7.16}$$

$$P(Y|z_0, z_1, \ldots, z_n, X_a) \in [b_n, c_n]. \tag{7.17}$$

Therefore there is an order on the prediction accuracy bounds as follows:

$$b_0 \leq b_1 \leq \ldots \leq b_n \tag{7.18}$$

and

$$c_0 < c_1 < c_2 < \ldots < c_n. \tag{7.19}$$

This shows that we can improve the classification accuracy bounds using denser geometric constraints on z.

Proof. Given lemma 2, it follows that: Given that:

$$I(Y; z_i | X = X^a) > 0 \tag{7.20}$$

it can be deduced that:

$$\begin{aligned} I(Y; z_0, z_1, \ldots, z_n | X^a) &> I(Y; z_0, z_1, z_2 | X^a) \\ > I(Y; z_0, z_1 | X^a) &> I(Y; z_0 | X^a) = I(Y; X^a) \end{aligned} \tag{7.21}$$

In the next step of our proof, first consider an output space with c classes. Given, our prediction $\hat{Y}$, we define an error bound on our prediction accuracy as follows; $P(\hat{Y} = Y) \geq 1 - \epsilon$. Next using Fano's inequality the following equation holds.

$$H(Y|X^a) \leq H(\epsilon) + \epsilon \cdot \log(n-1) \tag{7.22}$$

Here, $H(\epsilon) = \epsilon \log \epsilon + (1-\epsilon)\log(1-\epsilon)$. We then add $H(Y)$ to both sides of the equation. We also know the mutual information between Y and X_a is defined as: $I(Y; X^a) = H(Y) - H(Y|X^a)$ to form the following equation:

$$H(Y) - \epsilon \cdot \log(n-1) \geq H(\epsilon) - I(Y; X^a) \tag{7.23}$$

We rearrange equation 7.23 to form equation 7.24 below.

$$H(\epsilon) + \epsilon \log(n-1) \leq I(Y; X^a) + H(Y) \tag{7.24}$$

We redefine the left-hand side of equation 7.24 with a new term, $J(\epsilon)$ in equation 7.25.

$$J(\epsilon) \geq -I(Y; X^a) + H(Y) \tag{7.25}$$

We next multiply both sides of equation 7.25 with the inverse function, J^{-1}.

$$\epsilon \geq J^{-1}(-I(Y; X^a) + H(Y)) \tag{7.26}$$

Equation 7.26 is rearranged to form the following equation.

$$1 - \epsilon \leq 1 - J^{-1}(-I(Y; X^a) + H(Y)) \tag{7.27}$$

Note we previously placed an error bound on our prediction accuracy as $1 - \epsilon$ and therefore can deduce the following set of equations from equation 7.27.

$$\epsilon \leq c_0 = 1 - J^{-1}(-I(Y; X^a, z_0) + H(Y)), \tag{7.28}$$

$$\epsilon \leq c_1 = 1 - J^{-1}(-I(Y; X^a, z_0, z_1) + H(Y)), \tag{7.29}$$

$$\epsilon \leq c_2 = 1 - J^{-1}(-I(Y; X^a, z_0, z_1, z_2) + H(Y)), \tag{7.30}$$

$$\ldots \epsilon \leq c_n = 1 - J^{-1}(-I(Y; X^a, z_0, z_1, z_2 \ldots z_n) + H(Y)), \tag{7.31}$$

The equations above demonstrate that the upper bounds c_i are a function of the mutual information between the latent space z and output space Y. Thus, given $H(Y)$ is a constant, the greater the mutual information between z and Y then the greater the upper bound. In other words the denser the constraint or the larger the order of our group, then the greater the upper bound ($c_0 < c_1 < c_2 < \ldots < c_n$). Simultaneously, an identical proof can be derived to show denser constraints on z can increase the lower bound and thus $b_0 \leq b_1 \leq \ldots \leq b_n$. This concludes our proof. □

The T2 image and spatially transformed ADC map are passed through an encoder to produce their respective embeddings, z^1 and z^2 as shown in figure 7.1. Shape equivariance and texture invariance are enforced by satisfying equation 7.32.

$$\Phi_e \left(\sum_{i=0}^{n} T_s^i(T_i(x)) \right) = \sum_{i=0}^{n} T_s(z) \tag{7.32}$$

Therefore, we minimise the contrastive loss: $\mathcal{L}_{contr} = ||T_s(z^1) - z^2||_2^2$. Note, a contrastive loss only theoretically learns equivariance to the spatial transformations applied per sample. It does not constrain the convolutional layers to the group. We assume an approximate equivariance to the group by using our contrastive loss.

Quantisation

We quantise $z^1 \in \mathbb{R}^{256\times16\times16\times12}$ and $z^1 \in \mathbb{R}^{256\times18\times18\times12}$ for the prostate dataset and cardiac datasets respectively using vector quantisation. This is carried out by dividing z^1 into 16×16×12 and 18×18×12 components for the prostate and cardiac datasets respectively and replacing each component in z^1 denoted z_i^1 with its nearest component $e_k \in \mathbb{D}$ where $k = argmin_j ||z_i^1 - e_j||_2$. This produces the discrete shape latent space $\hat{z}$ which is inputted into the decoder to construct the segmentation output. The quantisation loss minimises the Euclidean distance between z_i^1 and its nearest component $e_k \in \mathbb{D}$ shown in equation 7.33 which is similar to the quantisation loss component of equation 5.2 with slight variable change in chapter 5.

$$\mathcal{L}_{Quant} = \frac{1}{m}\sum_{i=0}^{m-1} ||sg(z_i^1) - e_k||_2 + \beta||z_i^1 - sg(e_k)||_2 \tag{7.33}$$

We compute the Dice loss between the output, $\hat{y}$ and the T2 segmentation label y. Only T2-weighted images are required as input during inference.

During quantisation, the geometric structure in z_1 may be partially lost after quantisation. This is because it is quite feasible that a large proportion of the elements in z_1 is mapped to the same codebook vector during quantisation which leads to a many-to-one mapping between spatial elements in z^1 and $\hat{z}$. This is known as codebook collapse and a well-known phenomenon in VQ-VAE [BCCS23] which leads to invariant rather than desired equivariant properties. Therefore, to preserve geometry after quantisation, we impose a structure-preserving loss shown in equation 7.34. This loss aims to preserve the distance between the neighbouring elements $m_i, m_{i+1} \in z^1$ which are of dimension, $\mathbb{R}^{128\times1\times1\times1}$ after mapping to elements $n_i, n_{i+1} \in \hat{z}$ via quantisation ($n \in \mathbb{R}^{128\times1\times1\times1}$). This

loss is applied to all $16 \times 16 \times 12$ elements mapped from z^1 to $\hat{z}$

$$\mathcal{L}_{Struct} = \sum_{i=0}^{127} ||m_i.m_{i+1} - n_i.n_{i+1}||_2 \tag{7.34}$$

The total loss for training our framework is $\mathcal{L}_{total} = \mathcal{L}_{Dice}(\hat{y}, y) + \mathcal{L}_{contr} + \mathcal{L}_{Quant} + \mathcal{L}_{Struct}$.

7.4.2 Method 2: Group Equivariant Convolutional Kernels

In our previous method, we used a 2D/3D hybrid UNet to learn equivariance in the discrete latent space to the 3D dihedral groups using a contrastive method. However, in our second method, we constrain the convolutional kernels to group equivariant convolutional kernels of particular groups. Therefore we cannot use a hybrid network but either a 2D or 3D UNet. This is because imposing 2D equivariant constraints on the convolutional kernels in the initial layers of the network followed by 3D equivariant convolutions, will not make the network equivariant to the 3D group but only the 2D groups provided the 2D group is a subgroup of the 3D group. Therefore we use a 2D UNet to impose 2D group equivariant constraints due to the highly anisotropic nature of cardiac and prostate MRI images. We impose texture invariance in the latent space via a constrastive loss ($\mathcal{L}_{contr} = ||(z^1) - z^2||_2^2$) for the prostate dataset via the representations learnt for the T2-weighted images (z^1) and ADC images (z^2) and an initial randomised convolutional layer for the cardiac dataset. Therefore the method is similar to the contrastive method shown in figure 7.1 except here we remove the spatial equivariant constrastive loss and replace all convolutional kernels with group equivariant convolutional kernels.

Dihedral Group

We first describe the implementation of the group equivariant convolutions for the dihedral group. In the initial layer, the input feature maps and filters do not have orientation channels. Given n_0 input channels and n_1 filters for each of the n_0 input channels, we modify each of the n_1 filters using every transformation $g \in G$. This

creates an expanded filter bank consisting of $n_1 \times |G|$ filters. Therefore each of the n_0 channels is transformed by all the filters in our expanded filter bank.

In the second and higher-order layers, we then reshuffle our orientation channels. In the transformation phase, each filter undergoes modification by each $g \in G$, resulting in $n_{l+1} \times |G|$ adjusted filters which are then applied to each input channel to create as many output channels. The transformations used for the 2D group convolutions are sampled from the 2D dihedral groups which consist of rotations in the 2D plane and reflections in the Y axis. Therefore, the order of the 2D D4 group is 8. The point groups such as D4h do not apply to 2D group convolutions and therefore not used in this method.

SO(2) Group

We will also consider the SO(2) group in terms of its cyclic subgroup $Cn \subseteq SO(2)$. We therefore constrain the convolutional kernels in the 2D UNet to either the C4, C6, C8 or C16 cyclical groups. We also use irreducible representation in the form of circular harmonics for the implementation of SO(2) group equivariant convolutions. As previously described, SO(2) can be decomposed into irreducible representations (irreps). We use the escnn library devised by [WC19] to build our SO(2) equivariant kernels. We draw inspiration from harmonic networks and build SO(2) equivariant convolutional layers up to an order of 3; this is the first 3 irreducible representation of SO(2) which contains cosines and sines of frequency 0,1 and 2. Therefore, a convolutional layer is computed as the direct sum of the irreducible representations up to order 3 ($\oplus_{i=0}^{L} \psi_i$). T We use a multiplicity of 1 for each field representation. The feature fields of the SO(2)-equivariant layer are constrained to transform under irreducible representations shown in table 7.1.

Every model utilizes ELUs for scalar fields and employs norm-ReLUs for higher-order fields. We followed [WHS18] and experimented with applying a low-pass filter before performing the down-sampling step. This can be implemented either through an additional strided convolution using a Gaussian kernel. The decoding path utilizes trilinear or bilinear upsampling, an inherently equivariant operation. Standard instance normalization [UVL16] is applied to scalar features. For vector or tensor-valued fea-

tures, instance normalization is carried out by dividing by the average l_2 norm for each instance, expressed as: $\text{norm}(v) := \frac{v}{E(\|v\|)}$. In each convolutional block for each level of the encoder, we use a total of 16, 32, 64, 128 and 256 filters for levels 1, 2, 3, 4 and 5.

The SO(2) group equivariant constraint is the densest constraint imposed on the network as here we are technically imposing infinite constraints in the latent space z_i. Therefore, if we follow theorem 5 and its corresponding proof 5, this should in theory be the most robust model.

7.5 Experiments

Our experiment assesses the single domain generalisability (SDG) of various models. We evaluate segmentation performance when testing a trained segmentation model on an unseen target domain for the prostate and cardiac datasets. We adopt a cross-validation procedure by training a method on a single source domain and holding out the other domains for testing (3 for cardiac and 1 for prostate).

7.5.1 Datasets and Training

We use 2 datasets in our experiments.

Cardiac: This dataset is the M&Ms cardiac imaging 3-class segmentation dataset [CGI+21] which is divided into 4 domains determined by the MRI scanner vendor. All images were normalised between 0 and 1 and cropped to $288 \times 288 \times 12$ for 3D input in method 1 and 288×288 for 2D input in method 2.

Prostate: We use the NCI-ISBI13 Challenge [BMH+15b] dataset. To evaluate the single-domain generalizability of prostate zonal segmentation, we utilize two datasets, each with two labels representing the transitional and peripheral zones. The training set comprises 32 T2-weighted and ADC images from the prostate dataset in the Medical Segmentation Decathlon. These images were acquired from the Radboud University Nijmegen Medical Centre (RUNMC) [ARB+22].

For the test set, we employ 30 T2-weighted scans obtained from the Boston Medical Centre (BMC) as part of the NCI-ISBI13 Challenge [BMH+15b, CVS+13, BMH+15a]. We normalised images between 0 and 1 and centre cropped to $256 \times 256 \times 12$ for 3D input in method 1 and 256×256 for 2D input in method 2. The T2 weighted and ADC images are pre-registered.

Training: All models are trained with Adam optimisation [KB14] with a base learning rate of 0.0001 and weight decay of 0.05 for a maximum of 500 epochs on three NVIDIA RTX 2080 GPUs. To prevent over-fitting, we apply a simple augmentation scheme consisting of random rotation and flipping (horizontal and vertical) for training baseline models where our equivariant methods are not incorporated into the model. No augmentation is applied in our group equivariant models. We evaluate model performance with the Dice score and maximum Hausdorff distance (HD) in mm.

7.5.2 Method 1 Experiments

Experimental Setup

In the first set of experiments, we first explore method 1 applied to the cardiac and prostate datasets. We use a hybrid 2D/3D UNet as our baseline model to deal with the anisotropic prostate and cardiac MRI images. The encoder and decoder are made up of 5 levels consisting of 2D pre-activation residual blocks in the top 4 levels and a 3D pre-activation residual block in the bottleneck level. The pre-activation residual blocks incorporate group normalisation (number of groups = 8) and ReLU activation. The first level outputs 16 features (channels) and doubles at each level until we have 256 features in the bottleneck. We use 2D max pooling for downsampling and bi-linear interpolation for upsampling operations. We use the same encoder and decoder architecture for our approach with various geometric constraints. We compare our method to the following SDG methods applied to the baseline model: CutOut [DT17], BigAug [ZWY+20], AdvBias [CQQ+20] and RandConv [XLY+20]. In the prostate dataset, we have T2 weighted and ADC images and therefore use a contrastive method to impose texture invariance as shown in figure 7.1. In the cardiac dataset, we only have T2-weighted imaging and therefore we incorporate a randomised convolutional layer as

our initial layer to impose texture invariance.

Group Order Results

	Baseline	D4	D6	D8	D12	D16	D4h	D6h	D8h	D12h	D16h
						Transitional Zone					
Dice	0.64± 0.13	0.69± 0.08	0.70± 0.17	0.71± 0.11	0.72± 0.13	0.72± 0.10	0.68± 0.15	0.69± 0.14	**0.73± 0.13**	0.72± 0.14	0.71± 0.12
HD	24.33± 14.12	20.14± 3.55	18.93± 3.94	18.41± 4.50	19.21± 4.80	19.01± 4.34	19.10± 4.11	19.20± 5.30	**18.90± 4.45**	18.19± 5.50	19.07± 4.61
						Peripheral Zone					
Dice	0.41± 0.09	0.47± 0.12	0.49± 0.15	0.51± 0.10	0.52± 0.11	0.54± 0.09	0.48± 0.07	0.51± 0.10	0.53± 0.09	**0.55± 0.09**	0.53± 0.17
HD	41.88± 18.72	34.94± 15.54	31.74± 17.90	29.28± 14.72	29.90± 17.22	30.82± 13.99	32.62± 15.72	30.93± 12.95	**28.92± 9.92**	29.54± 12.83	29.10± 13.12

Table 7.2: Dice score and Hausdorff distance(HD) ± standard deviation for different order dihedral group shape equivariant models trained via contrastive learning approach. This table shows the results for the prostate dataset.

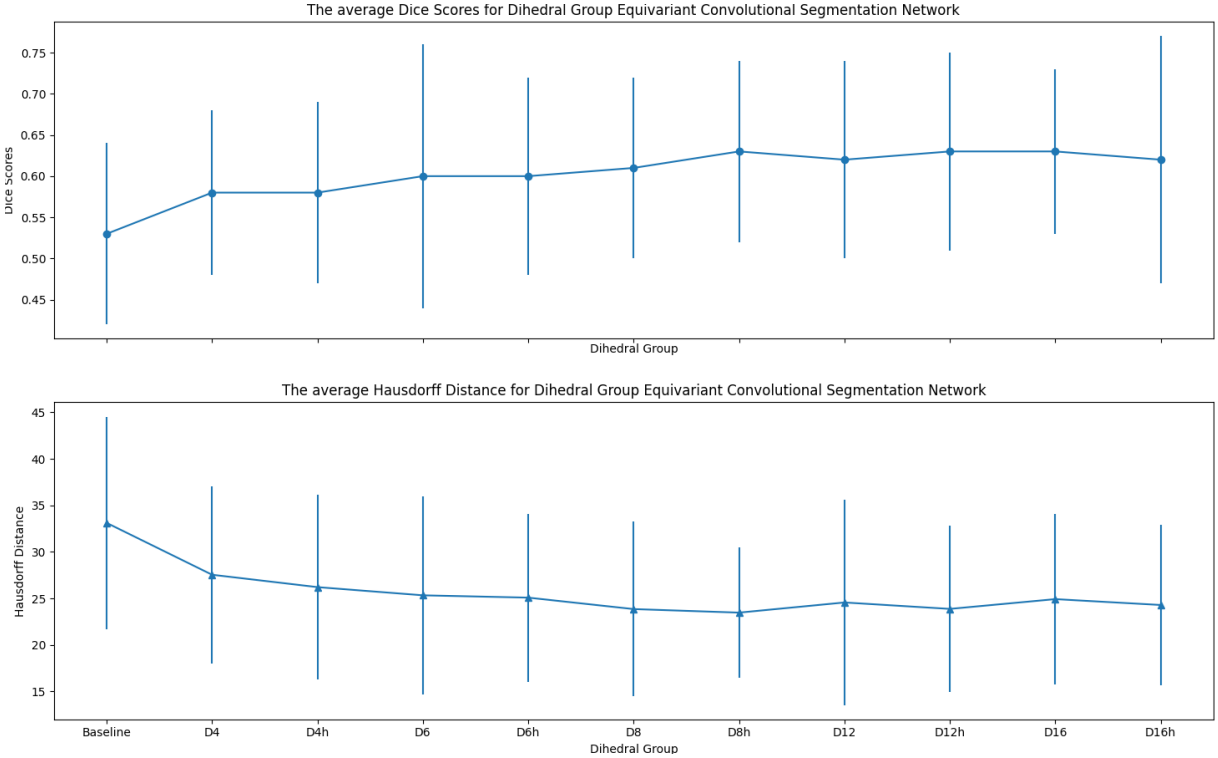


Figure 7.3: The average Dice and maximum Hausdorff distance scores ± standard deviation for the respective dihedral group equivariant CNNs trained via contrastive learning and applied to the prostate dataset. The scores are averaged across all classes.

First, we explore the different order dihedral group constraints imposed on our models. According to theorem 5, the denser the constraint or the higher the group order then the more robust the model. Therefore the D16h group equivariant model should yield the best performance.

	Baseline	D4	D6	D8	D12	D16	D4h	D6h	D8h	D12h	D16h
						Left Ventricle					
Dice	0.74± 0.13	0.78± 0.15	0.79± 0.12	0.80± 0.09	0.82± 0.13	0.80± 0.09	0.79± 0.12	0.80± 0.17	0.81± 0.11	**0.82**± **0.14**	0.81± 0.12
HD	15.33± 3.1	12.61± 2.75	11.13± 2.58	10.74± 3.79	10.91± 2.99	10.14 ± 3.13	11.73± 3.98	10.18± 3.24	**9.93**± **3.08**	10.72± 3.07	10.16± 3.82
						Right Ventricle					
Dice	0.72± 0.16	0.75± 0.20	0.77± 0.21	0.79± 0.17	0.78± 0.16	0.79± 0.14	0.74± 0.17	0.77± 0.10	**0.79**± **0.15**	0.78± 0.18	0.77± 0.15
HD	18.76± 4.43	14.18± 3.74	13.55± 2.92	14.01± 4.22	12.91± 3.08	12.93± 4.11	15.08± 3.00	14.49± 4.65	13.03± 2.87	12.82± 2.90	**12.11**± **3.04**
						Myocardium					
Dice	0.57± 0.21	0.63± 0.17	0.64± 0.15	0.62± 0.19	**0.66**± **0.11**	0.65± 0.13	0.62± 0.15	0.61± 0.22	0.64± 0.18	0.65± 0.13	0.64± 0.13
HD	26.55± 5.21	20.19± 4.87	20.48± 4.09	19.27± 5.11	18.87± 4.76	18.89± 5.03	21.10± 4.37	20.87± 4.88	18.29± 4.45	**18.11**± **3.91**	19.05± 5.04

Table 7.3: Dice score and Hausdorff distance(HD) ± standard deviation for different order dihedral group shape equivariant models trained via contrastive learning. This table shows the results for the cardiac dataset.

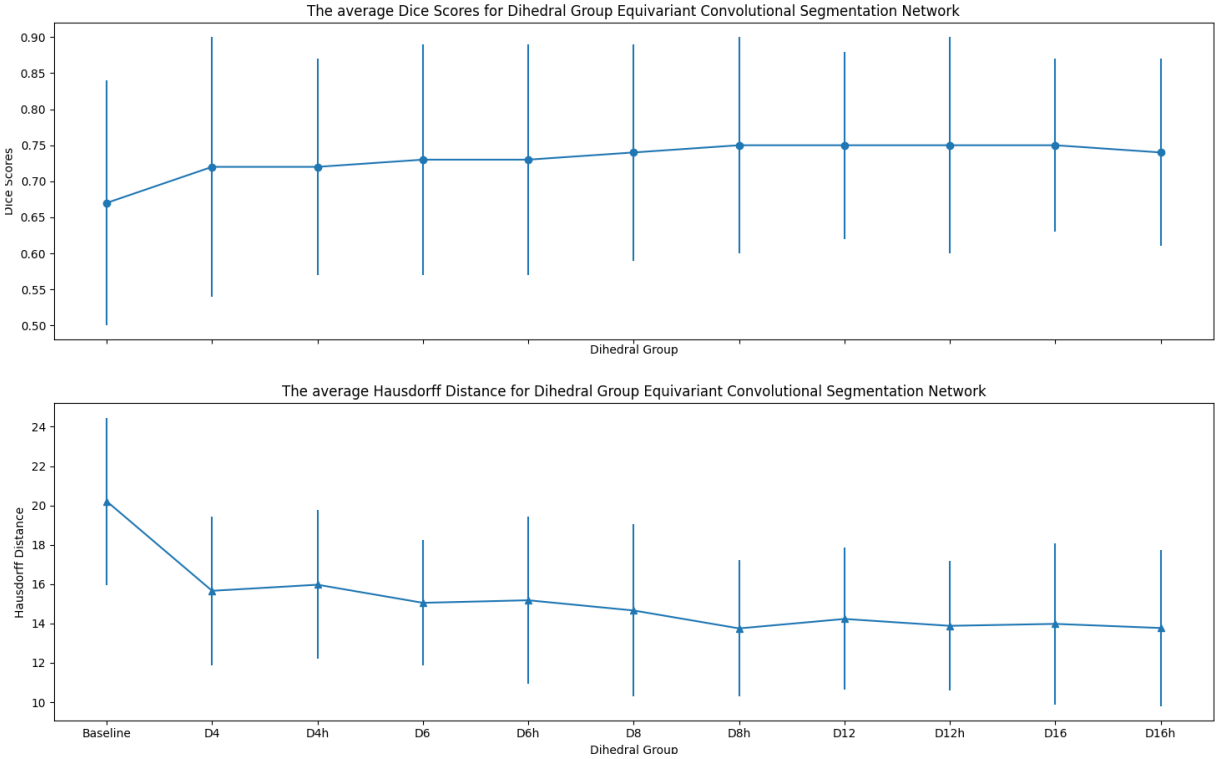


Figure 7.4: The average Dice and maximum Hausdorff distance scores ± standard deviation for the respective dihedral group equivariant CNNs trained via contrastive learning and applied to the cardiac dataset. The scores are averaged across all classes.

In table 7.2 and 7.3, we show all the Dice scores and maximum Hausdorff distances for each model trained with different order group equivariant constraints in the latent space. As previously mentioned, here group equivariance is learnt with contrastive learning. We note on average the D8h group equivariant model achieved the best Dice and maximum Hausdorff distance score followed by the D12h group equivariant model for both the cardiac and prostate datasets. For example, the models equipped with D8h

and D12h group equivariance in the latent space both achieved an average Dice score of 0.68 for the prostate dataset. In figure 7.3 and 7.4, we note that as we increase the density of the constraint in the latent space (the order of group increases) there is an overall trend upwards in terms of both the Dice score and maximum Hausdorff distance for the prostate and cardiac datasets which supports our theorem 5. However, the improvement in the metric scores appears to diminish as the order of the group increases. For example, the Dice scores were 0.75, 0.75, 0.75 and 0.74 for the D12, D12h, D16 and D16h groups respectively in the experiments using the cardiac dataset. This result may arise because by learning equivariance to the lower order groups which are subgroups of the higher order groups, one has already sufficiently learnt equivariance to higher order groups. This means group equivariant constraints provide increased robustness up to a certain order.

SDG Results

	Baseline	D8h	D12h	CutOut	BigAug	AdvBias	RandConv	Jigen
Transitional Zone								
Dice	0.64±0.13	**0.73±0.13**	0.72±0.14	0.65±0.20	0.72±0.15	0.67±0.17	0.69±0.14	0.67±0.18
HD	24.33±4.12	**18.01± 4.08**	18.19±5.00	23.55±4.45	19.81±5.03	21.17±3.98	20.98±4.29	22.98±5.68
Peripheral Zone								
Dice	0.41±0.09	0.53±0.09	**0.55± 0.09**	0.41±0.17	0.53±0.16	0.45±0.14	0.49±0.16	0.41±0.12
HD	56.88±18.72	**28.92± 9.92**	29.54±12.83	51.06±16.17	30.94±14.88	44.72±16.83	38.10±12.88	49.87±17.92

Table 7.4: The average Dice and maximum Hausdorff distance score ± standard deviation comparing the D8h and D12h group shape equivariant segmentation models to various other SDG methods. This table shows the results for the prostate dataset.

	Baseline	D8h	D12h	CutOut	BigAug	AdvBias	RandConv	Jigen
Left Ventricle								
Dice	0.74±0.13	0.80±0.17	**0.81±0.11**	0.75±0.13	0.78±0.19	0.76±0.11	0.77±0.18	0.75±0.18
HD	15.31±3.1	**9.95±3.08**	10.73±3.07	14.81±4.11	11.11±2.99	12.99 ±3.88	12.54±3.48	13.99±4.06
Right Ventricle								
Dice	0.72±0.16	**0.79±0.15**	0.78±0.18	0.72±0.14	0.77±0.13	0.74±0.17	0.75±0.13	0.73±0.18
HD	18.76±4.43	13.03±2.87	**12.82± 2.90**	17.05±3.88	14.08±3.34	15.82±4.62	14.19±3.29	16.91±4.50
Myocardium								
Dice	0.57±0.21	0.64±0.18	**0.65±0.13**	0.59±0.14	0.63±0.18	0.61±0.19	0.61±0.10	0.60±0.19
HD	26.55±5.21	18.29±4.45	**18.11± 3.91**	24.88±5.78	20.08±5.19	22.53±4.58	22.08±4.16	23.76±4.74

Table 7.5: The average Dice and maximum Hausdorff distance scores ± standard deviation comparing the D8h and D12h group shape equivariant segmentation models trained via contrastive learning to various other SDG methods. This table shows the results for the cardiac dataset.

Next, we compare the D8h and D12h equivariant segmentation models trained via contrastive learning to several established SDG methods in the literature. Table 7.4 and 7.5 showcases that our technique which includes models trained with D8h or D12h

equivariant constraints in the latent space, outperforms other SDG methods. This is shown in terms of both the metric scores used for the cardiac and prostate datasets. The notably better maximum HD scores from our method suggest its enhanced ability to yield topologically accurate segmentations. Thus, even without resorting to aggressive augmentation tactics or adversarial training, simply embedding equivariant constraints in a segmentation model can lead to equivalent or superior SDG results. The efficacy of our approach is visually emphasized in figures 7.5 and 7.6, where the equivariant models seem to generate more anatomically accurate prostate and cardiac segmentation maps.

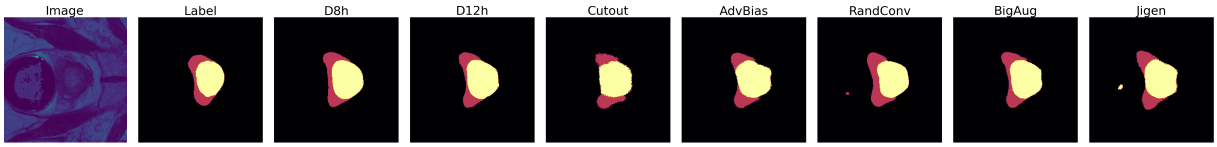


Figure 7.5: This figure shows an example prostate segmentation output comparing the D8h and D12h equivariant segmentation model trained using contrastive learning with the 5 other SDG methods.

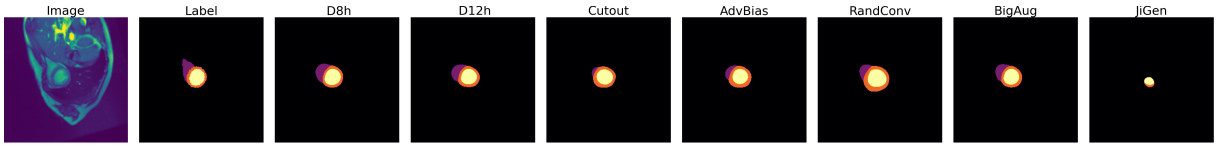


Figure 7.6: This figure shows an example cardiac segmentation output comparing the D8h and D12h equivariant segmentation model trained using contrastive learning with the 5 other SDG methods

7.5.3 Method 2 Experiments

Experimental Setup

The baseline model is a 2D UNet. This model consists of 4 levels with each level of the encoder and decoder consisting of a single 2D pre-activation residual block. The pre-activation residual blocks incorporate group normalisation (number of groups = 8) and ReLU activation for the cyclical and dihedral group equivariant convolutional kernel. The kernels with irreducible SO(2) group equivariant constraints use normalisation and activation functions as prescribed in section 7.4.2 The first level outputs 32 features (channels) and doubles at each level until we have 256 features in the bottleneck.

We use the same training setup as for the method 1 experiment and compare it to the same SDG methods as in the method 1 experiment. However, here we apply convolutional kernel constraint to the cyclical and SO(2) groups instead of the point groups.

Group Order Experiments

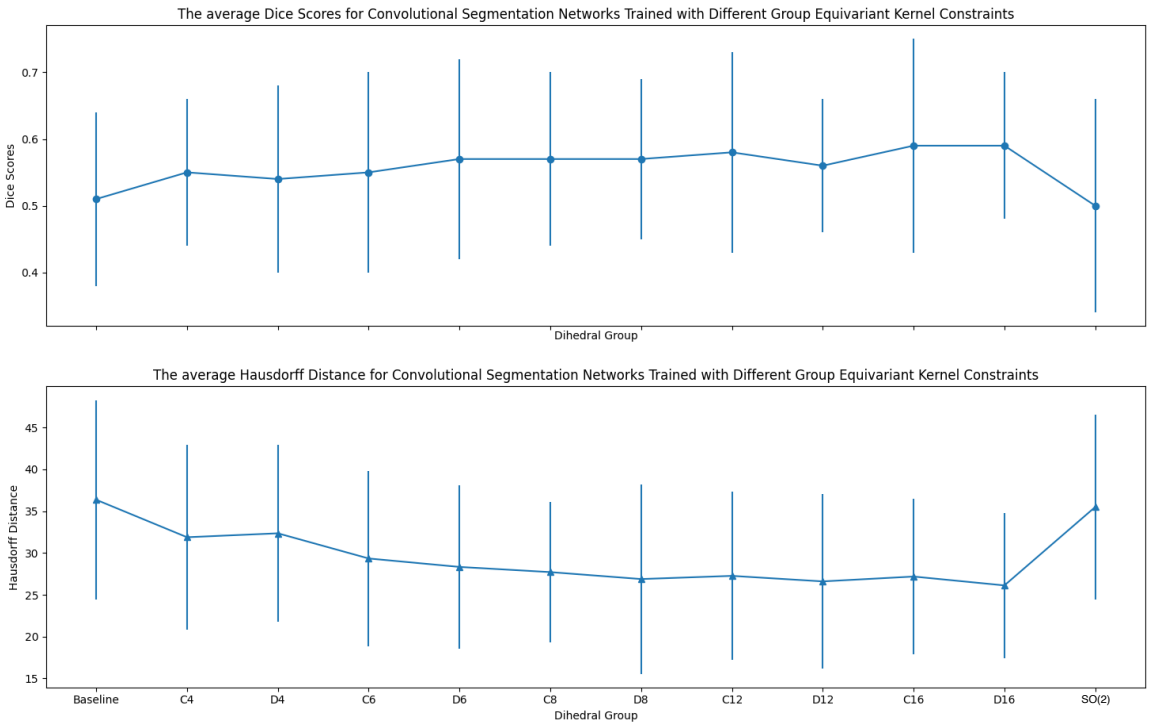


Figure 7.7: The average Dice and maximum Hausdorff distance score ± standard deviation for the respective group equivariant CNNs trained via kernel constraints and applied to the prostate dataset. The density of the equivariant constraints imposed on the kernel increases from C4 to SO(2) (left to right). The scores are averaged across all classes.

	Baseline	**D4**	**D6**	**D8**	**D12**	**D16**	**C4**	**C6**	**C8**	**C12**	**C16**	**SO(2)**
						Transitional Zone						
Dice	0.63± 0.16	0.66± 0.10	0.67± 0.14	0.70± 0.08	0.68± 0.11	**0.70± 0.08**	0.66± 0.13	0.65± 0.19	0.68± 0.11	0.68± 0.11	0.69± 0.15	0.63± 0.18
HD	27.18± 5.86	23.88± 4.75	21.46± 4.29	20.62± 4.82	20.71± 4.59	**20.18± 4.08**	24.47± 3.93	22.29± 4.23	21.83± 4.18	21.14± 3.69	21.36± 4.33	26.44± 5.26
						Peripheral Zone						
Dice	0.38± 0.10	0.42± 0.17	0.46± 0.15	0.44± 0.15	0.43± 0.08	0.47± 0.14	0.43± 0.08	0.44± 0.11	0.46± 0.15	0.48± 0.18	**0.49± 0.17**	0.37± 0.13
HD	45.53± 17.98	40.81± 16.49	35.39± 15.19	33.14± 17.82	32.47± 16.32	**32.06± 13.26**	39.29± 18.23	36.38± 16.74	33.58± 12.62	33.37± 16.43	32.99± 14.34	44.56± 16.85

Table 7.6: The average Dice and maximum Hausdorff distance score ± standard deviation for different order dihedral group shape equivariant models trained via the kernel constraint method. This table shows the results for the prostate dataset.

	Baseline	D4	D6	D8	D12	D16	C4	C6	C8	C12	C16	S02
						Left Ventricle						
Dice	0.71± 0.09	0.75± 0.13	0.76± 0.17	0.76± 0.09	0.77± 0.12	**0.78± 0.09**	0.73± 0.12	0.73± 0.17	0.76± 0.11	0.76± 0.10	0.77± 0.09	0.73± 0.16
HD	17.01± 4.09	15.09± 3.38	14.38± 2.94	12.97± 2.12	12.44± 1.98	12.03± 2.49	14.90± 3.15	15.04± 3.47	**13.49± 3.78**	13.41± 2.49	12.73± 3.31	15.89± 4.10
						Right Ventricle						
Dice	0.70± 0.12	0.72± 0.15	0.73± 0.18	0.75± 0.10	0.75± 0.13	**0.77± 0.12**	0.72± 0.11	0.72± 0.11	0.74± 0.16	0.73± 0.16	0.75± 0.09	0.70± 0.14
HD	19.87± 4.63	16.70± 4.08	15.90± 3.43	15.18± 3.89	14.29± 3.15	**13.27± 3.28**	17.91± 4.10	15.98± 3.57	15.19± 3.29	14.05± 3.08	13.37± 2.97	18.66± 4.19
						Myocardium						
Dice	0.57± 0.16	0.61± 0.14	0.63± 0.16	0.62± 0.15	0.63± 0.14	**0.65± 0.13**	0.59± 0.12	0.60± 0.19	0.64± 0.11	0.62± 0.09	0.64± 0.13	0.58± 0.15
HD	28.16± 4.88	23.19± 4.87	22.76± 4.28	20.81± 3.89	20.09± 3.39	**19.88± 3.27**	24.83± 3.86	23.28± 3.41	21.18± 4.04	21.86± 3.38	20.14± 4.29	26.29± 4.38

Table 7.7: The average Dice and maximum Hausdorff distance score ± standard deviation for different order dihedral group shape equivariant models trained via the kernel constraint method. This table shows the results for the cardiac dataset.

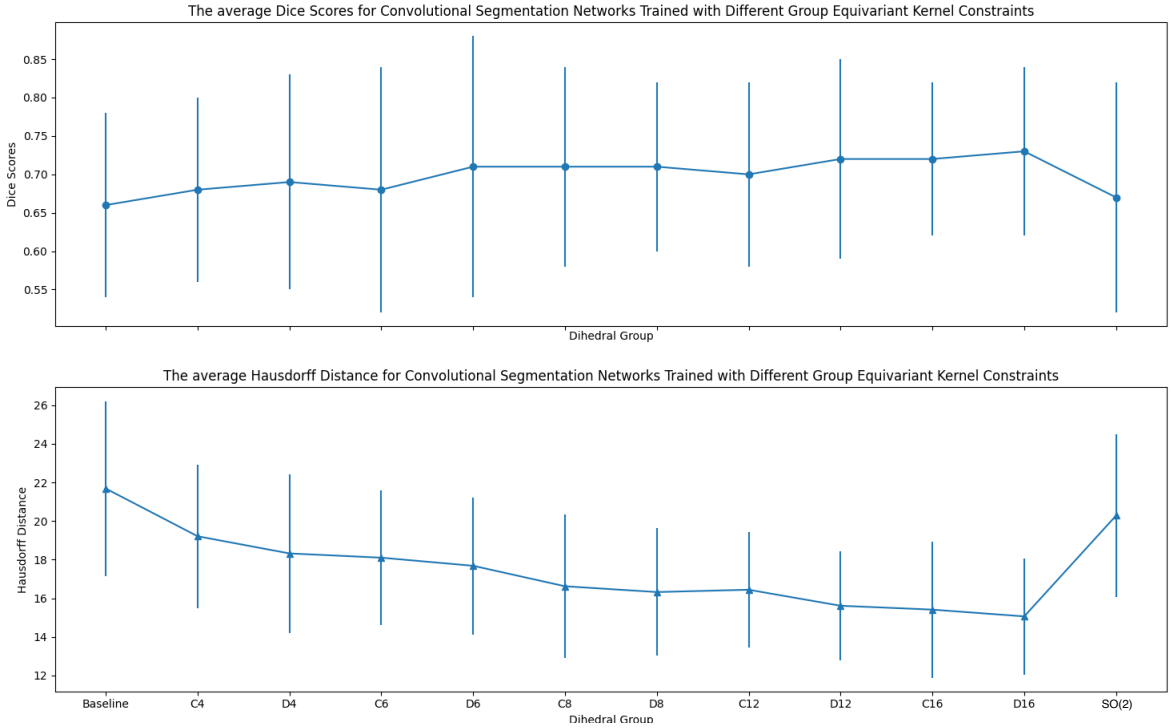


Figure 7.8: The average Dice and maximum Hausdorff distance score ± standard deviation for the respective group equivariant CNNS trained via kernel constraints and applied to the cardiac dataset. The density of the equivariant constraints imposed on the kernel increases from C4 to SO(2) (left to right). The scores are averaged across all classes.

The Dice and maximum HD scores for the different order group equivariant segmentation models are shown in table 7.6 (prostate dataset) and 7.7 (cardiac dataset).

In this set of experiments, the D16 equivariant segmentation model demonstrated the best segmentation performance for both the cardiac and prostate datasets. This is fol-

lowed by the C16 equivariant segmentation model. This is noted for all the classes in the cardiac and prostate datasets. Similar to our contrastive approach, we note the segmentation performance improves as we increase the density of the equivariant constraints imposed on the kernel from C4 to C16 (left to right) shown in figure 7.7 for the prostate dataset and figure 7.8 for the cardiac dataset. However, we note a significant drop in performance for the SO(2) group equivariant model for both the prostate (table 7.6) and cardiac (table 7.7) dataset. This may arise due to the limited expressiveness of SO(2) equivariant kernels which creates an overly constrained network. This empirically shows the equivariant constraints imposed on a segmentation network to improve robustness have an upper bound which does not fully satisfy theorem 5. Interestingly, we demonstrate superior performance in our first method where we impose shape equivariance in the latent space via a constrastive loss. This may arise because the baseline model in method 1 is a hybrid UNet which demonstrates superior performance to the baseline 2D UNet in method 2. The Dice score averaged across all three classes for the cardiac dataset was 0.67 for the hybrid UNet compared to 0.66 for the 2D UNet. The observed findings may also arise from the stricter constraints imposed in method 2.

SDG Results

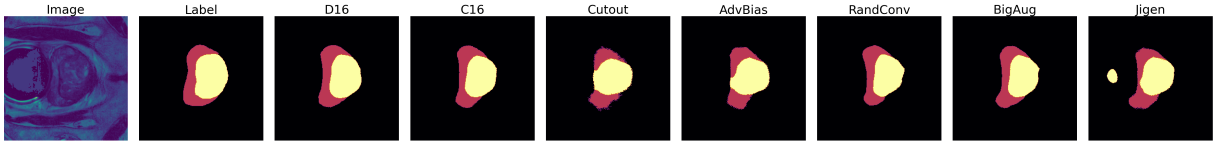


Figure 7.9: This figure shows an example prostate segmentation output comparing the C16 and D16 shape equivariant segmentation model trained via the kernel constraint method with 5 other SDG methods.

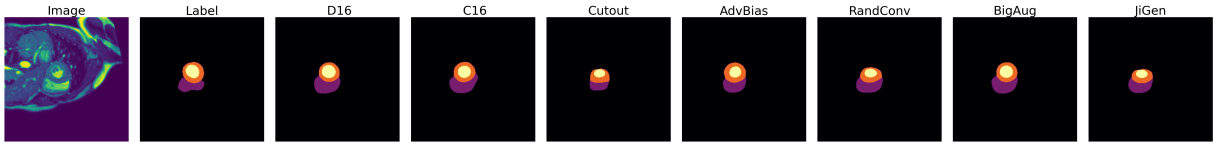


Figure 7.10: This figure shows an example cardiac segmentation output comparing the C16 and D16 shape equivariant segmentation model trained via the kernel constraint method with 5 other SDG methods.

Finally, we compare the C16 and D16 equivariant segmentation models trained with kernel constraints to several established SDG methods in the literature. Table 7.8 and

7.9 highlight that group equivariant constraints imposed in the kernels of a CNN-based segmentation model, perform competitively with other SDG methods for both the cardiac and prostate datasets. Specifically, we demonstrate that our approach outperforms other SDG methods including aggressive augmentation strategies in terms of the maximum HD score which was also shown in our contrastive equivariant method. Once again, our findings are visually emphasised with a prostate segmentation example in figure 7.5 and a cardiac segmentation example in 7.6.

	Baseline	D16	C16	CutOut	BigAug	AdvBias	RandConv	Jigen
Transitional Zone								
Dice	0.63±0.10	**0.70±0.08**	0.69±0.11	0.63±0.15	0.68±0.14	0.64±0.10	0.67±0.17	0.63±0.16
HD	27.18±5.86	**20.18± 4.08**	21.36±4.33	25.91±5.17	21.28±4.88	23.43±3.77	22.83±4.06	24.90±5.53
Peripheral Zone								
Dice	0.38±0.10	0.47±0.14	**0.49± 0.17**	0.40±0.15	0.46±0.11	0.43±0.18	0.45±0.17	0.41±0.10
HD	45.33±17.98	**32.06± 13.26**	32.99±14.34	42.81±12.86	35.38±14.21	37.83±15.05	35.33±15.72	41.74±16.02

Table 7.8: The average Dice and maximum Hausdorff distance scores ± standard deviation comparing the D16 and C16 group shape equivariant segmentation models trained via the kernel constraint method to various other SDG methods. This table shows the results for the prostate dataset.

	Baseline	D16	C16	CutOut	BigAug	AdvBias	RandConv	Jigen
Left Ventricle								
Dice	0.71±0.09	0.78±0.09	0.77±0.09	0.73±0.10	0.77±0.13	0.74±0.08	0.76±0.17	0.73±0.14
HD	17.01±4.09	**12.03±2.49**	12.73±3.31	16.18±4.37	14.06±3.37	15.51 ±4.19	14.90±3.70	16.20±4.18
Right Ventricle								
Dice	0.70±0.12	**0.77±0.12**	0.75± 0.09	0.70±0.21	0.76±0.10	0.73±0.19	0.75±0.14	0.71±0.11
HD	19.87±4.63	**13.27±3.28**	13.37±2.97	18.84±5.39	15.03±4.17	17.14±3.88	16.48±4.35	17.99±5.38
Myocardium								
Dice	0.57±0.16	**0.65±0.13**	0.64±0.13	0.59±0.18	0.63±0.20	0.60±0.21	0.61±0.16	0.58±0.14
HD	28.16±4.88	**19.88±3.27**	20.14±4.29	26.09±5.98	22.38±4.45	24.56±5.06	23.70±4.47	25.90±5.86

Table 7.9: The average Dice and maximum Hausdorff distance scores ± standard deviation comparing the D16 and C16 group equivariant segmentation models trained via the kernel constraint method to various other SDG methods. This table shows the results for the cardiac dataset.

Ablation Experiments

We carried out ablation experiments to emphasise the importance of each component of our method. We specifically show the value of enforcing a discrete latent space, followed by a texture invariant discrete latent space and finally a discrete latent space which is both texture invariant and shape equivariant. Ablations are carried out for the cardiac dataset. The baseline model here is a simple 2D UNet as previously described. We enforce equivariance to the D16 group when adding shape equivariance to our

	Baseline	Discrete Representation	Texture Invariant Discrete Representation	D16 Equivariant/ Texture Invariant Discrete Representation (Contrastive)	D16 Equivariant/ Texture Invariant Discrete Representation (Kernel Constraint)
			Left Ventricle		
Dice	0.71±0.09	0.73±0.14	0.75±0.18	**0.78±0.11**	**0.78±0.08**
HD	17.01±4.09	16.00±3.45	14.81±3.19	12.98±2.92	**12.03±2.49**
			Right Ventricle		
Dice	0.70±0.12	0.72±0.18	0.75±0.06	0.77±0.10	**0.78±0.09**
HD	19.87±4.63	16.18±4.04	15.49±3.31	14.02±4.00	**13.27±3.28**
			Myocardium		
Dice	0.57±0.16	0.60±0.10	0.62±0.17	**0.66±0.16**	0.65±0.08
HD	28.16±4.88	23.21±5.02	21.44±4.36	**19.05±4.04**	19.88±3.84

Table 7.10: The average Dice and maximum Hausdorff distance scores ± standard deviation demonstrating each component of our method. This table shows the results for the cardiac dataset.

model. This is applied with either a contrastive method or group equivariant convolutional kernels. Texture invariance in enforced with an initial randomised convolutional layer.

In table 7.10 we show that each component of our method appears to incrementally improve segmentation performance in terms of the Dice score and maximum Hausdorff distance for all three classes. Specifically, we note the improvement in segmentation performance by first creating a discrete latent space using vector quantisation followed by additionally enforcing a texture variant discrete latent space which creates a finite shape dictionary. Next, we additionally enforce the shape dictionary to be equivariant to the D16 group via our contrastive method or by using D16 equivariant convolutional kernels shown in the last two columns of table 7.10 respectively. Here, we demonstrate the improved segmentation performance by using both methods to force equivariance of our shape dictionary.

7.6 Conclusion

In this work, we propose to constrain the latent space for any segmentation model to a finite number of shape equivariant components to improve robustness. We impose these constraints either via a constrastive approach where equivariance is learnt or by imposing fixed constraints on the convolutional kernels themselves. We theorise and prove that the denser the equivariant constraints imposed either in the latent space or the kernels, the more robust the segmentation model. We demonstrate this to be true

by improving the domain generalisability of a segmentation model trained on either the cardiac or prostate dataset. We outperform the baseline models by imposing shape equivariance in the model and perform competitively with other SDG methods in the literature.

We anticipate that, over time, equivariant CNNs will be the go-to option for endeavours such as biomedical imaging, where natural symmetries exist. There remains a need for future studies to delve deeper into the myriad of design possibilities for steerable CNNs and further improve performance.

A limitation of this study is that components in the dictionary are sampled independently which can lead to anatomically and topologically inaccurate segmentations. In the next chapter, we will therefore explore methods to sample the shape components such that topology is preserved.

Chapter 8

Topology Preserving Compositionality for Robust Medical Segmentation

This chapter is adapted from the following publication.

- **Santhirasekaram, A.**, Winkler, M., Rockall, A. and Glocker, B., 2023. Topology Preserving Compositionality for Robust Medical Image Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 543-552).

8.1 Introduction

In previous chapters (Chapter 6 and 7), we improve the robustness of deep learning models for medical image segmentation using a discrete latent space with curvature and equivariant-based constraints. However, components sampled from the dictionary to construct the segmentation output are sampled independently from each other which can lead to topologically implausible segmentation outputs. Therefore, in this chapter, we will now develop a method to sample the dictionary in latent space such that topology is preserved from the latent space to the output space. Similar to the previous chapter, we impose a discrete latent space using vector quantisation. Specifically, we incorporate a topological prior using persistent homology in the sampling of our

dictionary to ensure topological accuracy after the composition of the components. We further improve robustness by deep topological supervision applied in an hierarchical manner. However, to emphasise the value of preserving topology when sampling the codebook, we omit geometric and curvature-based constraints which are imposed in previous chapters.

We will demonstrate the effectiveness of our method under various perturbations and in two single-domain generalisation tasks.

8.2 Background

8.2.1 Related Work

Persistent homology for deep representation learning is limited. Topological autoencoders [MHRB20] is the first work however to incorporate persistent homology to preserve the topological structure of the data manifold in the latent representation. Attempts have also been made to incorporate topological signatures into deep neural networks [HKNU17]. For example, [HKNU17] takes a topological signature in the form of a persistence diagram as input into a deep neural network using a novel input layer. They showed SOTA results for graph and shape classification. Persistent homology has also been used as a complexity measure to analyse deep neural network architectures [RTB+18].

The application of persistent homology to deep learning-based segmentation is limited to the output space to produce topologically meaningful segmentations [HLSC19] or as a post-processing method [CBO+20]. Cubical persistent homology [WW16, DW18] has recently been used to analyse fMRI data [RYB+20].

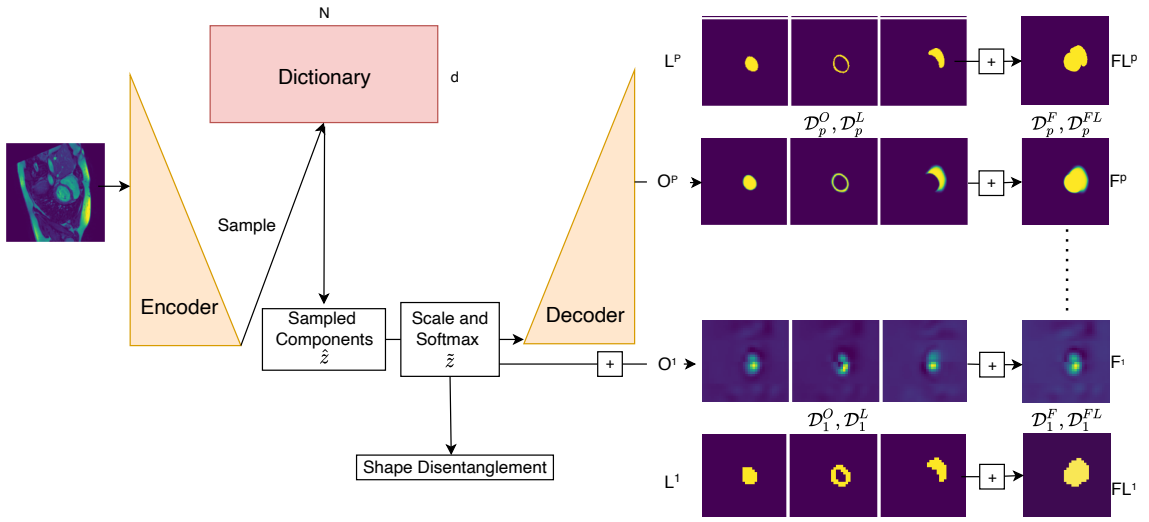


Figure 8.1: Overview of our Topology Preserving Compositionality (TPC) method incorporated in a segmentation model with p levels (scales). $\mathcal{D}$ signifies a persistence diagram derived from a cubical complex C. For example, the persistence diagram for the foreground label segmentation at level p(top right) is expressed as $\mathcal{D}_p^{FL}$ derived from $C(FL^p)$. Background segmentations omitted.

8.3 Methods

8.3.1 Compositionality for Segmentation

Similar to previous chapters, we firstly assume a segmentation network can be decomposed into an encoder Φ_e to map the input space to a lower dimensional embedding space $\Phi_e : \mathcal{X} \rightarrow \mathcal{E}$ and a decoderΦ_d which maps the embedding space to the segmentation output $\Phi_d : \mathcal{E} \rightarrow \mathcal{Y}$.

We continue the claim of previous chapters that the output space of a segmentation model for medical imaging is highly correlated across subjects. Hence, one can hypothesise the low dimensional embedding features to have low variance across the sampled distribution. We therefore again propose to constrain the embedding space to a set of N discrete points required to capture the entire training and test distribution where each discrete point, e_i is a component representing a spatial structure which makes up the set of N components in a dictionary $\mathbb{D}$ shown in figure 8.1.

Similar to chapter 5 and 7, $\mathbb{D}$ is learnt through vector quantisation as proposed by [VDOV+17] to discretise the latent space. We use the same quantisation loss function in equation 7.33 in chapter 7 whereby the codebook is updated in the exact same manner.

Here, we however quantise each feature in the latent space as opposed to quantising spatially.

The sampling of the nearest component in $\mathbb{D}$ leads to a quantised embedding space denoted $\hat{z}$ where $\hat{z} = e_k$. $\hat{z}$ then undergoes scaling before passing through a softmax function to form $\tilde{z}$. There are c compositions corresponding to the number of classes in O^1 as shown in figure 8.1 where c = 3 (background omitted). Therefore, $\tilde{z}$ is subdivided into m/c components which are each summed to form O^1_k (class segmentation outputs from the bottleneck of the model: level 1). A post-quantisation 1×1 convolutional layer is applied to $\tilde{z}$ to increase the number of channels before being passed into the decoder.

8.3.2 Topology Preserving Compositionality

The quantisation process assumes a uniform prior such that components in $\mathbb{D}$ are sampled independently. This was an issue in the previous chapters. Therefore, we propose to incorporate a topological prior into the sampling process such that components are sampled and composed together to form topologically accurate shapes i.e. segmentation maps. In this work, the composition of components is simply the summation over the components. We impose three restrictions to preserve topology. We flatted each feature in $\tilde{z}$ into d dimensional vectors.

Shape Disentanglement:

Firstly, the sampled components should only represent shape features which are distinct from one another so that there is no spatial overlap between the features i.e. disentangled. This led to the following lemma.

Lemma 3. *A shape disentanglement loss term ($\mathcal{L}_{DL}$) forcing the inner product between pairs of d-dimensional quantised latent features passed through the softmax function, $(\tilde{z}_i, \tilde{z}_j)$ to be close to 0 (orthogonal) [SEUD10] as shown in equation 8.1 yields a shape-only latent space.*

$$\mathcal{L}_{DL} = \sum_{i=0}^{i=m-1} \sum_{j=i+1}^{j=m-1} \tilde{z}_i . \tilde{z}_j = 0 \tag{8.1}$$

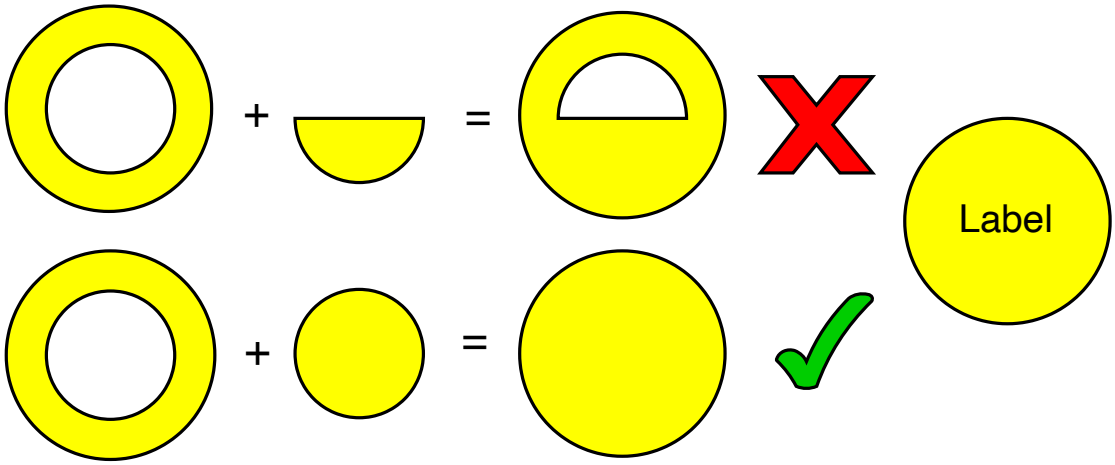


Figure 8.2: Toy example demonstrating the basic principle of TPC.

Proof. Due to the nature of the softmax function, the summation over all m features in $\tilde{z} \in \mathbb{R}^d$ yields a d dimensional vector of 1s. This means if $\mathcal{L}_{DL} = 0$, then this implies that the intersection of the space between $\tilde{z}_i$ and $\tilde{z}_j$ forms an empty set; $\bigcap_{i=0}^{i=m-1} \tilde{z}_i = \emptyset$. We can also infer from this definition that $\tilde{z}_i$ must then only consist of the integers 0 or 1; $\tilde{z}_i \in (\mathbb{Z}/2\mathbb{Z})^d$. For example, if $xy = 0$ and $x + y = 1$ then, $x, y \in \mathbb{Z}/2\mathbb{Z}$. Therefore, by design, we additionally remove textural/style information from the semantics of the latent space, by approximately learning binarised feature maps, $\tilde{z}_i$. This means the k^{th} class segmentation output in O^1 denoted O^1_k where $k \in \mathbb{Z}/3\mathbb{Z}$ must also be binary as demonstrated in equation 8.2.

$$O^1_k = \sum_{i=mk/c}^{i=m(k+1)/c} \tilde{z}_i,\ O^1_k \in (\mathbb{Z}/2\mathbb{Z})^d |\ \tilde{z} \in (\mathbb{Z}/2\mathbb{Z})^d,\ \mathcal{L}_{DL} = 0 \tag{8.2}$$

□

An overlap between features will yield integers greater than 1 under the stated assumptions. For example, in the scaled (0-1) heat maps in figure 8.1 one can visualise brighter regions in O^1 representing overlapping components.

Preserving Topology

Next, we must sample components such that the topology of the composed output O^1_k matches the down-sampled label map L^1_k (see figure 8.1) i.e. same number of connected

components and holes. In the toy example highlighted in figure 8.2, the label is a circle and in the top row, we show an example where the two sampled components; a ring and a semi-circle are composed together to form a shape with a semi-circle shaped hole. The Betti numbers of this shape would be $(1, 1)$; 1 connected component and 1 tunnel while the label has Betti numbers $(1, 0)$ and therefore the homology groups do not match. However, in the bottom row, a ring and a smaller circle are composed to form a shape with Betti numbers $(1, 0)$ which matches the label map. Given, we sum over a non-overlapping set of sampled components when $DL = 0$ to form O^1, then we must enforce that the union over $\tilde{z}$ should form a single connected component like in figure 8.2 with our topological loss.

We next calculate the persistent homology of the cubical complex of each composition in O^1 and L^1 denoted, $PH(C(O^1_k))$ and $PH(C(L^1_k))$ respectively. We can then create persistence diagrams for each k^{th} class segmentation in O^1, denoted $\mathcal{D}^O_{1,k}$ which we aim to match with the persistence diagrams of the down-sampled label map L^1 (see figure 8.1). We minimise the p^{th} Wasserstein distance [MHRB20] of two persistence diagrams shown in equation 8.3 where $\eta; \mathcal{D} \rightarrow \mathcal{D}'$ is a bijection between the persistence diagrams and $p = 2$. This loss function is proven to be stable to noise [ST20] and differentiable [CCG+21].

$$d_w(\mathcal{D}, \mathcal{D}') = \left(inf_{\eta; \mathcal{D} \rightarrow \mathcal{D}'} \sum_{x \in \mathcal{D}} \|x - \eta(x)\|^p_\infty \right)^{\frac{1}{p}} \tag{8.3}$$

Therefore, by minimising the distance $d_w(\mathcal{D}^O_{1,k}, \mathcal{D}^L_{1,k})$, we ensure the class segmentation outputs at level 1 of the decoder $(O^1_{0:2})$ all have one single connected component and only O^1_1 has one tunnel.

Additionally, since the label maps are binary, all topological features are born at 1 and die at 0. Therefore, by minimising the Wasserstein distance, one is also forcing O^1_k to be binary and if $DL = 0$, then $\tilde{z}_i$ must also be 0 or 1 by deduction. This topological loss is hence an additional method to remove the textural information in $\tilde{z}$ which in turn also removes textural information from the dictionary. One can also apply the topological loss to $\tilde{z}$ directly with the assumption that the topological label for $\tilde{z}_i$ is a single connected component which is born at 1 and dies at 0. We can then minimise the Wasserstein distance loss for the 0^{th} dimensional homology with this fixed topological label to

make sure $\tilde{z}$ are binary components. We found this beneficial but too computationally expensive. We also didn't binarise the embedding space via a threshold method such as a steep sigmoid function. This is because of the gradient vanishing either side of the threshold value which are the constant portions in the steep sigmoid function where most of the values will lie. A Dice score loss is added in order to increase the overlap between L^1 and O^1 as the topological loss is position and size invariant.

To preserve the topological constraints imposed in the quantised embedding space through the decoder, we apply deep topological and Dice supervision at each level of the decoder. Therefore, as shown in figure 8.1, we sum the feature maps outputted at each level of the decoder to produce c class segmentation outputs before applying our deeply supervised class loss shown in equation 8.4.

Hierarchical Topology: We apply an additional topological and Dice loss for the foreground segmentation output denoted, F^i formed from the addition of the foreground class segmentation maps in O^i. This hierarchical loss expressed in equation 8.5 is further enforcing accurate class segmentation maps under the assumption they should be composed to form a topologically meaningful foreground segmentation. For example, in figure 8.1 the foreground segmentation is the whole heart which is forced to be a binary single connected component with no tunnels when the segmented parts of the heart (class segmentations) are combined via summation.

$$\mathcal{L}_{Class\_top} = \sum_{i=1}^{i=p} \sum_{k=0}^{k=c-1} d_w(\mathcal{D}^O_{i,k}, \mathcal{D}^L_{i,k}) + Dice(O^i_k, L^i_k) \tag{8.4}$$

$$\mathcal{L}_{Hier\_top} = \sum_{i=1}^{i=p} d_w(\mathcal{D}^F_i, \mathcal{D}^{FL}_i) + Dice(F^i, FL^i) \tag{8.5}$$

The total loss for training the topological preserving compositionality (TPC) framework incorporated into a segmentation model is formally defined as:

$\mathcal{L}_{total} = \mathcal{L}_{Qaunt} + \mathcal{L}_{Class\_top} + \mathcal{L}_{Hier\_top} + \mathcal{L}_{DL}$.

8.4 Experiments

8.4.1 Datasets and Training

We use 3 datasets in our experiments.

Abdomen: This dataset is the BTCV dataset [LXI+15]. All images are normalised between 0 and 1 and randomly cropped $96 \times 96 \times 96$ patches are used for training. During testing, the volume is divided into $96 \times 96 \times 96$ patches and we evaluate each patch in the test image.

Cardiac: This dataset is the M&Ms cardiac imaging 3-class segmentation dataset [CGI+21] which is divided into 4 domains determined by the MRI scanner vendor.

Prostate: We use the NCI-ISBI13 Challenge [BMH+15b] dataset divided into 2 domains with different MRI scanner types and acquisition protocols. We normalised images between 0 and 1 and centre cropped to 256×256.

Training and Evaluation: All models are trained with Adam optimisation [KB14] with a base learning rate of 0.0001 and weight decay of 0.05 for a maximum of 500 epochs on three NVIDIA RTX 2080 GPUs. In order to prevent over-fitting, we apply a simple augmentation scheme for training our method consisting of random rotation and flipping (horizontal and vertical). We evaluate model performance with the Dice score and Betti error [HLSC19]. In the perturbation experiments, We evaluate the relative corruption error in terms of the Dice score after TPC is included in each model. We additionally want to evaluate the topological corruption after various input perturbations, so we also evaluate the relative corruption error in terms of the Betti error.

8.4.2 Perturbation Experiments

In the first set of experiments we analyse how incorporating TPC into three popular segmentation 3D architectures (UNet [RFB15], nn-UNet[IJK+21] and Swin-UNetr[HNT+22]) improves performance under various types of perturbations in the input space for the abdomen dataset. We randomly split the dataset into 18 and 12 for our train and

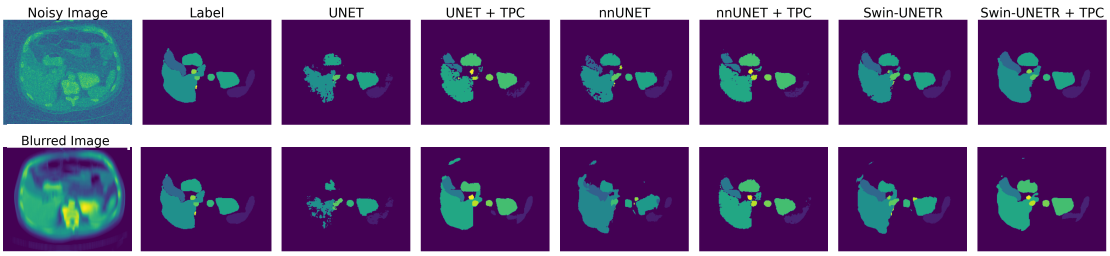


Figure 8.3: In this figure, we show the segmentations of an abdominal CT slice with 20% Gaussian noise addition (top row) and Gaussian blur (bottom row) by the UNet, nn-UNet and Swin-UNetr and after including TPC in each model.

test sets respectively. We use the same hyper-parameters for each perturbation as the previous perturbation experiments in section 6.4.1 of chapter 6.

	Baseline	Gauss	Poisson	S&P	Blur	Motion	Contrast	Intensity
					Dice			
UNet	0.77±0.08	0.71±0.10	0.72±0.09	0.69±0.06	0.72±0.11	0.63±0.13	0.62±0.14	0.68±0.09
+ TPC	**0.79±0.04**	**0.75±0.08**	**0.77±0.07**	**0.74±0.08**	**0.78±0.05**	**0.67±0.10**	**0.65±0.08**	**0.73±0.11**
nnUNet	0.86±0.05	0.80±0.08	0.81±0.03	0.79±0.07	0.80±0.08	0.73±0.12	0.74±0.11	0.75±0.10
+ TPC	**0.87±0.04**	**0.83±0.10**	**0.84±0.09**	**0.82±0.11**	**0.82±0.14**	**0.75±0.14**	**0.77±0.08**	**0.80±0.08**
Swin-UNetr	**0.88±0.05**	0.83±0.09	0.80±0.08	0.81±0.07	0.85±0.04	0.75±0.16	0.78±0.19	0.77±0.11
+ TPC	0.87±0.07	**0.86±0.06**	**0.84±0.06**	**0.85±0.08**	**0.88±0.08**	**076±0.10**	**0.80±0.06**	**0.79±0.05**
					Betti Error			
UNet	0.78±0.19	2.98±1.05	3.03±1.21	2.85±0.90	3.18±1.31	2.72±1.17	2.89±1.22	3.01±1.43
+ TPC	**0.39±0.11**	**0.91±0.41**	**1.10±0.34**	**0.98±0.29**	**1.26±0.38**	**0.57±0.18**	**0.83±0.48**	**0.90±0.21**
nnUNet	0.51±0.24	2.34±0.93	2.52±0.97	2.22±0.92	2.69±0.88	2.50±1.05	3.02±1.09	2.59±1.15
+TPC	**0.25±0.13**	**0.82±0.28**	**0.61±0.11**	**0.76±0.20**	**1.01±0.16**	**0.43±0.10**	**1.29±0.36**	**0.88±0.30**
Swin-UNetr	0.42±0.13	2.39±1.30	2.53±0.95	2.77±0.91	2.94±0.96	2.94±0.89	2.39±0.90	2.81±0.77
+TPC	**0.21±0.08**	**0.83±0.20**	**0.66±0.34**	**0.73±0.25**	**0.90±0.33**	**0.70±0.28**	**0.97±0.21**	**0.85±0.23**

Table 8.1: The mean Dice score and Betti error ± standard deviation before and after TPC is applied to 3 segmentation models under various perturbations in the input space. The results for the Abdominal dataset are shown. The baselines refer to no perturbations applied.

	UNet	nnUNet	Swin-UNetr
Dice RCE	70.9	76.7	54.4
Betti RCE	25.1	28.3	26.3

Table 8.2: The relative corruption error (RCE) (%) in terms of the Dice score and Betti error after TPC is included in the UNet, nnUNet and Swin-UNetr.

In table 8.1, we note significant improvement in both evaluation metrics after incorporating TPC in all three models under various perturbations. Our method has a better effect with a convolutional backbone as opposed to the vision transformer (Swin-UNetr) for textural perturbations. This is likely because the coarse attention mechanism is more robust to noise where it was shown that the vision transformer is acting like a low-pass filter to remove noise whereas convolutions are behaving like high-pass filters [PK21]. In table 8.2, we note the Dice relative corruption error of the UNet (70.9%), nnUNet (76.7%) and Swin-UNetr (54.4%) being significantly less than 100%

after TPC is incorporated into each model. Furthermore, the Betti relative corruption error ranged between 20 to 30 % after TPC is included for all three models highlighting the significantly reduced topological corruption of the segmentation maps under various perturbations.

We demonstrate the value of TPC with a visual example in figure 8.3 where we show the more visually accurate segmentation masks produced by incorporating TPC in the 3 models. We as expected note a greater value of TPC in the CNN models when either 20% Gaussian noise or blur is applied to the image. TPC significantly reduced the number of unconnected components and holes produced by the baseline segmentation models. Additionally, the baseline UNet and nnUNet appear to miss some of the smaller segmentation structures because of noise which reappears by incorporating TPC. This is because by design our topological loss strongly imposes the number of connected components in the label and segmentation should be equal as it is also acting as a component count loss function. Overall, despite the spatial or textural perturbations, our method has learnt a strong anatomical/structural shape prior governed by a dictionary of components which is sampled correctly to ignore the perturbation effect.

8.4.3 Single Domain Generalisation

In the single domain generalisation (SDG) study we evaluate segmentation performance when testing a trained segmentation model on an unseen target domain for the prostate and cardiac datasets. We adopt a leave-rest-out procedure by training a method on a single source domain and holding out the other domains for testing (3 for cardiac and 1 for prostate). We then average the scores across the hold-out experiments (4 for cardiac and 2 for prostate).

In these experiments, we use an adapted Residual-UNet architecture with five levels as our baseline. In the encoder, we double the number of feature channels from 32 at the first level to 512 at the bottleneck. Each scale of the encoder and decoder consists of a single pre-activation residual block [HZRS16b], with group normalisation [WH18] and ReLU activation. We compare our TPC approach with the following single domain generalisation methods: CutOut [DT17], AdvBias [CQQ+20], RandConv [XLY+20],

BigAug [ZWY+20], Jigen [CDB+19] and the baseline model.

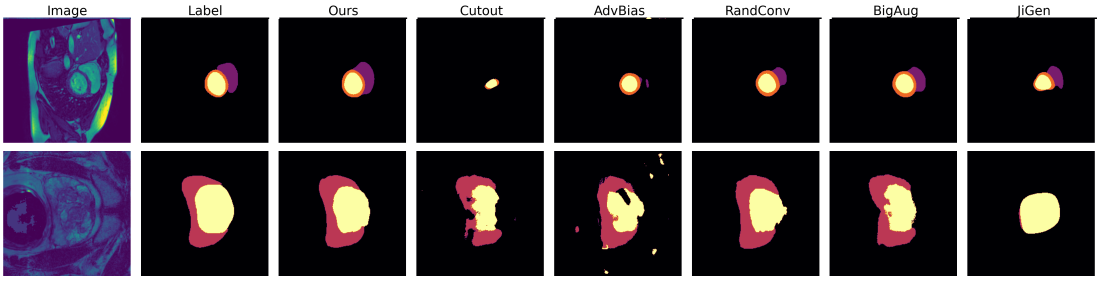


Figure 8.4: Segmentation of a cardiac slice in the top row and prostate slice in the bottom row when testing on a target domain with different SDG methods and our method (TPC).

	Baseline	CutOut	AdvBias	RandConv	BigAug	Jigen	TPC(Ours)
	Transitional Zone						
Dice	0.64±0.14	0.64±0.13	0.65±0.17	0.68±0.14	**0.73± 0.18**	0.65±0.17	0.70±0.14
Betti Error	2.09±1.02	1.90± 0.98	1.53±0.84	1.72±0.51	1.04±0.39	1.62±0.57	**0.80±0.31**
	Peripheral Zone						
Dice	0.38±0.18	0.40±0.18	0.43± 0.16	0.45±0.22	0.55±0.16	0.47±0.14	**0.56± 0.16**
Betti Error	3.85±1.28	3.24±1.08	3.03±0.92	2.08±0.75	1.74±0.43	2.69±1.05	**1.24±0.55**

Table 8.3: Dice score and Betti error ± standard deviation comparing TPC to various other SDG methods. This table shows the results for the prostate dataset.

	Baseline	CutOut	AdvBias	RandConv	BigAug	Jigen	TPC(Ours)
	Left Ventricle						
Dice	0.72±0.11	0.75±0.14	0.73±0.11	0.77±0.10	0.78±0.09	0.71±0.17	**0.78±0.08**
Betti Error	1.24±0.55	1.04±0.50	1.10±0.49	0.74±0.29	0.56±0.21	1.12±0.45	**0.32±0.14**
	Right Ventricle						
Dice	0.74±0.18	0.76±0.15	0.76±0.18	0.73±0.19	0.76±0.11	0.72±0.10	**0.77±0.10**
Betti Error	1.40±0.63	1.33±0.61	1.30±0.58	0.89±0.41	0.60±0.29	1.28±0.97	**0.50±0.16**
	Myocardium						
Dice	0.55±0.15	0.56±0.14	0.61±0.19	0.63±0.14	0.65±0.15	0.64±0.23	**0.67±0.12**
Betti Error	3.45±1.43	2.97±0.96	1.50±0.62	1.37±0.74	1.15±0.55	2.34±4.58	**0.80±0.33**

Table 8.4: Dice score and Betti error ± standard deviation comparing TPC to various other SDG methods. This table shows the results for the cardiac dataset.

Table 8.3 and 8.4 demonstrate that our method achieves the best Betti error scores among the SDG methods for both the prostate and cardiac datasets. The significantly greater Betti error scores by our approach are indicative of our method's superior capability of producing more topologically meaningful segmentations. However, we outperform all methods in the Dice metric except for BigAug where we achieve similar scores. The comparable performance of TPC to BigAug shows how a thorough augmentation strategy can tackle SDG. However, we demonstrate without any aggressive augmentation strategies or adversarial training, simply imposing topological constraints into a segmentation model can still achieve similar or better SDG performance. We visually highlight the effectiveness of our method in figure 8.4 where TPC appears to better

match the topology of the label for both the cardiac and prostate images. For example, in methods such as CutOut and AdvBias applied for prostate segmentation, there are unconnected components or tunnels (holes) and the segmentation maps appear less smooth.

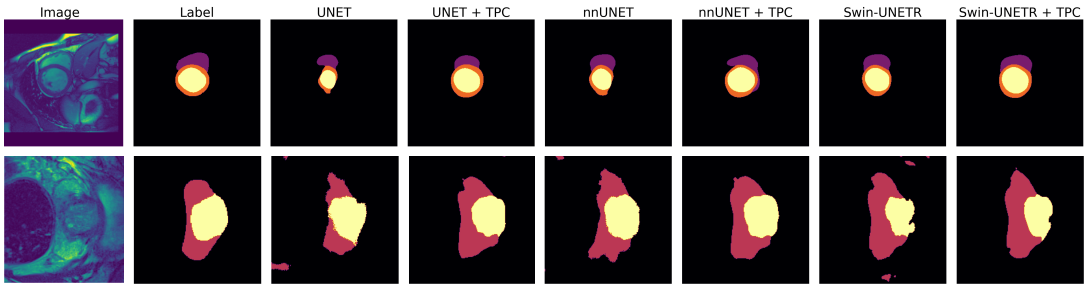


Figure 8.5: Segmentation of a cardiac slice in the top row and prostate slice in the bottom row when testing on a target domain before and after incorporating TPC into 3 segmentation architectures.

8.4.4 Model comparison

	UNet	UNet + TPC	nnUNet	nnUNet + TPC	Swin-UNetr	Swin-UNetr + TPC
	Transitional Zone					
Dice	0.58±0.10	**0.62±0.10**	0.68±0.08	**0.71±0.07**	0.69±0.03	**0.71±0.11**
Betti Error	2.59±0.98	**0.82±0.29**	1.29±0.70	**0.66±0.21**	1.57±0.51	**0.61±0.22**
	Peripheral Zone					
Dice	0.34±0.12	**0.48±0.08**	0.48±0.10	**0.55±0.07**	0.51±0.05	**0.55±0.15**
Betti Error	3.59±1.48	**1.10±0.47**	2.45±0.82	**0.84±0.35**	2.21±0.77	**0.95±0.44**

Table 8.5: The average Dice score and Betti error before and after TPC is applied to 3 segmentation models after testing on the target domains. The results for the prostate dataset are shown.

	UNet	UNet + TPC	nnUNet	nnUNet + TPC	Swin-UNetr	Swin-UNetr + TPC
	Left Ventricle					
Dice	0.70±0.10	**0.74±0.13**	0.80±0.10	**0.83±0.06**	0.80±0.05	**0.82±0.08**
Betti Error	1.79±0.90	**0.53±0.25**	1.01±0.34	**0.45±0.15**	0.97±0.41	**0.40±0.27**
	Right Ventricle					
Dice	0.68±0.07	**0.74±0.11**	0.77±0.12	**0.83±0.08**	0.81±0.06	**0.83±0.09**
Betti Error	1.86±0.84	**0.57±0.28**	0.97±0.33	**0.47±0.18**	0.92±0.44	**0.44±0.20**
	Myocardium					
Dice	0.48±0.22	**0.62±0.15**	0.65±0.11	**0.71±0.10**	0.58±0.09	**0.66±0.16**
Betti Error	2.77±1.59	**1.03±0.46**	1.74±0.62	**0.85±0.36**	1.62±0.89	**1.02±0.46**

Table 8.6: The average Dice score and Betti error before and after TPC is applied to 3 segmentation models after testing on the target domains. The results for the prostate dataset are shown.

In these experiments, we evaluate the improvement in the segmentation performance in SDG by incorporating TPC into the 2D UNet, nn-UNet and Swin-UNetr. In table 8.5 and 8.6, we note significantly improved domain generalisability of all three

models when incorporating TPC by virtue of improved disc scores and Betti errors. We can demonstrate this visually in figure 8.5 where similar to previous experiments, TPC appears to produce smoother segmentations which are more topologically correct compared to the baseline models. This is especially true for prostate segmentation (bottom row) where there are fewer training examples and a larger domain shift. We further note the incomplete rings formed by the baseline UNet and nnUNet for the myocardium segmentations (top row) which is not the case after applying TPC. This experiment once again highlights the benefit of incorporating topological shape priors into structured segmentation tasks.

8.4.5 Ablation Studies

	Cardiac	**Prostate**	**Abdomen**
Size	128	64	512
Dimensionality	324	256	1728
Channels	256	192	112

Table 8.7: Hyper-parameters used in our experiments. The 'size' is the number of components in the dictionary. The 'dimensionality' is the dimension of the codebook components. 'Channels' is the number of embedding features in the model bottleneck.

Dictionary Experiments

We want to learn a dictionary which is as sparse as possible without affecting segmentation performance. Therefore, for all three datasets, we carry out ablation studies to find the minimal required number of components in the dictionary which is incorporated into an adapted Residual-UNet [ÇAL+16, RFB15]. We show our results in table 8.7 below and use these hyper-parameters for our experiments.

Model Experiments

We carry out ablation studies to validate each component of TPC in improving the robustness of our baseline model (adapted Residual-UNet) in the SDG experiments for the prostate and cardiac datasets. Figure 8.6 shows the Dice scores and Betti errors

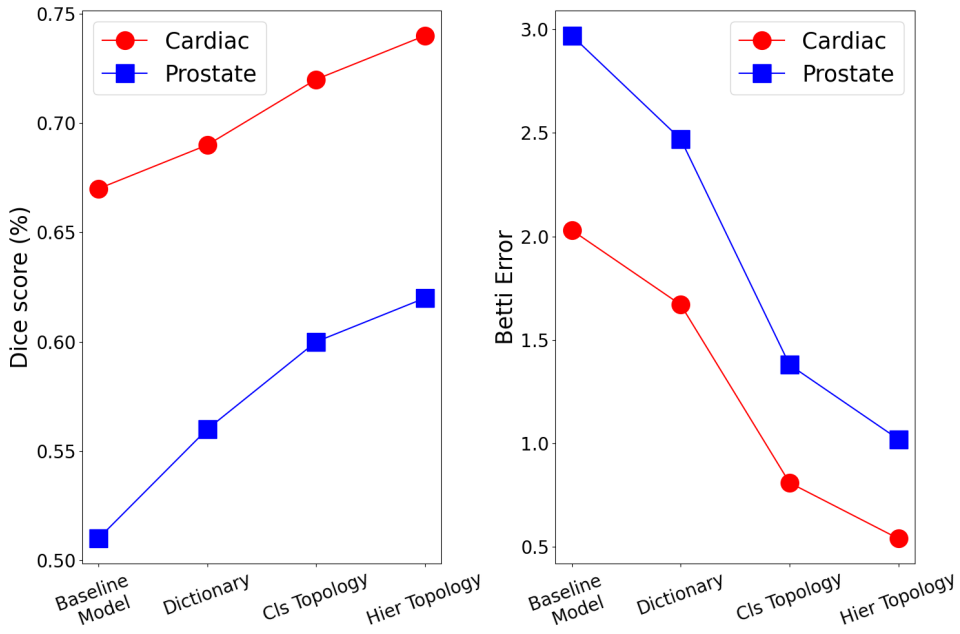


Figure 8.6: The mean Dice score (left) and Betti error (right) after each component of TPC is sequentially added (left to right) into the Baseline Model to assess SDG segmentation for the cardiac and prostate datasets. 'Cls Topology' refers to incorporating the topological loss over each class segmentation. 'Hier Topology' is adding the second topological loss for the foreground segmentation.

steadily improving over the baseline model, firstly with the incorporation of the dictionary followed by adding the class topological and then the hierarchical topological loss terms for both the cardiac and prostate datasets. Incorporating the class topological loss term is likely as hypothesised aiding with sampling the correct dictionary components to compose class segmentation maps. This is further enforced by including the hierarchical topological loss function.

8.5 Discussion

TPC can also be used as a semi-supervised method when there is unlabelled data given the homology of the class segmentations outputs do not vary between subjects i.e. the two class segmentations of the prostate are always two single connected components with no tunnels. This will be explored in future work. We also aim to test our method in more challenging tasks such as instance segmentation where we believe it will be beneficial due to the ability of TPC to match the number of connected components. We noted this property to be useful in the abdominal segmentation tasks where there were

multiple labels.

In conclusion, we propose to improve the robustness of medical imaging segmentation models by constraining the embedding space to a dictionary of disentangled shape components. We use persistent homology to incorporate a hierarchical topological prior in sampling the dictionary to produce topologically accurate segmentations at multiple scales using deep supervision. We demonstrate the effectiveness of our method by incorporating TPC into 3 common segmentation architectures to improve performance under various perturbations and domain shifts. We finally show that our approach beats various SOTA methods in single domain generalisability.

In the next chapter, we realise persistent homology is a very coarse descriptor of a segmentation map and develop a more fine-grained descriptor of the structure of a segmentation map using cellular sheaf theory.

Chapter 9

Sheaf Theoretic Perspective for Robust Segmentation

This chapter is adapted from the following publication.

- **Santhirasekaram, A.**, Pinto, K., Winkler, M., Rockall, A. and Glocker, B., 2023, October. A Sheaf Theoretic Perspective for Robust Prostate Segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 249-259). Cham: Springer Nature Switzerland.

9.1 Introduction

In the previous chapter, we used persistent homology as a tool to preserve gross topological information from the latent space to the output space in a deep learning model for segmentation. Persistent homology however only measures topological properties without measures for more descriptive geometric properties such as size and curvature. The topological properties are also location invariant. In the figure below we show 3 examples with one connected component (Betti number = $(1, 0)$). We can construct identical persistent diagrams for each image in figure 9.1 with a single coordinate, $(1, 0)$ that signifies the birth of a single connected component at $\epsilon = 1$ which dies at $\epsilon = 0$.

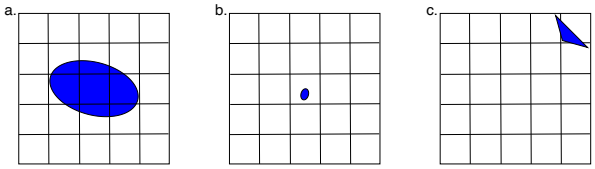


Figure 9.1: Example of 3 images with the same persistence diagrams and Betti numbers.

In figure 9.1, note the significant variation in the geometric properties of all three images which have different areas, perimeters and curvature. The location of each connected component is also different for each image with figure 9.1c showing a connected component in the top right corner. In this work, we propose to develop a more descriptive measure of the shape in terms of its connectivity. Specifically, we want a way of dividing a shape into components and describing how each component is connected to form this shape. This is similar to constructing a mesh over a shape. Other similar constructs include describing the connectivity of a shape with a graph such as a Reeb graph which is useful in particular for tubular structures [BGSF08]. We propose to correctly compose shape components with local and global topological constraints by tracking topological features as we compose the components in an ordered manner. This also allows us to describe the connectivity of the components to form the global shape. We draw on the language of cellular sheaf theory to model compositionality driven by local and global topological correctness [Bre12, Hu20].

9.2 Related work

The use of sheaves is limited in topological data analysis and has only really been used to study algebraic geometry [Har13]. Cellular sheaf theory was used to highlight the geometry of the graph structure was instrumental in determining the poor performance of graph neural networks in the heterophilic setting and their over-smoothing behaviour [BDGC+22]. [BDGC+22] introduces the sheaf Laplacian to provide a framework to define the diffusion process over a graph which aids in node classification in the heterophilic graph setting. The sheaf Laplacian was then used to equip the graph with the right sheaf structure to avoid over-smoothing and problems due to heterophily which led to the sheaf convolutional network. This work was extended by [BBdOBL22]

to develop sheaf attention networks by equipping the attention mechanisms with geometric inductive biases imposed by cellular sheaves which outperformed the graph attention network on synthetic and real-world benchmarks. Sheaves were also recently used to provide a framework for assessing how well a statistical model fits locally on subsets of the data as well as on the entire dataset, which gave valuable insight into understanding the feature space of deep learning models [KJJP21].

Recently, cellular sheaves were used as a method of detecting patch-based merging relations in binary images by uncovering local information from the persistent homology of such images [HC21]. Finally, [ACM22] recently proposed using sheaf theory to construct a shape space which allows one to precisely define how to glue shapes together in this shape space.

9.3 Methods

9.3.1 Cellular Sheaf Over a Poset

The homology of segmentation maps provides a useful tool for analysing global topology but does not describe how local topology is related to constructing global topological features. Sheaf theory provides a way of composing or 'glueing' local data together to build a global object (new data) that is consistent with the local information [Bre12]. This lends well to modelling compositionality. In order to develop a formal language to model the compositionality of a shape via the notion of glueing a set of finite shape components, we must define a cellular sheaf over a poset.

Definition 40. *A* cellular sheaf $\mathcal{F}$ *over a poset* P *is constructed by mapping each element* $p \in P$ *to a vector space* $\mathcal{F}(p)$ *over a fixed field which preserves the ordering in* P *by linear transformations,* $\rho_{\cdot,\cdot}$ *which are inclusion maps in this case [Hu20]. In the language of category theory, a cellular sheaf of vector spaces over a partially ordered set* $(P, \leq)$ *can be described as a functor mapping the elements and partial relations in* P *to the category of vector spaces over a specific field [Cur14, Cur15].*

We next define the important notion of a local section in a cellular sheaf.

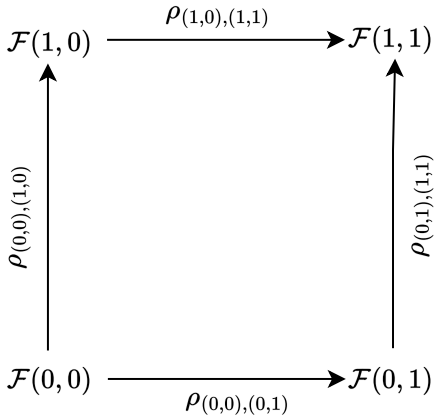


Figure 9.2: Cellular sheaf constructed from the poset P visualised through a commutative diagram.

Definition 41. *Given $\rho_{\cdot,\cdot}$ are inclusion maps and $a \leq b \leq c$ in P, we identify the* local sections *as the matching elements s and t in $\mathcal{F}(a)$ and $\mathcal{F}(b)$ respectively which are also in $\mathcal{F}(c)$*

.

We provide an example to highlight a cellular sheaf over a poset. We consider the poset, $P = \{(0,0),(0,1),(1,0),(1,1)\}$ which are a set of tuples in $\mathbb{Z}/2\mathbb{Z}^2$ with binary relation $\leq$.

In figure 9.2, local sections are formally defined as elements in the form of a tuple (s,t) in the Cartesian product $\mathcal{F}(0,1)\times\mathcal{F}(1,0)$ which satisfy the equation; $\rho_{(0,1)(1,1)}(s) = \rho_{(1,0)(1,1)}(t)$ in $\mathcal{F}(1,1)$. In other words, we describe the relationship between vectors spaces $\mathcal{F}(0,1)$ and $\mathcal{F}(1,0)$ by discovering vectors s and t in $\mathcal{F}(0,1)$ and $\mathcal{F}(1,0)$ respectively which are also in $\mathcal{F}(1,1)$.

9.3.2 Cellular Sheaves for Composing Shapes

We first consider a topological space, Y such as a segmentation output divided into a finite number of subspaces, $\{\emptyset, Y_1, Y_2, \ldots, Y_n\}$ which are the base spaces for Y or equivalently the patches in a segmentation map. If we sequentially glue base spaces together in a certain order to form increasingly larger subspaces of Y starting with $\emptyset$, one can construct a filtration of Y such that; $\emptyset \subseteq Y_1 \subseteq Y_1 \cup Y_2, \ldots, \subseteq Y_1 \cup Y_2, \ldots, \cup Y_n \subseteq Y$. We neatly formalise the subspaces and how subspaces are glued together with a poset.

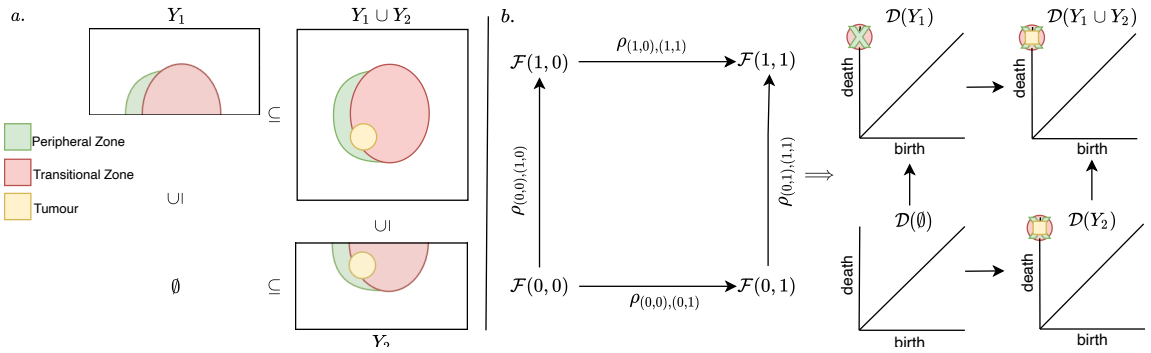


Figure 9.3: a. 2 patches (Y_1, Y_2) glued together in an ordered manner defined by the poset, P, via inclusions maps to form the prostate segmentation output $Y = Y_1 \cup Y_2$. b. cellular sheaf $\mathcal{F}$ over the poset P.

Hence, we can map each element in P with a subspace in Y and binary relations in P are mapped to inclusion maps in Y through the functor $\mathcal{F}$.

In chapter 4, shapes in $\mathbb{D}$ sampled with a uniform prior can lead to anatomically implausible segmentations after composition. In chapter 8 we impose a topology preserving loss to tackle this problem. However, the loss is size and location invariant as demonstrated in figure 9.1. Here, we tackle this problem through the language of cellular sheaves to model the connectivity of patches in an image which provides a connectivity-based loss function. We additionally show how our loss function will improve the localisation of objects in an image. This is particularly important in medical imaging for the detection of tumours.

The quantised embedding space, $\hat{z}$, is split into c groups. The composition of each group in $\hat{z}$ to form each class segmentation, Y_c in the output, Y involves two steps. Initially, a decoder with grouped convolutions equal to the number of classes followed by the softmax function maps $\hat{z} \in \mathbb{R}^{128\times16\times16\times12}$ to $C \in \mathbb{R}^{p\times c\times256\times256\times24}$ where p is the number of patches for each class c. The second step of the composition uses a cellular sheaf to model the composition of Y_c by glueing the patches together in an ordered manner defined by a poset while tracking its topology using persistent homology. This in turn enforces $\mathbb{D}$ to be sampled in a topological preserving manner as input into the decoder/composer to improve both the local and global topological correctness of each class segmentation output, Y_c after composition.

Illustration: We illustrate our methodology of using cellular sheaves with a simple

example in figure 9.3. Here, we show Y as perfectly matching the ground truth label (not one-hot encoded) divided into 2 patches. Y is a topological space with the subspaces, $\mathcal{V} = \{\emptyset, Y_1, Y_2, Y_1 \cup Y_2\}$. A 4-element poset, $P = \{(0,0), (1,0), (0,1), (1,1)\}$ is constructed where given $(x_1, x_2), (y_1, y_2) \in P$ then $(x_1, x_2) \subseteq (y_1, y_2)$ only if $x_1 \leq y_1 \wedge x_2 \leq y_2$. Each element in P is associated with a subspace in $\mathcal{V}$ such that the inclusion relationship is satisfied. Therefore, in figure 9.3a, P defines that Y_1 and Y_2 associated with $(1,0)$ and $(0,1)$ respectively are glued together to form $Y = Y_1 \cup Y_2$ which maps with $(1,1)$. A cellular sheaf $\mathcal{F}$ over P is created by assigning a vector space to $p \in \mathcal{P}$ by deriving a persistence diagram $\mathcal{D}$ for each element in $\mathcal{V}$ associated with p as shown in figure 9.3b. The arrows in figure 9.3b are inclusion maps defined as $\rho_{\cdot,\cdot}$. Persistence diagrams are computed from the sequence of nested cubical complexes of each subspace in $\mathcal{V}$. We do not include $\emptyset$ as a section as we know its persistence homology is 0. The persistence diagrams in figure 9.3b are formed by overlapping the persistence diagrams for each class segmentation. Note that persistence diagrams contain infinite points in the form (τ, τ) (diagonal line in persistence diagrams) which always allows a bijection between two persistence diagrams. The main advantage of our approach is that in addition to ensuring correct local topology (patch level) and global topology (image level), we also force our network to produce topologically accurate patches correctly merged in a topology-preserving manner which matches the ground truth. For example in figure 9.3b, Y_2 contains 3 connected components glued onto Y_1 containing 2 connected components to form Y, which also has 3 connected components. This means an extra connected component is added by Y_2 due to the tumour. This improves tumour localisation as we now know which patch the tumour is in. It also indicates the other 2 connected components in Y_2 are merged into the 2 connected components in Y_1 to form 2 larger connected components (peripheral and transitional zone) in Y i.e. the number of connected components does not change. Hence, the same 2 vectors present in both $\mathcal{F}(1,0)$ and $\mathcal{F}(0,1)$ representing the peripheral and transitional zone are also in $\mathcal{F}(1,1)$, also known as a local section in $\mathcal{F}$.

Implementation: In practice during training, a cellular sheaf is built for each class in the output Y and label $\hat{Y}$ denoted Y^c and $\hat{Y}^c$ respectively. Y^c and $\hat{Y}^c$ are divided into $2 \times 2 \times 1$ patches. Starting with the top left patch, we sequentially glue on each patch in a zigzag manner until the entire image is formed which is formally defined by posets

P^c for Y^c and $\hat{P}^c$ for $\hat{Y}^c$, each containing 7 elements this time. Each element in the poset is associated with a patch i.e. Y_i^c or a sequence of patches glued together. Hence, P^c is bijective with $\{Y_1^c, Y_2^c, Y_3^c, Y_4^c, Y_1^c \cup Y_2^c, Y_1^c \cup Y_2^c \cup Y_3^c, Y^c\}$. We construct cellular sheaves $\mathcal{F}$ over P^c and $\hat{P}^c$ and minimise the distance between these cellular sheaves.

We firstly plot persistence diagrams $\mathcal{D}$ from the set of vectors (τ^i, τ^j) in $\mathcal{F}(P_i^c)$ and $\mathcal{F}(\hat{P}_i^c)$. Next, we minimise the total p^{th} Wasserstein distance (topological loss) between the persistence diagrams $\mathcal{D}(\mathcal{F}(P_i^c))$ and $\mathcal{D}(\mathcal{F}(\hat{P}_i^c))$ shown in equation 9.1 where $\eta : \mathcal{D}(\mathcal{F}(P_i^c)) \rightarrow \mathcal{D}(\mathcal{F}(\hat{P}_i^c))$ is a bijection between the persistence diagrams [MHRB20] and $p = 2$. This loss function is proven to be stable to noise [ST20] and differentiable [CCG+21].

$$\mathcal{L}_{sheaf} = \sum_{c=1}^{c=4} \sum_{i=1}^{i=7} \left(inf_{\eta: \mathcal{D}(\mathcal{F}(P_i^c)) \rightarrow \mathcal{D}(\mathcal{F}(\hat{P}_i^c))} \sum_{x \in \mathcal{D}(\mathcal{F}(P_i^c))} ||x - \eta(x)||_\infty^p \right)^{\frac{1}{p}} \tag{9.1}$$

9.3.3 A Measure Theoretic Approach for Using Cellular Sheaves for Composing Shapes

Cellular Sheaf for Object Localisation Lemma

An increase in the number of patches or in other words increase in the number of elements in the poset also led to improved local topological correctness and patchy connectivity which leads to better tumour localisation but at the cost of increased computational time. In other words, the upper bound of the location accuracy for any finite object in an image Y which is calculated with a cellular sheaf C over a poset P is determined by the number of elements in the poset where each each element in the poset corresponds to a subset of Y. This leads us to our first lemma below.

Lemma 4. *Consider the finest possible discretisation of image Y to produce a maximum of m elements representing voxels whose disjoint union composes Y. The maximum size of the poset is generated by m elements sequentially glued to form increasingly larger sets Y_i until $Y_m = Y$. We denote the collection of subsets Y_i as $\mathcal{Y}$. The union of the m singletons and $\mathcal{Y}$ form a π system which provides the upper bound of the location information for any finite object x in an image Y and also satisfies our construction of the poset P.*

We next define the π system.

Definition 42. *Given a set Ω, a π system consists of a collection of subsets P in Ω which satisfy the following properties:*

- *P is non-empty.*
- *if $X, Y \in P$ then, $X \cap Y \in P$*

Before we begin our proof, we must define our measurable spaces and functions. We must first construct measurable space from Y by first considering all subsets of Y denoted $\mathbb{P}$ which is also the power set and define a σ-algebra, $\mathcal{M}$ with a uniform probability measure μ_p. We start with all the possible collection of subsets $A_i \in \mathbb{P}$ whose disjoint union forms Y defined below in equation 9.2.

$$\mathbb{S} = \left\{ \bigsqcup_{A_i \in \mathbb{P}} = Y \right\} \tag{9.2}$$

We next construct a set of the smallest σ-algebras, $\mathbb{M}$ from each collection of subsets in $\mathbb{S}$. This is a bijective mapping, $\mathcal{Z} : \mathbb{S} \to \mathbb{M}$ because we know the smallest σ-algebra is unique and exists [Bre12].

We must now create another measurable space to define the vector space representing a persistence diagram. In this proof, we will only work with connected components. This is easily extendable to higher-order homological dimensions. The maximum number of connected components in Y is m because the smallest possible connected component can be measured as one of the m elements. Therefore, we generate a σ-algebra from the m possible connected components where n connected components is represented as the tuple $(1, 0)$ repeated n times where $(1, 0)$ corresponds to (b, d) in the persistence diagram i.e. 2 connected components corresponds to $(1, 0), (1, 0)$. We abuse notation and define n connected components as $(1, 0)^n$. Note, $(1, 0)^1 \in (1, 0)^2 ... \in (1, 0)^m$.

Lemma 5. *The collection of m $(1, 0)^n$ elements is a σ-algebra denoted $\mathcal{N}$ if we include the empty set.*

Proof. Lemma 5 is true because we fulfil the criteria of a σ-algebra shown below.

- $(1,0)^m$ is the entire set, ω and we know this exists by definition.
- Next the complement of any set $(1,0)^n$ is $(1,0)^{m-n}$ for $n \leq m$ which exists in the set.
- The union of any two sets $(1,0)^n$ and $(1,0)^{n+r}$ where $2n + r \leq m$ is $(1,0)^{2n+r}$ is in the set.

□

We now have a measurable space with counting measure μ_c where the set space ω_v is the same as the same smallest σ-algebra $\mathcal{N}$. We now must define a measurable function, $\mathcal{J} : (\mathbb{S}_i, \mathcal{M}_i, \mu_p) \to (\omega_v, \mathcal{N}, \mu_c)$. We define the probability measure on $A_i \in \mathbb{S}_i$ as $1/|A_i|$. In order for $\mathcal{J}$ to be measurable then the pre-image of elements in the $\mathcal{N}$ must exist in $\mathcal{M}$ formally denoted, $\mathcal{J}^{-1}(\mathcal{N}) \in \mathcal{M}$. The smallest σ-algebra containing $\mathcal{J}^{-1}(\mathcal{N})$ and a subset of $\mathbb{S}_i$ is the set $\{\emptyset, \mathbb{S}_i\}$. In addition, in order for $\mathcal{J}$ to be measurable then we must assume at least one connected component exists in Y and it's not therefore empty.

Intuitively, given an object $x \in Y$, the smaller the cardinality of A_i ($|A_i|$) is in the disjoint union subsets of Y, then the more localisation information contained within A_i.

Definition 43. *Given we defined a probability measure, μ_p on A_i, then in order for the entropy to decrease as $\mu_p(A_i)$ increases, we must define the entropy of A_i as shown in the equation below.*

$$H(A_i) = -\frac{1}{\mu_p(A_i)\log(\frac{1}{\mu_p(A_i)})} \tag{9.3}$$

We now prove lemma 4.

Proof. First, to define the localisation entropy of the measurable function, we must map $\mathcal{N}$ to the element $\mathbb{S}_j \in \mathbb{S}$ which constructs the σ-algebra $\mathcal{M}_j$ and define an entropy function over $\mathbb{S}_j$. This is shown below.

$$H(\mathcal{J}) = \sum_{A_i \in \mathbb{S}_j} H(A_i); A_i = \mathcal{Z}^{-1}(\mathcal{J}^{-1}(\mathcal{N})) \tag{9.4}$$

The above equation demonstrates the finer the discretisation of Y or in other words, the larger the cardinality of $\mathbb{S}_j$, then the greater the localisation information in $\mathcal{J}$. Therefore,

the smallest π system containing Z is all the elements in Z with the inclusion of $\emptyset$. This is a π system because it is non-empty and the intersection of any two elements in this system is $\emptyset$. Unfortunately, the $\leq$ relationship does not exist between any two elements of this π. Therefore, in order to satisfy the properties of a poset defined in our method, we sequentially glue each element in Z to form increasingly larger sets Y_i until $Y_m = Y$. The union of the collection of subsets Y_i and Z defines a π system which provides the upper bound for the location information of any object, x in Y and satisfies our construction of the poset, P. This concludes our proof of lemma 4.

□

We can now also make the following corollary given the above proof.

Corollary 1.

$$\mathcal{M}_i \subseteq \mathbb{M}_j \rightarrow ||\mathbb{S}_i|| \leq ||\mathbb{S}_j|| \rightarrow H(\mathcal{J}_i) : (\omega_y \mathcal{M}_i, \mu) \rightarrow (\omega_v, \mathcal{N}, \mu) \leq H(\mathcal{J}_j) : (\omega_y \mathcal{M}_j, \mu) \rightarrow (\omega_v, \mathcal{N}, \mu) \tag{9.5}$$

Cellular Sheaf for Shape Analysis Lemma

We now will use the language of measure theory to prove the advantage of using cellular sheaves for extracting richer geometric information such as size from a shape.

We start with first producing the following lemma.

Lemma 6. *We take all $\mathbb{S}_j \in \mathbb{S}$ such that, $\forall (A_i, A_j) \in \mathbb{S}_j, |A_i| = |A_j|$ to construct a new collection of subsets $\mathbb{T} \in \mathbb{S}$. These are all possible discretisations of Y where A_i is equal in size. The entropy of the smallest σ-algebra generated from $\mathbb{T}_i$ is greater than the smallest σ-algebra generated from $\mathbb{T}_j$ if the number of elements in $\mathbb{T}_i$ is more than the number of elements in $\mathbb{T}_j$.*

To prove lemma 6 we first define the entropy of a σ-algebra.

Definition 44. *The equation for the entropy of a σ algebra $\mathcal{M}_j$ generated from $\mathbb{T}_j$ is shown below. We denote all possible collection of subsets in $\mathcal{M}_j$ whose disjoint union is Y as $\mathbb{K}$*

$$H(\mathcal{M}_j) = \sup_{K_i \in \mathbb{K}} \sum_{A_i \in K_i} \mu_p(A_i) \log(\mu_p(A_i)) \tag{9.6}$$

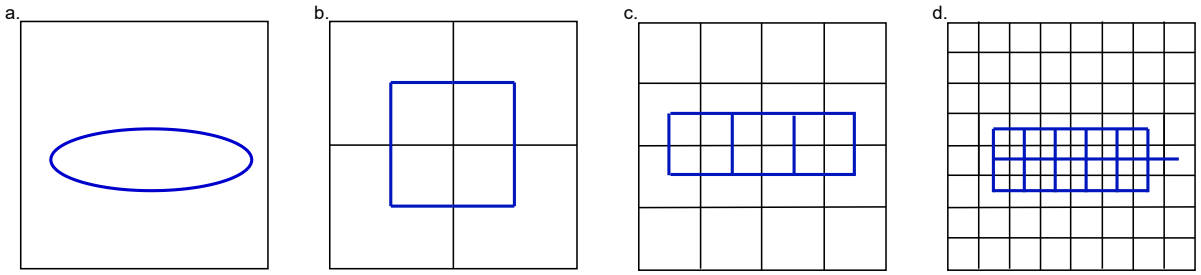


Figure 9.4: a.Original shape b,c,d. Edges between patches defined by the cellular sheaf over a poset defined in our method on a b. 2×2, c. 4×4, d. 8×8 grid.

Proof. Given the definition 44, the collection of subsets $K_i \in \mathbb{K}$ with the highest number of elements has the highest σ-algebra entropy. Therefore, the σ-algebra containing the K_i with the most elements has the highest entropy. This means the finer the discretisation of Y or in other words as the number of elements in $\mathbb{T}_j$ increases, then the entropy of the σ-algebra generated from $\mathbb{T}_j$ increases. □

Intuitively, we can understand the entropy of the σ-algebra to increase as the number of possible collections of subsets in the σ-algebra whose disjoint union is the entire set also increases.

We also note by analysing the connectivity of patches we can extract geometrical shape information. One can extract all pairs of horizontally and vertically connected adjacent patches and connect an edge between pairs of patches which have local sections defined by the cellular sheaf over the poset. Each pair of patches will generate a 4-element poset and a cellular sheaf over the poset as shown in 9.3b. We next can extract the edges between patches which define the boundary of the shape to extract geometric information about the shape. This leads to our next lemma.

Lemma 7. *Given a finite shape x in Y, the finer the discretisations of Y, then the lower the boundary entropy of the shape x or in other words, the better the approximation of the boundary function which defines shape x.*

We intuitively demonstrate why lemma 7 is true with an example above. In figure 9.4 one can clearly see that the shape in figure 9.4a is better approximated as the discretisation of Y becomes finer from a 2×2 grid to a 8×8 grid.

In order to prove lemma 7, we first make the following two assumptions.

Assumption 7. *Given a finite shape x in Y, then this shape has a well-defined boundary function $\mathcal{Z}$ which is continuously differentiable.*

Assumption 8. *We only consider discretisations of Y such that all subsets, Y_i in the disjoint unions of subsets which construct Y are equal cardinality of the order 2^n. This is all possible discretisations of Y where Y is partitioned equally into 2^n elements.*

Before we begin our proof, we must construct a set representing the boundary of shape x. In lemma 6 we constructed a collection $\mathbb{T}$ where each element in the collection represented the disjoint union of subsets to construct Y where the cardinality of each subset is equal. Here, we define the patches in $\mathbb{T}$ as the coordinates $[0, a] \times [0, a] \in \mathbb{R}^2$ of the patch centre and the union of patches as the collection of coordinates of each patches' centre in the union. Next, from each $\mathbb{T}_i \in \mathbb{T}$, we extract all pairs of subsets $\mathbb{W}_i = (A_i, A_j)$ which are adjacent in image Y to construct a new collection $\mathbb{W}_i \in \mathbb{W}$.

We next construct a measurable space over each $\mathbb{W}_i \in \mathbb{W}$ by first constructing its smallest σ-algebra $\mathcal{W}_i$. We now define a function $\mathcal{Z}_i$ mapping $\mathbb{W}_i \in \mathbb{W}$ to binary values $\{0, 1\}$ on the real line, $\mathbb{R}$ with the Borel σ-algebra $\mathcal{B}$ and Borel measure, μ formally defined, $\mathcal{Z}_i : (\Omega, \mathcal{W}_i, \mu) \to (\mathbb{R}, \mathcal{B}, \mu)$. Ω is the entire set in $\mathbb{W}_i$ i.e. the union of elements in $\mathbb{W}_i$ and $\mathcal{Z}_i$ is a binary random variable. Each element $W_i \in \mathcal{W}_i$ is mapped 1 if W_i is a boundary of x and 0 if it does not represent a boundary. The boundary is also a local section in our cellular sheaf over the defined poset. The pre-images of all other $B_i \in \mathcal{B}$ is $\emptyset \in \mathcal{W}_i$.

Lemma 8. *$\mathcal{Z}_i$ is a measurable function.*

We prove lemma 8 as follows.

Proof. $\mathcal{Z}_i$ is measurable because the pre-image of all Borel sets B in $\mathcal{B}$ lies in $\mathcal{W}_i$.

$$\mathcal{Z}_i^{-1}(B) = \begin{cases} W_i & \text{if } 1 \in B, 0 \notin B \\ W_i^c & \text{if } 1 \notin B, 0 \in B \\ \emptyset & \text{if } 1 \notin B, 0 \notin B \\ \Omega & \text{if } 1 \in B, 0 \in B \end{cases} \tag{9.7}$$

□

We now have a set of functions $\mathcal{Z}_i$ mapping each W_i to binary values $\{0, 1\}$ on the real line $\mathbb{R}$ with the Borel σ-algebra.

Lemma 9. *$\mathcal{W}_i$ becomes a Borel σ-algebra $\mathcal{B}$ generated from $[0, a] \times [0, a] \in \mathbb{R}^2$ as $i \to \infty$*

We now prove lemma 9.

Proof. The number of elements in $\mathbb{T}_i$ increases as i approaches ∞ which allows us to make the following implication below.

$$\uparrow |\mathbb{T}_i| \to \uparrow |\mathcal{W}_i| \quad \text{as} \quad i \to \infty \tag{9.8}$$

It is trivial to note also as $i \to \infty$, then Y is discretised into infinite partitions. This leads to the following implication.

$$\forall (A_i, A_j) \in \mathbb{T}_i, A_i - A_j \to 0 \to \sigma(\mathbb{T}_i) = \mathcal{B} \quad \text{as} \quad i \to \infty \tag{9.9}$$

The above implication then means that $\mathcal{W}_i$ becomes the Borel σ-algebra generated from $[0, a] \times [0, a] \in \mathbb{R}^2$ as $i \to \infty$. □

Given, the proof of lemma 8 and lemma 9, we can now prove lemma 7 as follows.

Proof. We now define the pre-image of each function $\mathcal{Z}_i$ as the composition of the true boundary function pre-image $\mathcal{Z}^{-1}$ and another function κ_i shown below.

$$\mathcal{Z}_i^{-1} = \kappa_i \circ \mathcal{Z}^{-1} \tag{9.10}$$

κ_i is a measurable map defined as a mapping from the Borel σ-algebra with finite Borel measure μ generated from the domain $[0, a] \times [0, a] \in \mathbb{R}^2$ which is the pre-image of the function $\mathcal{Z}$ to the pre-image of $\mathcal{Z}_i$ shown below.

$$\kappa = ([0, a] \times [0, a] \in \mathbb{R}^2, \mathcal{B}, \mu) \to (\Omega, \mathcal{W}_i, \mu) \tag{9.11}$$

κ is measurable because κ^{-1} always lies in the Borel σ-algebra generated by $[0,a]\times[0,a] \in \mathbb{R}^2$.

This implies the pre-image of the function $\mathcal{Z}$ becomes $\mathcal{W}_i$ as $i \to \infty$. Therefore, the following deduction can be made.

$$\mathcal{Z}^i \uparrow \mathcal{Z} \quad \text{as} \quad i \to \infty \tag{9.12}$$

The above deduction states as $i \to \infty$, $\mathcal{Z}_i$ better approximates $\mathcal{Z}$ pointwise in a monotone increasing way. This therefore implies the finer we discretise Y, then the better the approximation of the boundary function of $x \in Y$ and hence the geometric properties of x. This concludes our proof of lemma 7.

□

9.4 Experiments

9.4.1 Topology Experiments

In the first set of experiments, we compare our method to using Persistent homology over the entire image as a segmentation loss as implemented by [CBO+20] denoted 'TopoLoss'. This topological loss is applied to each of the segmentation class outputs.

Datasets

In this set of experiments, we apply our method to two datasets.

Cardiac: Firstly, we have a cardiac dataset, which is the M&Ms cardiac imaging 3-class segmentation dataset [CGI+21]. All images were normalised between 0 and 1 and cropped to 288×288. This dataset consists of 4 different source domains as described in chapter 2.

Prostate: We use the NCI-ISBI13 Challenge [BMH+15b] with 2 source domains as described in chapter 2. We normalised images between 0 and 1 and centre cropped to

256×256.

Model and Training

The model we used was a simple 2D Residual UNet with 5 levels. The architecture of this model is identical to the baseline model used in the SDG experiments in chapter 8. All models were trained with Adam optimisation [KB14] with a base learning rate of 0.0001 and weight decay of 0.05 for a maximum of 500 epochs on three NVIDIA RTX 2080 GPUs. To prevent over-fitting, we apply a simple augmentation scheme for training our method consisting of random rotation and flipping (horizontal and vertical). We evaluate model performance with the Dice score, Betti error [HLSC19] and maximum Hausdorff distance (mm). A batch size of 5 is used for both datasets. We apply the loss function in equation 9.13 given a label $\hat{Y}$ and output Y for training the model with our cellular sheaf loss defined in equation 9.1. Our baseline is the Dice loss function used to train our model. Our comparison is 'TopoLoss' plus a Dice loss applied to the baseline model

$$\mathcal{L}_{total} = \mathcal{L}_{Dice}(\hat{Y}, Y) + \mathcal{L}_{sheaf}(\hat{Y}, Y) \tag{9.13}$$

Results

We apply our method in the single-domain generalisation setting. The setup for these experiments is identical to that of the SDG experiments in chapter 8. We again train a method on a single source domain and hold out the other domains for testing.

	Baseline	**TopoLoss**	**SheafLoss** (2×1)	**SheafLoss** (2×2)	**SheafLoss** (4×4)	**SheafLoss** (8×8)
	Transitional Zone					
Dice	0.64±0.14	0.66±0.15	0.67±0.11	0.67±0.10	0.68±0.15	**0.70±0.10**
Betti Error	2.09±1.02	1.01±0.40	0.90±0.31	0.92±0.30	0.85±0.25	**0.80±0.24**
HD	27.45±9.56	25.48±8.38	23.17±9.08	22.15±9.59	19.28±6.49	**17.68±5.98**
	Peripheral Zone					
Dice	0.38±0.18	0.40±0.10	0.42±0.14	0.43±0.17	0.45±0.11	**0.46±0.16**
Betti Error	3.85±1.28	1.58±0.53	1.44±0.41	1.38±0.33	1.31±0.39	**1.26±0.35**
HD	45.21±16.93	40.27±15.31	36.34±14.12	33.16±15.76	32.37±13.18	**30.47±9.56**

Table 9.1: Dice score,Betti error and maximum Hausdorff distance (HD) ± standard deviation comparing 'TopoLoss' to our cellular sheaf-based loss with different discretisations of the segmentation output and label. This table shows the results for the prostate dataset.

	Baseline	TopoLoss	SheafLoss (2×1)	SheafLoss (2×2)	SheafLoss (4×4)	SheafLoss (8×8)
	Left Ventricle					
Dice	0.72±0.11	0.73±0.13	0.73±0.15	0.75±0.11	0.77±0.17	**0.78±0.12**
Betti Error	1.24±0.55	0.67±0.20	0.63±0.24	0.60±0.17	0.59±0.15	**0.55±0.13**
HD	17.18±5.82	15.90±4.60	14.98±3.49	12.86±3.86	10.17±3.72	**9.78±3.57**
	Right Ventricle					
Dice	0.74±0.18	0.75±0.15	0.76±0.11	0.77±0.16	0.77±0.10	**0.79±0.10**
Betti Error	1.40±0.63	0.72±0.22	0.70±0.28	0.69±0.20	0.69±0.18	**0.67±0.16**
HD	20.44±6.05	18.23±5.71	17.68±5.89	15.72±4.91	14.17±4.04	**12.21±3.36**
	Myocardium					
Dice	0.55±0.15	0.56±0.18	0.58±0.12	0.58±0.18	0.59±0.11	**0.60±0.12**
Betti Error	3.45±1.43	1.01±0.48	0.87±0.30	0.85±0.37	0.84±0.31	**0.82±0.28**
HD	28.73±8.17	25.29±6.25	23.45±5.31	21.98±4.19	20.08±3.90	**18.68±3.22**

Table 9.2: Dice score, Betti error and maximum Hausdorff distance (HD) ± standard deviation comparing 'TopoLoss' to our cellular sheaf-based loss with different discretisations of the segmentation output and label. This table shows the results for the cardiac dataset.

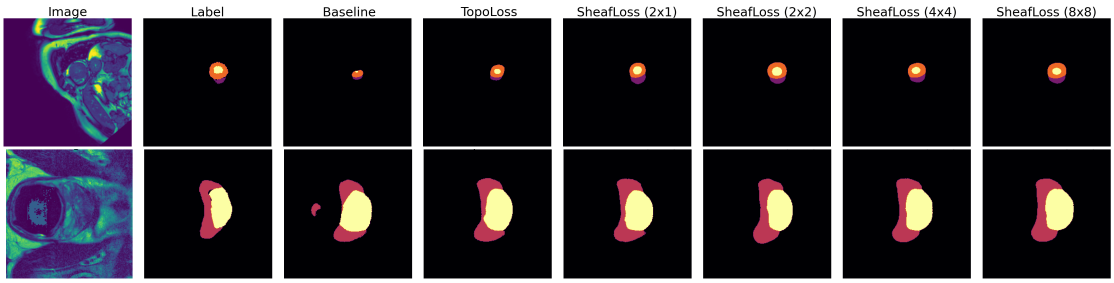


Figure 9.5: In this figure, we show the segmentation outputs from the baseline model trained with the Dice loss (Baseline), Dice loss plus TopoLoss ('TopoLoss') and Dice loss plus our cellular sheaf-based loss (SheafLoss). We apply our sheaf-based loss to $2 \times 1, 2 \times 2, 4 \times 4, 8 \times 8$ discretisations of the segmentation. The segmentation outputs of a cardiac slice are shown in the top row and a prostate slice in the bottom row when testing on an unseen target domain.

In table 9.2 and 9.1, we observe our cellular sheaf-based loss function improves the Dice score, maximum Hausdorff distance and Betti error from the baseline model trained only on the Dice loss when testing on domains which are different from the single source domain which we trained our model on. We also note the cellular sheaf-based loss function achieves better performance in all metric scores compared to 'TopoLoss'. The results in table 9.2 and 9.1 overall show that topological constraints improve the robustness of a segmentation model for both the prostate and cardiac datasets. Additionally, we show we can improve robustness further by increasing the strictness of the topological constraints imposed on our output space in the form of our cellular sheaf-based loss function. Note for the prostate and cardiac dataset in table 9.2 and 9.1 respectively, we demonstrate by imposing a cellular sheaf constrain over increasing discretisations of Y and $\hat{Y}$, we improve segmentation performance further.

'TopoLoss' can be seen trivially as applying our loss function to the entire image (1×1 grid); this is the cellular sheaf over a single element poset. The best segmentation performance is achieved by our method applied on a 8×8 grid.

In figure 9.5, we show visually that our loss functions improve the geometric structure of the segmentation output. Specifically, as the discretisation of our label and image increases, the shape of the segmentation output appears to better match the label. This is highlighted in table 9.2 where the maximum Hausdorff distance which is a good measure of boundary accuracy steadily decreases with incorporating our loss function over finer discretisations of Y. This is observed for both the cardiac and prostate segmentation outputs. Also, note in figure 9.5, the correct number of components in the prostate and cardiac segmentation output by training our baseline model with a topological loss function.

9.4.2 SDG Comparison Experiments

In the next set of experiments, we apply our cellular sheaf-based loss function to the best-performing model in chapter 7 demonstrated in figure 7.1. As mentioned in chapter 7, the best performing model, was our contrastive based method trained with $D8h$ group equivariance. We apply our methods to the same datasets as our topology experiments in section 9.4.1.

Experimental Setup

In these experiments, we use the exact model (hybrid UNet), image pre-processing and training setup used for the $D8h$ group equivariant model which is described in section 7.5.2 of chapter 4. Note that the segmentation output and label is 3D. We apply our loss function to the $D8h$ group equivariant model in order to combat the issue of sampling the shape dictionary independently, which can lead to anatomically implausible segmentations. Due to the increased computational complexity, of applying our loss on a finer discretisation of a 3D segmentation, we use a $2 \times 2 \times 1$ partition of the label and output to apply our loss. Note, a finer discretisation will lead to improved segmentation performance at the cost of increased computational complexity. The datasets

	Baseline	CutOut	BigAug	AdvBias	RandConv	Jigen	vMFNet	D8h	D8h+SheafLoss
	Transitional Zone								
Dice	0.64±0.13	0.65±0.20	0.72±0.15	0.67±0.17	0.69±0.14	0.67±0.18	0.72±0.11	0.73±0.13	**0.75±0.12**
Betti Error	2.10±1.02	1.89±0.86	1.51±0.64	1.70±0.73	1.52±0.60	1.80±0.82	1.39±0.51	1.21±0.46	**0.78±0.12**
HD	24.33±4.12	23.55±4.45	19.81±5.03	21.17±3.98	20.98±4.29	22.98±5.68	19.01±5.38	18.01±4.08	**16.92±4.51**
	Peripheral Zone								
Dice	0.41±0.09	0.41±0.17	0.53±0.16	0.45±0.14	0.49±0.16	0.41±0.12	0.52±0.15	0.53±0.09	**0.56± 0.12**
Betti Error	3.80±1.31	3.09±1.27	2.18±0.94	3.30±1.14	2.19±1.01	3.05±1.16	2.02±0.82	1.89±0.78	**0.97±0.49**
HD	56.88±18.72	51.06±20.17	30.94±14.88	44.72±16.83	38.10±12.88	49.87±17.92	32.19±12.48	28.92±9.92	**24.71±10.29**

Table 9.3: Dice score, Betti error and maximum Hausdorff distance (HD) ± standard deviation comparing the D8h shape equivariant and D8h shape equivariant+SheafLoss segmentation models trained via contrastive learning to various other SDG methods. This table shows the results for the prostate dataset.

	Baseline	CutOut	BigAug	AdvBias	RandConv	Jigen	vMFNet	D8h	D8h+SheafLoss
	Left Ventricle								
Dice	0.74±0.13	0.75±0.13	0.78±0.19	0.76±0.11	0.77±0.18	0.75±0.18	0.78±0.11	0.80±0.17	**0.82±0.13**
Betti Error	1.23±0.52	1.15±0.45	0.81±0.31	0.96±0.28	0.80±0.27	1.03±0.33	0.74±0.23	0.70±0.20	**0.56±0.16**
HD	15.31±3.1	14.81±4.11	11.11±2.99	12.99 ±3.88	12.54±3.48	13.99±4.06	10.08±3.77	9.95±3.08	**8.04±3.01**
	Right Ventricle								
Dice	0.72±0.16	0.72±0.14	0.77±0.13	0.74±0.17	0.75±0.13	0.73±0.18	0.79±0.14	0.79±0.15	**0.82± 0.10**
Betti Error	1.38±0.60	1.23±0.48	0.90±0.36	0.94±0.31	0.89±0.37	0.99±0.40	0.86±0.28	0.85±0.21	**0.66±0.20**
HD	18.76±4.43	17.05±3.88	14.08±3.34	15.82±4.62	14.19±3.29	16.91±4.50	14.04±3.76	13.03±2.87	**10.92± 2.25**
	Myocardium								
Dice	0.57±0.21	0.59±0.14	0.63±0.18	0.61±0.19	0.61±0.10	0.60±0.19	0.63±0.15	0.64±0.18	0.66±0.14
Betti Error	3.41±1.49	2.98±1.18	2.05±1.05	2.28±0.97	1.93±0.71	2.63±1.26	1.81±0.64	1.69±0.53	**0.81±0.30**
HD	26.55±5.21	24.88±5.78	20.08±5.19	22.53±4.58	22.08±4.16	23.76±4.74	20.45±4.43	18.29±4.45	**16.54± 4.08**

Table 9.4: Dice score, Betti error and maximum Hausdorff distance (HD) ± standard deviation comparing the D8h shape equivariant and D8h shape equivariant+SheafLoss segmentation models trained via contrastive learning to various other SDG methods. This table shows the results for the cardiac dataset.

and train/test splits are identical to that of the topology experiments in section 9.4.1. We evaluate performance with the Betti error as well as the Dice score and maximum Hausdorff distance.

We compare our method with several approaches to tackle SDG segmentation namely, RandConv [XLY+20], AdvBias [CQQ+20], CutOut [DT17], Jigen [CDB+19] and BigAug [ZWY+20] applied to the baseline model. We also compare to a compositionality-driven segmentation method called the vMFNet [LTS+22].

Results

We note segmentation performance of the D8h shape equivariant model improves for all three metrics after training with our cellular sheaf-based loss function. We show this to be true for both the prostate and cardiac datasets in table 9.3 and 9.4 respectively. The Betti error especially improved by using our loss function whereby the Betti error

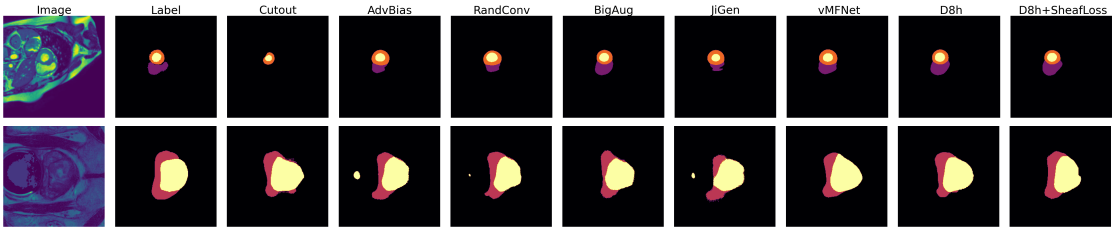


Figure 9.6: In this figure, we show the segmentation outputs comparing the D8h shape equivariant model with the *D8h* shape equivariant model trained with our cellular sheaf loss. We also show the segmentation output of the baseline model trained with various SDG techniques. The segmentation outputs of a cardiac slice are shown in the top row and a prostate slice in the bottom row when testing on an unseen target domain.

was reduced by an average of 0.68 and 0.40 points for the prostate and cardiac datasets respectively (averaged across all classes). The improvement of the Betti error and the maximum Hausdorff distance for all classes is reflective of the geometry and topology of the segmentation output better matching the label. Furthermore, our D8h shape equivariant model with and without our novel loss function demonstrates superior segmentation performance in the task of single domain generalisation compared to the other SDG methods (table 9.3 and 9.4). Specifically, we demonstrate in table 9.3 and 9.4 that our method which constrains the latent space to a finite number of components outperforms a similar method which identifies the finite components in the latent space as Von Mises-Fisher kernels (vMFNet).

In figure 9.6, we visually demonstrate that training our D8h shape equivariant model with our cellular sheaf-based loss function produces a more topologically accurate segmentation output. For example, the peripheral zone in the prostate segmentation produced by the D8h shape equivariant model has an unconnected component in an adjacent patch which is then joined up using the loss function.

9.4.3 SDG Tumour Segmentation Experiments

In the final set of experiments, we evaluate the performance of our D8h shape equivariant model trained with the Dice score plus our cellular sheaf-based loss function to improve the localisation of tumours. This is a 3-class segmentation problem assessing single domain generalisability of prostate tumour and zonal segmentation.

Datasets

We compose an internal dataset for training which contains 205 T2 weighted and ADC images obtained from a single site across 6 years.

The test set has 68 T2 weighted images from the PROSTATEx-2 challenge [LDB+17, LKB+17] acquired from RUNMC on a Siemens 3 Tesla scanner with a surface coil. All experiments were validated on 10 samples.

Experimental Setup

Our baseline model is the same hybrid UNet as in section 9.4.2. Image pre-processing and model training are identical to the experiments in section 9.4.2. We do not compare to CutOut [DT17] in this set of experiments as it will be inappropriate for tumour segmentation. We however include the nnUNet [IJK+21] with 3D input as an additional comparison for this experiment.

The segmentation performance of zonal anatomy was evaluated using the Dice score, Betti error [HLSC19] and maximum Hausdorff distance (mm). We evaluate tumour localisation by determining a true positive if the tumour segmentation overlaps by a minimum of one pixel with the ground truth.

Results

	Baseline	**CutOut**	**BigAug**	**AdvBias**	**RandConv**	**Jigen**	**vMFNet**	**D8h+SheafLoss**
	Transitional Zone							
Dice	0.76±0.12	0.78±0.16	0.82±0.17	0.80±0.18	0.80±0.13	0.77±0.13	0.81±0.13	**0.84±0.09**
Betti Error	1.47±0.48	0.78±0.21	0.69±0.19	0.73±0.20	0.70±0.16	1.00±0.26	0.71±0.22	**0.45±0.16**
HD	22.18±4.07	20.76±3.98	18.76±4.21	19.08±6.67	18.54±4.71	22.21±7.59	17.82±3.31	**15.02±3.13**
	Peripheral Zone							
Dice	0.58±0.22	0.65±0.14	0.68±0.19	0.66±0.20	0.66±0.21	0.59±0.17	0.64±0.19	**0.70± 0.19**
Betti Error	2.55±0.96	1.42±0.67	1.03±0.57	1.45±0.24	1.28±0.20	2.78±1.60	1.27±0.34	**0.83±0.24**
HD	48.23±14.16	40.49±20.39	24.66±10.23	39.65±19.39	28.59±16.99	44.38±22.89	31.02±15.43	**21.43±10.12**

Table 9.5: Dice score, Betti Error and maximum Hausdorff distance (HD) ± standard deviation comparing the D8h shape equivariant+SheafLoss segmentation model trained via contrastive learning to various other SDG methods. This table shows the segmentation results for the prostate zones.

In the task of anatomical segmentation, table 9.5 demonstrates that our method improves segmentation performance in all evaluation metrics compared to the baseline,

	Dice	HD	Betti Error	Sensitivity	Specificity	PPV
Baseline	0.38±0.17	1.03±0.32	5.11±2.90	0.37±0.11	0.60±0.20	0.29±0.11
nnUNet	0.45±0.15	0.81±0.22	4.43±2.32	0.46±0.11	0.70±0.14	0.37±0.14
AdvBias	0.42±0.10	0.90±0.19	4.41±2.16	0.45±0.13	0.66±0.21	0.35±0.17
RandConv	0.43±0.18	0.80±0.27	4.19±2.01	0.47±0.16	0.65±0.17	0.35±0.19
BigAug	0.47±0.12	0.68±0.19	4.03±1.89	0.48±0.18	0.73±0.22	0.40±0.19
Jigen	0.42±0.11	0.88±0.21	4.51±2.43	0.42±0.18	0.65±0.13	0.33±0.09
vMFNet	0.46±0.12	0.80 ±0.20	3.33±1.18	0.47±0.15	0.66±0.21	0.38±0.11
Ours	**0.51±0.13**	**0.57±0.16**	**2.99±0.97**	**0.50±0.18**	**0.79±0.20**	**0.45±0.18**

Table 9.6: The average Dice score, Betti error, maximum Hausdorff distance (HD), sensitivity, specificity and positive predictive value (PPV) ± standard deviations using our method, several SDG methods and the nnUNet for tumour segmentation.

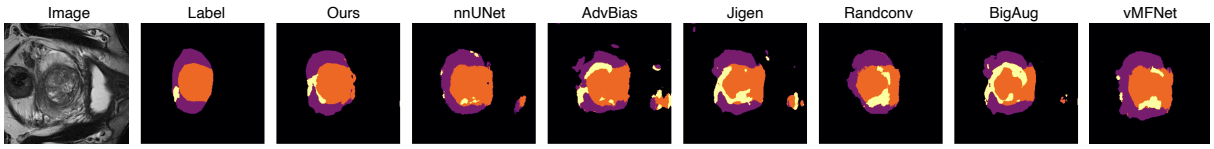


Figure 9.7: Tumour and zonal segmentation of a prostate slice from the RUNMC dataset when training on the internal dataset.

nnUNet and the other SDG methods.

In table 9.6, we note our method significantly improves tumour segmentation and localisation performance. We visualise our findings with an example in figure 9.7, where there is improved localisation of the tumour and the correct number of tumour components enforced by our cellular sheaf loss which is also demonstrated in table 9.6 in the improved Betti error. This significantly reduces the false positive rate highlighted in table 9.6. We also show that our method produces more accurate segmentation of the tumour in the form of improved Dice scores and maximum Hausdorff distances seen in table 9.6. Additionally, note the more anatomically plausible zonal segmentations in figure 9.7. However, our method is restricted by the number of low-dimensional shape components in the shape dictionary used to compose the high-dimensional segmentation output. Therefore, our approach can fail to segment the finer details of prostate tumours due to its high shape variability which leads to coarser but better localised tumour segmentations.

9.5 Conclusion

In this chapter, we tackled a key issue in chapter 8. Firstly, we note that the use of persistent homology as a loss function over an entire segmentation map is location and shape invariant. This approach does not contain geometric information such as curvature and area. We therefore develop a loss function based on modelling the connectivity of a finite number of patches whose composition constructs the segmentation map. We use the notion of a cellular sheaf over a poset to model the connectivity of patches which allows one to extract more accurate localisation information and geometric properties of shapes in a segmentation output. We proved using measure theory that one can increase the amount of localisation and boundary information in our cellular sheaf based function applied to a segmentation output by increasing the number of patches whose disjoint union construct the segmentation. We show this approach improves segmentation accuracy over the classical persistent homology approach on two datasets. Finally, we use our novel loss function to improve the method developed in chapter 7 which learns a shape equivariant dictionary so that shape components are sampled to produce geometrically and topologically plausible segmentation outputs.

Chapter 10

Robust Hierarchical Symbolic Explanations in Hyperbolic Space for Image Classification

This chapter is adapted from the following publication.

- **Santhirasekaram, A.**, Kori, A., Winkler, M., Rockall, A., Toni, F., Glocker, B., 2023. Robust Hierarchical Symbolic Explanations in Hyperbolic Space for Image Classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 561-570).

10.1 Introduction

In this chapter, we will focus on explaining deep discriminative models for images at multiple levels of abstraction, from fine-grained to fully abstract explanations. We will specifically focus on symbolic explanations and why choosing the right curvature in which to embed these symbols is important for developing robust explanations. In this work, we use the natural properties of hyperbolic geometry to more efficiently model a hierarchical relationship of symbolic features with decreased distortion to generate robust hierarchical explanations.

Explainable AI (XAI) aims to improve the transparency and trustworthiness of models [DVBK17, Kro15]; XAI can help with identifying biases, which is important for the safe and fair use of prediction models [KWG+18]. XAI approaches can be broadly categorised into ante-hoc and post-hoc methods [Lip18]. Ante-hoc explainability focuses on developing inherently transparent models [SSK21, LRD+22]. For example, [SSK21] introduces a symbolic reasoning block to promote transparency during training while [LRD+22] uses natural language descriptions that complement model predictions. Post-hoc explanations are the most commonly explored approaches, including explanations via feature-attribution [RSG16, LL17, SGK17], saliency maps [SCD+17, CSHB18, SSP+21] counterfactuals [GWE+19] or concept extraction [GWZK19, GKL+21, KNSK21]. Feature attribution methods [RSG16, LL17] focus on assigning importance weighting to features in input space, indicating their contribution towards the classifier's decision. Saliency-based methods [SCD+17] generate attention maps in input space indicating the image regions responsible for deriving the classifier's decision. These methods by design provide 'single level' explanations as they do not consider any form of reasoning via feature interaction as perceived by humans [AGG83, BH21]. Concept-based explanations [GWZK19, GKL+21, KNSK21] go beyond feature-attribution and saliency-based methods by constructing higher-level concepts indicating their influence on the classifier's decision. [KNSK21, GKL+21] focus on developing traces, which indicate the flow of reasoning to make certain predictions, and thereby considering feature interaction.

Hierarchical concept-based reasoning is the most commonly posited learning principle in systems neuroscience [BH21, VN14, WVT+01, MLF+09]. Inspired by this form of reasoning, we provide the first work to develop hierarchical symbolic explanations for deep discriminative models trained on imaging data. We propose to distil the knowledge in an image classifier into a knowledge tree where nodes represent symbols. One can then transverse the tree to derive hierarchical symbolic explanations in the form of a chain of increasingly abstract symbols leading to the class symbol which we denote as a chain rule. In order to efficiently embed the symbols in the knowledge tree without distortion and overlap, we propose to learn symbols as discrete hyperbolic representations.

10.2 Related Work

Concept learning for image classification has been used as an ante-hoc method towards building more interpretable image classifiers. For example, [AMJ18] develops an encoder to learn a set of discrete concepts and a parallel neural network to learn the relevance of the concepts to making a classification decision. Instead, we want to learn links between concepts. This is related to a popular wave of AI termed neuro-symbolic learning [HM19, VDL+19, MGK+18, YWG+18], where a data-driven deep learning method is used to learn sub-symbolic representations to denote concepts while exploiting symbolic methods to capture reasoning. Inductive logic programming (ILP) is a framework often used for symbolic reasoning with learnt relational theories, including using heuristics and physical properties to understand images [MDS+18]. The solution we propose is closely related to a family of methods called knowledge graph embeddings [SDNT18, TWR+16, ZCY+19] which models relations between discrete entities or symbols for knowledge graph reasoning. NeuralLP [YYC17] can learn a relational path from the subject to the object which, in ILP terms, can be formulated as a chain-like first-order rule. This approach is devised for answering knowledge base queries but will form the basis of our work for deriving chain rules from our generated knowledge trees for explanations.

A natural objective for embedding symbolic data in graphs is for the distances between symbols, defined by the space in which they reside, to correlate with their semantic similarity. Yet, to model increasingly complex relations between symbols, one is bounded by the dimensionality of Euclidean embeddings [NJT14]. This is because the number of nodes generally grows exponentially as the graph distance from the centre node increases, while Euclidean space grows polynomially. This leads to distorted embeddings and information loss [SDSGR18]. Hyperbolic neural networks were proposed to perform all the operations of a neural network in hyperbolic space [GBH18]. This was exploited in the development of hyperbolic graph convolutional neural networks (HGCNN) [CYRL19, LNK19] which forms the structure in our method for reasoning in hyperbolic space.

10.3 Preliminaries

10.3.1 Hyperbolic Transformations

In our work, we apply feature transformations in hyperbolic space, using hyperbolic linear layers [GBH18, CYRL19]. The operations of Mobius addition $\oplus$ and Mobius scalar multiplication $\otimes$ in hyperbolic space can be shown to be analogous to the Euclidean vector space operations of scalar multiplication and addition. [GBH18] proves that Mobius scalar multiplication is equivalent to applying a logarithmic mapping of a point $v \in \mathcal{H}^{d,1}$ to the tangential space at o and multiplying by the scalar r before mapping the scaled point back to hyperbolic space as shown in equation 10.1 below:

$$r \otimes^1 v = \exp_o^{\mathcal{H},1}(r \ \log_o^{\mathcal{H},1}(v)) \tag{10.1}$$

In a hyperbolic linear layer, we also add a bias term b. [GBH18] derives a simple equivalent solution to Mobius addition $(v \oplus b)$ shown in equation 10.2 below. One firstly defines, b on $\mathcal{T}_o\mathcal{H}^{1,d}$ which is parallel transported to the tangential space $\mathcal{T}_v\mathcal{H}^{1,d}$ before mapping back to hyperbolic space.

$$v \oplus^1 b = \exp_v^{\mathcal{H},1}(P^1_{o \to v}(b)) \tag{10.2}$$

We let $W \in \mathbb{R}^{d' \times d}$ and $B \in \mathbb{R}^d$ to define the parameters in a hyperbolic linear feature transformation $h(x)$ for both $\mathbb{H}^{d,1}$ and $\mathbb{B}^{d,1}$ by combining equation 10.1 and equation 10.2 to form:

$$h(x) = (W \otimes^1 x) \oplus^1 B \tag{10.3}$$

10.3.2 Knowledge Tree

In this work, we only consider classifiers of the form $C = f \circ g$, where a feature extractor $f : \mathcal{X} \to \mathcal{E}$ maps input images to the latent space $\mathcal{E}$ and the feature classifier $g : \mathcal{E} \to \mathcal{Y}$ maps the latent space to class labels. We distil the high abstraction knowledge of the continuous feature classifier (g) into a hierarchy of related symbols to form a

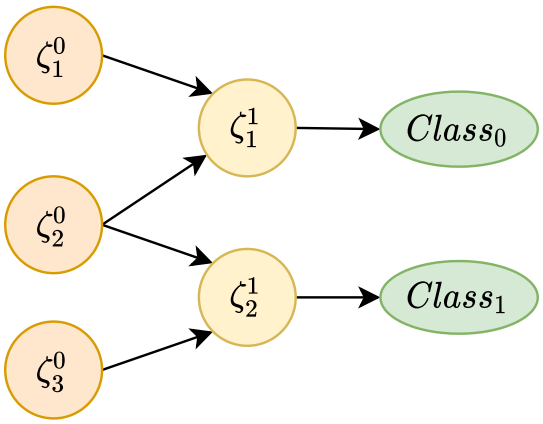


Figure 10.1: Example knowledge tree. Here, symbols ζ_1^0 and ζ_2^0 are being abstracted to form ζ_1^1 which forms class0. Symbols ζ_2^0 and ζ_3^0 are being abstracted to form ζ_2^1 which forms class1.

knowledge tree whereby a symbol is a discrete vector. We assume sufficient training data is available to develop a closed-world knowledge tree.

We denote ζ_j^i to be the j^{th} symbol in the i^{th} level of the knowledge tree. The i^{th} level in the tree is a dictionary of hyperbolic symbols represented as a hyperbolic codebook $\zeta^i \in \mathcal{H}^{M_i \times d'}$; these levels provide different levels of abstraction as shown in figure 10.2. The total number of codebooks n and embedding dimensionalities d are hyper-parameters that are selected based on the use case. $M_0, M_1, \ldots, M_n$ are hyper-parameters corresponding to the total number of symbols in codebooks $\zeta^0, \zeta^1 \ldots, \zeta^n$ respectively. n should be greater than 2 to develop a hierarchy and the greater n, the more levels of abstraction and the 'deeper' the explanations. However, we do not want n to be inappropriately large to prevent the exponential increase in chain rules and reduce interpretability. We strive for an Occam's Razor approach in terms of the number of symbols to generate explanations and carry out ablations to find the smallest number of symbols, M_n required in each codebook before knowledge distillation accuracy from a pre-trained classifier diminishes.

Formally, a knowledge tree is developed by learning the function $\mathcal{K}$ which collapses the Euclidean continuous latent space $\mathcal{E}$ into ideally $\lfloor \log_2 N + 1 \rfloor$ symbols in ζ^n as this represents the minimum number of positive symbols to encode N classes. We can decompose $\mathcal{K}$ such that $\mathcal{K} = \mathcal{R} \circ \mathcal{VQ}$, where $\mathcal{VQ}$ denotes discretisation of $\mathcal{E}$

using vector quantisation[VDOV+17]. $\mathcal{R}$ expresses the hyperbolic symbol abstraction module to produce increasingly abstract hyperbolic symbols. $\mathcal{R}$ can be decomposed into $\mathcal{R}^n \circ \mathcal{R}^{n-1} \circ \ldots \circ \mathcal{R}^1$ with the output of each $\mathcal{R}^l$ producing hyperbolic symbols of different levels of abstraction in the form of ζ^i (see again figure 10.2). We train $\mathcal{K}$ by sampling z from $\mathcal{E}$ and sequentially mapping and discretising z to increasing levels of abstraction in the knowledge tree to produce z_q^i with the final level of abstraction used to classify z.

As a simple illustration, in figure 10.1, we have two levels in the knowledge tree corresponding to codebooks ζ^0 and ζ^1 each with 3 and 2 symbols respectively , which are getting abstracted to form $Class_0$ and $Class_1$. For example, it can be seen that symbols $\zeta_2^0$and ζ_3^0 are getting merged to form an abstract concept ζ_2^1 which is further getting merged with ζ_3^1 to form the higher level abstraction concept $Class_1$. This forms a class-level induced tree which indicates a subtree corresponding to a specific class.

10.3.3 Symbolic Relations

We assume the task of single-label image classification only requires reasoning in a hierarchical manner. Therefore, in our generated knowledge tree, paths which link symbols at the bottom of the tree to the class label at the top of the tree, represent chain rules which implies a class is true for a given image. We formally describe this chain rule in ILP terms. In ILP, rules are usually described in the form of clauses where H is the *head* of the clause and $B_1, \ldots, B_n$ is the *body* of the clause formally denoted as: $H \leftarrow B_1, B_2, \ldots, B_n \quad (n \geq 1)$. A clause of this form captures relational information of the form: the head is true when all the elements in the body are true. Specifically, the first level of the tree indicates whether a symbol exists (*ex* in short) in an image, x. Thereafter, the symbols are being merged to form a more abstract symbol in the next level and therefore an arrow/relation between symbol ζ_i^{n-1} and ζ_i^{n-2} indicates that ζ_i^{n-2} is part of (*pf* in short) symbols ζ_i^{n-1}; $pf(\zeta_i^{n-1}, \zeta_i^{n-2})$. For example, a cat's ear is part of the cat's face. $pf(y, \zeta_i^n)$ indicates symbol ζ_i^n is part of class y. Therefore, a path between the symbols at the bottom of the tree to the top of the tree which are the class symbols can be formally stated in the form of the chain rule shown in equation 10.4 which implies that image x belongs to class y. We formally aim to learn the relations (*ex*, *pf*) in the

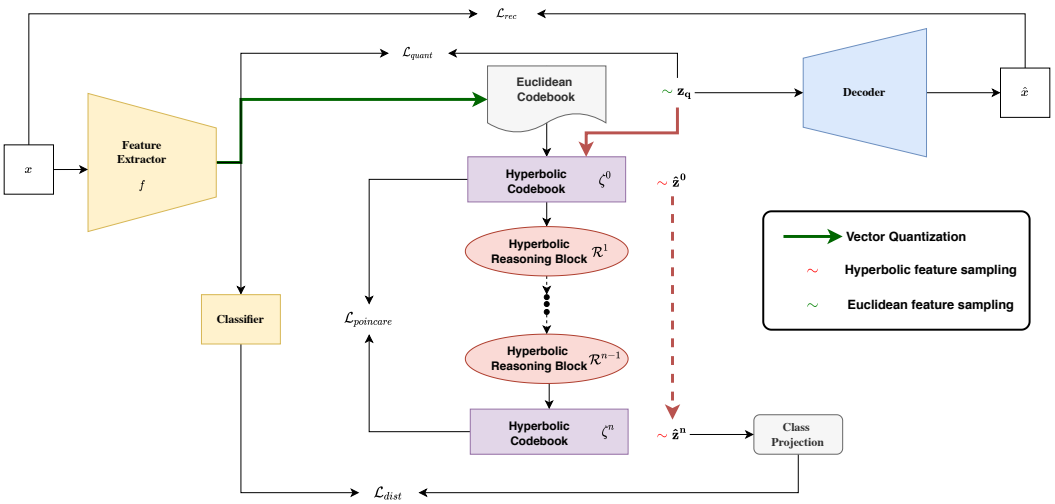


Figure 10.2: Overview of the proposed framework, in which the feature extractor and classifier describe the trained blocks of the given model. The Euclidean codebook forms a discrete representation of the continuous latent space from the feature extractor, followed by hyperbolic codebooks and reasoning blocks to obtain a knowledge tree, which is later used in extracting explanations. The decoder block is independently trained to obtain visual semantics for the extracted hierarchical symbolic rules.

form of the chain rule shown in equation 10.4 which implies that image x belongs to class y if $pf(y, \zeta_i^n)$, $pf(\zeta_i^{n-1}, \zeta_i^{n-2}) \ldots pf(\zeta_i^1, \zeta_i^0)$, $ex(x, \zeta_i^0)$ are all true. The learnt set of chain rules is used to construct a knowledge tree as shown in figure 10.1. This is achieved in our works using a binary neural network [HWG+19].

$$\begin{aligned} class(y, x) \leftarrow & pf(y, \zeta_i^n) \ \wedge \ pf(\zeta_i^{n-1}, \zeta_i^{n-2}) \ \wedge \ldots \\ & pf(\zeta_i^1, \zeta_i^0) \ \wedge \ ex(x, \zeta_i^0) \end{aligned} \tag{10.4}$$

10.4 Methods

10.4.1 Symbol formation

The first step in our framework, symbol formation, is performed by learning discrete symbols in the form of d' dimensional vectors using vector quantisation ($\mathcal{VQ}$) to form a fixed sized Euclidean codebook $\mathbb{C} \in \mathbb{R}^{M_0 \times d'}$ (Figure 10.2). We do not quantise in hyperbolic space as we found this significantly less stable.

Given $(\hat{z}, z) \in \mathbb{R}^{K \times d'}$ and $k \in K$, we define a deterministic process which maps each

embedding vector $z_k \in z$ to the nearest Euclidean codebook vector to form $\hat{z}_k \in \hat{z}$ shown below:

$$\hat{z}_k = \arg\min_j \|z_k - \mathbb{C}_j\|_2, \forall \quad k \in K \tag{10.5}$$

Equation 10.5 defines a sampling process which is non-differentiable but in order to update/learn the symbols which form $\mathbb{C}$ based on this sampling method, we apply straight through gradient approximation. Similar to equation 7.33 in chapter 7, this then allows our discrete surrogate model to be trained end to end with the following quantisation loss [VDOV+17]: $\mathcal{L}_{quant} = \|sg(z) - \hat{z}\|_2 + \beta\|z - sg(\hat{z})\|_2$. This process of sampling the Euclidean codebook is equivalent to learning the *exists* predicate in the first step of the chain rule. Our ablations determined an optimum value of 0.2 for β. Next, we reduce the dimensionality using a linear projection layer to the desired embedding dimensionality d in the hyperbolic codebooks ζ^i before applying an exponential mapping with equation 2.60 to Poincaré space. This leads to the first hyperbolic codebook $\zeta^0 \in \mathbb{B}^{M_0 \times d, 1}$. We chose Poincaré space in this work due to the enhanced visual interpretability of 2D embeddings on the Poincaré disc [NK17]. We however note it may be more beneficial for ζ^i to be embedded in the hyperboloid when d is greater than 2 where the advantage of embedding in Poincaré space is reduced. We additionally observe improved stability and convergence of using hyperboloid codebooks.

10.4.2 Hyperbolic Symbol Abstraction

The goal of the hyperbolic symbol abstraction module ($\mathcal{R}^l$) is to produce increasingly abstract hyperbolic symbols at each level of the tree by merging symbols with edges (*pf* relation) from the previous level, i.e. the hyperbolic symbols for cat eyes and nose are merged to form the symbol, cat face in the next level. ($\mathcal{R}^l$) is similar to a single layer HGCNN [CYRL19] but here we learn the edges of the graph between ζ^i and ζ^{i+1} using a binary function ($1 = edge, 0 = no\ edge$). This constructs a knowledge graph structure equivalent to a tree which is used for reasoning about Euclidean external representations of the visual world. Similar to [CYRL19], the first stage of hyperbolic symbol abstraction is a hyperbolic feature transformation shown in equation 10.3 which is performed in the unit hyperboloid where we found training to be more stable compared to within the Poincaré unit disc/ball. Therefore, we map a codebook ζ^i from

Poincaré space to the hyperboloid using equation 2.66 before applying a hyperboloid linear layer ($h(x)^{\mathbb{H},1}$) to each codebook vector using equation 10.3. This is followed by a logarithmic mapping (equation 2.60) to the tangent space $\mathcal{T}_o\mathbb{H}^{1,d}$. The second stage of hyperbolic abstraction is the aggregation/merging of symbols in $\mathcal{T}_o\mathbb{H}^{1,d}$ as proposed in [CYRL19] to form the next codebook ζ^{i+1}. This is achieved by first learning the edges or equivalently the relations/*partOf* predicate with a binary layer [HWG+19] between symbols in consecutive codebooks with weights: $w_l \in \mathbb{R}^{M_i \times M_{i+1}} \in \{0, 1\}$.

We learn a weighted aggregation denoted a_l of the symbols with edges from ζ^i to ζ^{i+1}, before mapping back to Poincaré space (equation 2.57). This aggregation is an approximation of the Frechet mean in hyperbolic space as assumed in [CYRL19]. A single pass through $\mathcal{R}^l$ is summarised in equation 10.6 below.

$$\zeta^{i+1} = \exp_o^{\mathbb{B},1}(a_l^\intercal \ \log_o^{\mathbb{H},1}(h^{\mathbb{H},1}(\psi_{\mathbb{B}^{d,1}\rightarrow\mathbb{H}^{d,1}}(\zeta^i)))) \tag{10.6}$$

For the hyperbolic symbol abstraction module to update its weights to form an accurate knowledge tree, it needs to map each feature in $\hat{z}^0$ to the correct class. Firstly, we apply an exponential mapping of $\hat{z}^0$ to the Poincaré unit ball defined in equation 2.60 in order to map every feature in $\hat{z}^0$ to the nearest codebook vector by Poincaré distance in ζ^0 using equation 2.53 to then form $\hat{z}^1$. This process is repeated sequentially for every hyperbolic codebook until the last codebook where z_q^n is formed by sampling ζ^n. The abstraction process is completed by mapping z_q^n into Euclidean space with a logarithmic mapping and then applying a linear class projection layer to map to the class prediction. As this process is done for each of the k features in $\hat{z}^1$, we have k chain rules. However, there are multiple chain rules which are repeated and therefore are removed to form a unique set of chain rules for each image. Note, unlike in $\mathcal{VQ}$ for symbol formation, we do not directly update the symbols in ζ_i so that it is only learned by a function ($\mathcal{R}^l$) of ζ_{i-1} in order to faithfully generate a knowledge tree. Therefore, straight-through gradient approximation is not required in this abstraction process.

The process of sequentially sampling each hyperbolic codebook by Poincaré distance is equivalent to learning to extract the best chain rules represented as a set of discrete vectors transversing the hyperbolic knowledge tree to classify the image. The notion of sampling rules based on distance also allows us to rank the best rules as well as

ascertain uncertainty over the rules.

The knowledge distillation loss is defined as the cross-entropy loss between the classifier prediction (y) and the hyperbolic discrete surrogate model's prediction ($\hat{y}$): $\mathcal{L}_{dist}(\hat{y}, y)$. We determine that the Poincaré distances between codebooks correspond to graph distances in the tree. Therefore, a Poincaré codebook loss $\mathcal{L}_{Poincare}$ is calculated such that symbols with an edge are closer together and those without an edge are pushed apart in hyperbolic space. First, let u to be any symbol in the set of all ζ^i while v and v' are defined as symbols with and without an edge with u respectively; then two sets $\mathcal{P}$ and $\mathcal{W}$ are created such that: $u, v \in \mathcal{P}$ and $u, v' \in \mathcal{W}$. Given this, we can calculate the Poincaré codebook loss shown in equation 10.7.

$$\mathcal{L}_{Poincare} = \frac{\sum_{u,v \in \mathcal{P}} e^{d^{\mathbb{B},1}(u,v)}}{\sum_{u,v' \in \mathcal{W}} e^{d^{\mathbb{B},1}(u,v')}}. \tag{10.7}$$

We now define our total training loss as:
$\mathcal{L}_{Total} = \mathcal{L}_{dist} + \mathcal{L}_{quant} + \mathcal{L}_{Poincare}$.

Continuous weights in our framework are updated with Adam optimisation while the binary weights in $\mathcal{R}^l$ are updated using the *Bop* algorithm proposed by [HWG+19]. In our case, we map our weights to {0, 1} rather than {-1, 1}.

10.4.3 Explanations and Visual Semantics

We derive a unique set of chain rules for each class to form a class tree. An explanation for the classification of an image is derived in the form of a unique set of chain rules sampled from the knowledge tree and the visual semantics for symbols. These chain rules can be combined to form an image-level tree.

For example, in figure 10.1, the subtree for $Class_0$ consists of the chain rules: $class(y, x) \leftarrow pf(y, \zeta_1^1) \wedge pf(\zeta_1^1, \zeta_1^0) \wedge ex(x, \zeta_1^0)$ and $class(y, x) \leftarrow pf(y, \zeta_1^1) \wedge pf(\zeta_1^1 \zeta_2^0) \wedge ex(x, \zeta_2^0)$. An image-level tree is input-dependent, and if correctly classified forms a subtree of the class-level tree for the ground truth class or in other words a subset of the unique chain rules for the class label.

The extraction of visual semantics for symbols first requires us to train a decoder $\mathbb{D}$ to reconstruct images (x) in Euclidean space as perceived by the classifier, with a reconstruction loss defined as: $\mathcal{L}_{recon} = \|\mathbb{D}(z_q) - x\|_2^2$. During training, we make sure that the gradients from the decoder block do not affect the weights of the discrete surrogate model, to maintain faithfulness of the discretisation process.

In order to explain the semantic meaning of any symbol $\hat{z}_j^i \in \zeta^i$, we must project the symbol onto $\mathbb{C}$ and visualise the semantic difference using a decoder $\mathbb{D}$. In the first step to compute the semantic difference $\delta(\hat{x})$, we get all the symbols in $\hat{z}_j^0 \sim \zeta^0$ responsible for activating $\hat{z}_j^i \in \zeta^i$ using the image-level tree to form a set $\mathcal{S} \subset z_q^0$. ζ^0 has a one to one mapping with $\mathbb{C}$ such that one can now form a new set $\mathcal{A} \subset z_q$. We use $\mathcal{A}$ to perform a controlled intervention on $z_q \sim \mathbb{C}$ which is inputted into $\mathbb{D}$ to calculate $\delta(\hat{x})$. We can formally define this process below in equation 10.8.

$$\delta(\hat{x}) = \mathbb{D}(z_q) - \mathbb{D}(z_q; do(z_{qk} = 0), \forall \quad z_{qk} \in \mathcal{A}), z_{qk} \in z_q. \tag{10.8}$$

10.5 Training details and architectures

10.5.1 Discrete surrogate model

We consider an image dimension of 32×32 for the MNIST dataset. The image dimensions for the AFHQ and STL10 datasets is 128×128. The image dimensions for the MIMIC dataset is 224×224. Our modulation layer reduces the number of features (latent dimension) of z from 64 to 8 channels for the MNIST experiments and 1024 to 32 channels for the AFHQ, STL10 and MIMIC experiments. We perform all hyperbolic linear operations in $\mathcal{T}_x\mathbb{H}$ and do not directly update the embeddings $\zeta^i \in \mathbb{H}^{1,d}$ or $\in \mathbb{B}^{1,d}$ which means we can perform Euclidean optimisation. Therefore, for the experiments in the main paper, we use Adam optimisation with a learning rate of 0.0002 and batch size of 50 in all experiments to train the discrete surrogate model for 40 epochs on a single NVIDIA RTX 2080 GPU. We experimented with Riemmanian stochastic gradient descent [ZJRS16] but found Adam optimisation to be more stable.

Note, we do not perform linear operations in $\mathcal{T}_x\mathbb{B}$ due to stability issues. Also, note

we do not directly update the embeddings of ζ^i in $\mathbb{H}^{1,d}$ or $\mathbb{B}^{1,d}$ as the embeddings are updated only as a function of the learned hyperboloid linear operations in $\mathcal{T}_x\mathbb{H}$ followed by mapping to $\mathcal{T}_x\mathbb{B}$.

The only embeddings we update directly through vector quantisation are in the Euclidean codebook $\mathbb{E} \in \mathbb{R}^{d' \times M_0}$. [CYRL19] applies hyperboloid non-linear activation to aid with the smoothly varying curvature. We do not include non-linearity in our hyperboloid linear layer which uses fixed negative curvature as it did not add any benefit to training convergence or knowledge distillation performance.

10.5.2 Binary Weights

In the *boop* [HWG+19] algorithm, the strength of the gradient signal at time t is determined by looking at the continuous exponential moving average m_t of accumulated gradients up to the gradient, g_t at t. The binary weights are then updated subject to m_t exceeding a threshold τ and the sign of $w_{t,l}^{j,k}$ matching m_t. We initialize weights randomly $\in \{0, 1\}$ and modify the update rule proposed in [HWG+19]. This is shown in equation 10.9 below, which calculates the continuous exponential moving average with γ being the adaptivity rate [HWG+19] and equation 10.10 defines the update rule [HWG+19]. By following this update rule, we ensure that weights are either 0 or 1 while at the same time, following all the properties described in [HWG+19].

$$m_t = (1 - \gamma)m_{t-1} + \gamma g_t \tag{10.9}$$

$$w_{t,l}^{j,k} = \begin{cases} |w_{t-1,l}^{j,k} - 1|, & \text{if } |m_t^i| > \tau \text{ and} \\ & \text{sign}(m_t^i) = \text{sign}(w_{t-1,l}^{j,k}) \\ w_{t-1,l}^{j,k}, & \text{otherwise} \end{cases} \tag{10.10}$$

In our experiments, we use an adaptivity rate γ of 0.0004 and threshold τ of $1e-8$. The adaptivity rate is analogous to the learning rate and can be seen as the consistency of a gradient signal required to induce a flip in weights from 0 to 1 or vice versa [HWG+19]. A high γ can therefore induce a flip quicker given a new gradient signal but this can also mean noisy training [HWG+19]. τ is reflective of the strength of the gradient signal

required to induce a flip [HWG+19].

10.5.3 Decoder

The Decoder for MNIST experiments initially consists of a 1×1 convolutional block to increase the number of channels back to 64 channels before 3 up-sampling blocks. Each up-sampling block consists of bi-linear interpolation before two pre-activation 3×3 convolutional blocks with residual connections [HZRS16b]. We use batch normalisation and ReLU activation. In our AFHQ, STL10 and MIMIC experiments, the decoder also initially starts with a 1×1 convolutional block to increase the number of channels back to 1024 before 4 up-sampling blocks identical to the one used in our MNIST experiments. We train the decoder in both experiments for 40 epochs using Adam optimisation with a learning rate of 0.0002.

10.5.4 Pre-trained classifiers

The MNIST pre-trained classifier consists of 7 convolutional blocks made up of a 3×3 convolutional layer followed by batch normalisation and ReLU non-linearity. This is followed by global average pooling and a single-layer linear classifier. Max pooling is applied after the first, third and fifth layers. The number of channels corresponding to the final convolutional block is 64. We train this classifier for 50 epochs with a batch size of 50 on three NVIDIA RTX 2080 GPUs. We use Adam optimisation with an initial learning rate of 0.001 and weight decay of 0.001. We achieve 99% accuracy with this pre-trained classifier

We use the DenseNet-121 as our pre-trained classifier for the AFHQ, STL10 and MIMIC dataset [IMK+14]. We train using Adam optimisation with an initial learning rate of 0.001 and weight decay of 0.05. We reduce the dimensionality of the MNIST, AFHQ, STL10 and MIMIC images to 4×4 in the final layer before being inputted into the classifier.

10.6 Experiments

We use our framework to explain a model pre-trained on the MNIST dataset [Den12] achieving 98% accuracy as a proof of concept study. In order to indicate the scalability and generalisability of our proposed method we explain models pre-trained on AFHQ [CUYH20], STL10 [CNL11] and the MIMIC chest x-ray dataset [JPB+19, JPG+19], which achieved 98%, 97% and 81% accuracy respectively. We choose a 3-level hierarchy.

10.6.1 Qualitative Experiments

We compare the explanations obtained from our framework against standard post-hoc explainability frameworks: LIME [RSG16], SHAP [LL17], deepLIFT [SGK17], and gradCAM [SCD+17]. We note the far richer and more expressive explanations by our method shown on the right of figure 10.3, 10.4 and 10.5 where we show hierarchical visual explanations.

Figure 10.3 demonstrates our explanations on a pre-trained classifier for MNIST. Figure 10.3(b) shows the class-level *global-tree* representing all possible chain rules for a particular class. To explain a given image we construct an image-level *local-tree* indicating the sampled chain rules which are shown in figure 10.3(c). Figure 10.3(d) provides a visual hierarchical description of an image-level tree. In this example, the visual semantics for the symbolic relations obtained for a given image x corresponding to $pf(Class2,\ \zeta_0^2)$ and $pf(Class2,\ \zeta_3^2)$ is shown in the first row of figure 10.3(d). The second row of figure 10.3(d) visualises the obtained symbolic relations corresponding to $pf(\zeta_3^2,\ \zeta_{21}^1)$ and $pf(\zeta_3^2,\ \zeta_{22}^1)$. The last row visualises sampled symbols from ζ^0 which are part of ζ_1^{21}. As we move down the level of the hierarchy or, in other terms, as we move closer to the boundary of Poincaré space, visually the symbols start to move from a complete digit heatmap to a more focused region in a digit, demonstrating a visual hierarchical explanation. One can also note the distinct semantic differences between individual symbols highlighting the advantage of discretising features.

Similar explanations and visual semantic behaviour can be observed for models trained on the AFHQ and STL10 datasets, as shown in figure 10.4 and figure 10.5 respectively,

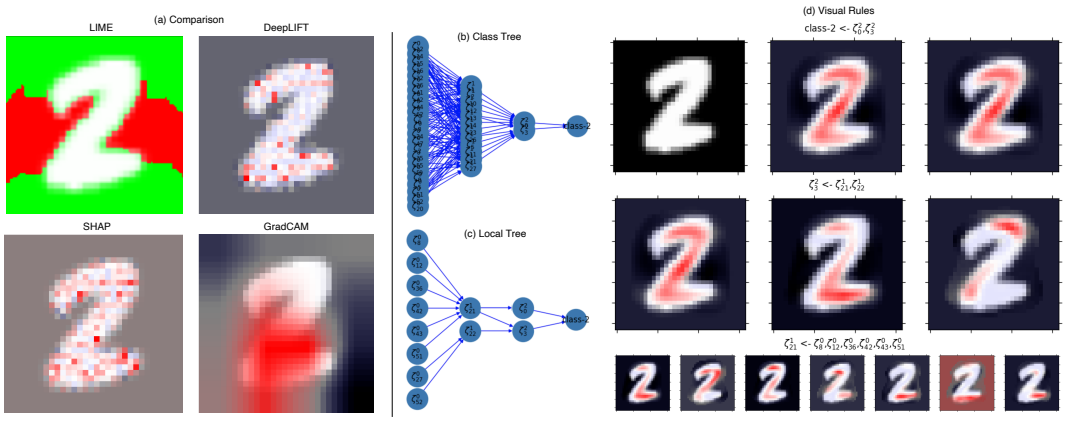


Figure 10.3: Explanations obtained using the proposed framework for an MNIST classifier. (a) Shows the 4 post-hoc XAI methods we compared with. (b) Demonstrates the obtained class-level tree, which is a complete set of FOL chain rules responsible for 'class 2', (c) indicates an image-level tree formed from the sampled chain rules responsible for making a decision for a given image, and (d) demonstrates a subset of the visual $PartOf$ relations obtained from an image-level tree.

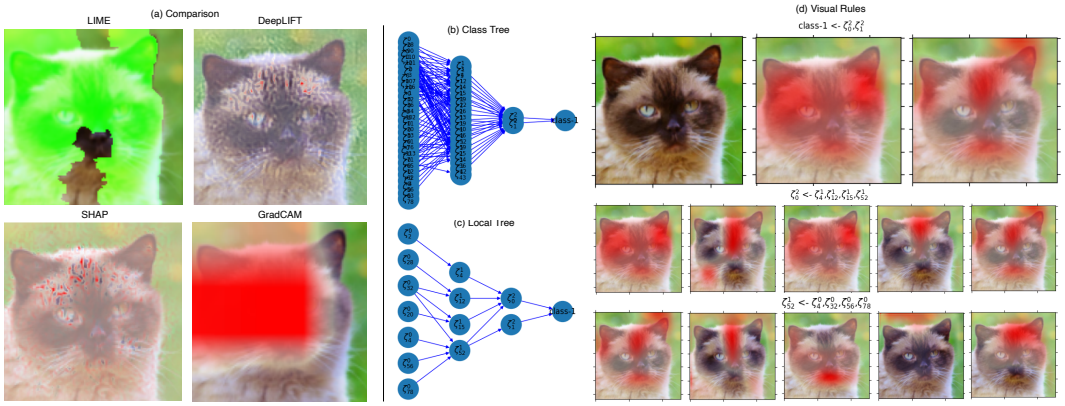


Figure 10.4: Explanations obtained for an AFHQ classifier deciding upon the 'class cat'. The subset of the visual $PartOf$ relation is shown.

where we again observe symbols getting localized as we go deeper into the hierarchy. Additionally, figure 10.6 shows explanations for a model trained on the MIMIC dataset for the class 'abnormal' where we again note increasingly abstract symbols being sampled down the hierarchy. We note most of the existing explanation methods shown on the left of figure 10.3, 10.4 and 10.5 focus on either pixel importance or gradient-based attention to provide only single level explanations. These explanations do not yield any form of reasoning between the features and the class.

Our method goes beyond feature attribution by allowing the user to decide on the level of abstraction (length of chain rule) upon which to provide the symbolic and corresponding visual semantic relationships.

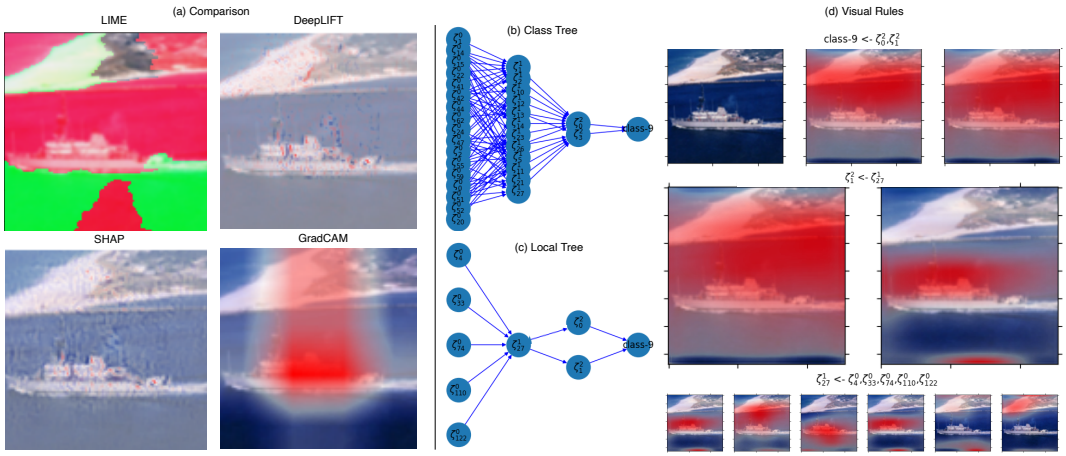


Figure 10.5: Explanations obtained using the proposed method for an STL10 classifier deciding upon 'class 9 (Boat)'.

10.6.2 Robustness Experiments

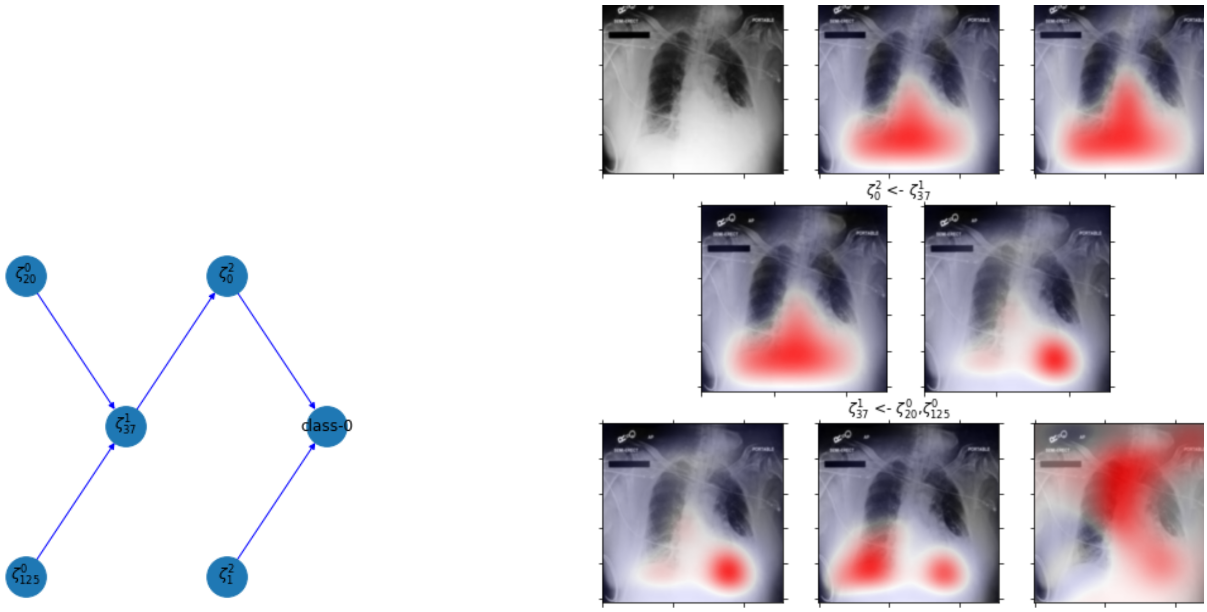
We evaluate the robustness of our method compared to LIME [RSG16], SHAP [LL17], deepLIFT [SGK17], and gradCAM [SCD+17] by measuring the change in explanations under noise perturbations in the input space. Specifically, we measure the average variance in the heatmaps (visual semantics) generated by our method at the last level of abstraction compared to popular post-hoc methods under 1%, 5%, 10% and 30% Gaussian, Salt and Pepper (s&p) and Poisson noise. We show in table 10.1 that there is significantly less variance in the explanations generated by our method with the addition of Gaussian noise, highlighting the robustness of our explanations. We note similar results for s&p and Poisson noise in tables 10.2 and 10.3 respectively.

	MNIST	STL10	MIMIC	AFHQ
LIME [RSG16]	0.236	1.341	0.785	1.109
SHAP [LL17]	0.923	1.514	1.143	1.381
deepLIFT[SGK17]	0.535	0.483	0.253	0.585
gradCAM[SCD+17]	1.477	1.664	1.966	1.644
Ours	$\mathbf{1e^{-6}}$	$\mathbf{2e^{-5}}$	$\mathbf{4e^{-5}}$	$\mathbf{1e^{-5}}$

Table 10.1: Average variance in the heatmaps generated by various post-hoc explainability methods under Gaussian addition.

	MNIST	STL10	MIMIC	AFHQ
LIME [RSG16]	0.785	1.342	1.178	1.157
SHAP [LL17]	1.911	13.124	6.746	14.494
deepLIFT[SGK17]	0.935	6.759	5.903	3.185
gradCAM[SCD+17]	52.472	66.441	19.664	55.488
Ours	$\mathbf{8e^{-7}}$	$\mathbf{2e^{-6}}$	$\mathbf{1e^{-5}}$	$\mathbf{1e^{-5}}$

Table 10.2: Average variance in the heatmaps generated by various post-hoc explainability methods under Salt and Pepper (S&P) noise addition.



(a) Image-level tree

(b) Visual rules

Figure 10.6: This figure describes the hierarchical visual explanations obtained using our proposed framework for the MIMIC classifier classifying class 'abnormal(class 0)', with atoms corresponding to $pf(class0, \zeta_0^2), pf(class0, \zeta_1^2)$ described in the first row. The second row visualises the obtained atoms corresponding to $pf(\zeta_0^2, \zeta_{37}^1)$. The last row visualises the atoms corresponding to symbols in ζ^0 which are part of ζ_{37}^1.

	MNIST	STL10	MIMIC	AFHQ
LIME [RSG16]	0.206	0.747	0.577	1.132
SHAP [LL17]	0.147	0.611	0.483	1.943
deepLIFT[SGK17]	0.458	1.173	0.563	1.477
gradCAM[SCD+17]	0.966	1.100	1.074	1.094
Ours	$\mathbf{1e^{-6}}$	$\mathbf{1e^{-5}}$	$\mathbf{1e^{-5}}$	$\mathbf{1e^{-5}}$

Table 10.3: Average variance in the heatmaps generated by various post-hoc explainability methods under Poisson noise addition.

A visual example to demonstrate our finding is shown in figure 10.7 where under Gaussian noise addition the heatmaps generated by our method show no change compared to the other XAI methods. We show a further visual example of the robustness of our method under 30% Gaussian noise for an MNIST dataset example in figure 10.8 and under 30% salt and pepper noise for a MIMIC dataset example in figure 10.9. In figure 10.10, we finally demonstrate an example of the robustness of our method under 20% Poisson noise for an image in the STL10 dataset.

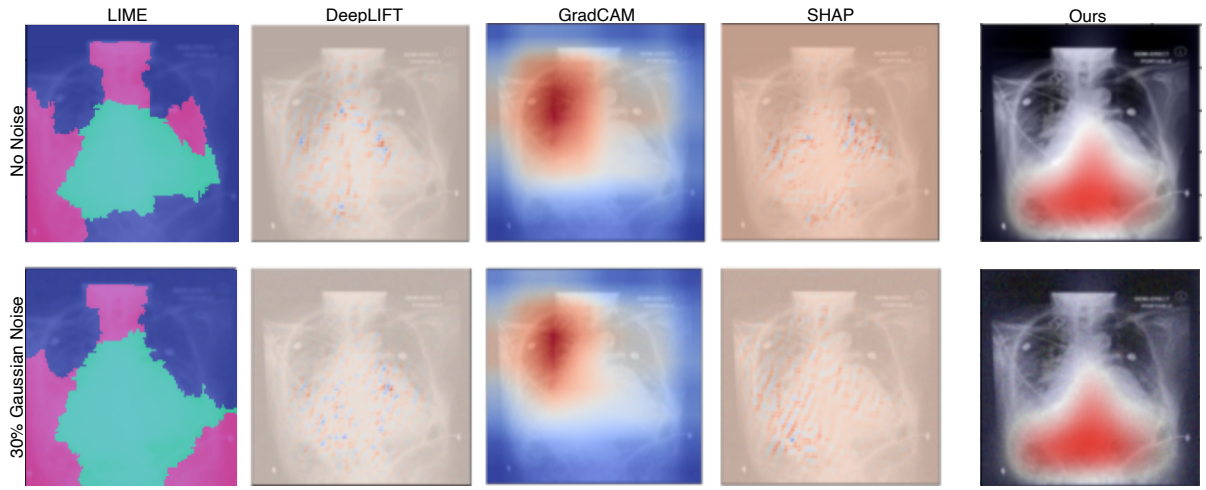


Figure 10.7: Robustness comparison of considered methods with our proposed method. In our method, we show the visual semantic of a single symbol sampled for the last codebook; ζ_0^2. The top shows the heatmaps with no noise added. The second row shows heatmaps with 30% Gaussian noise added.

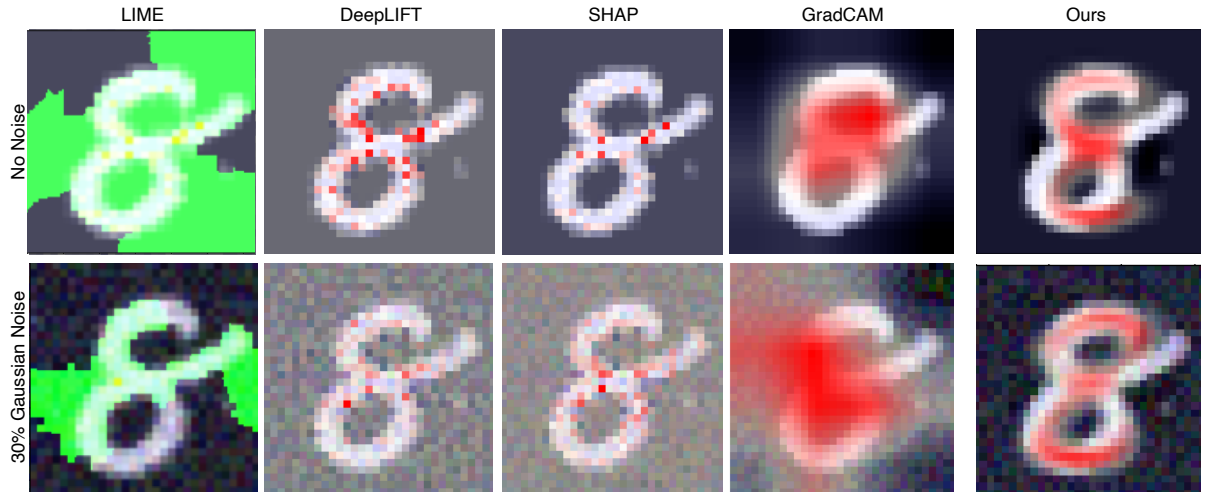


Figure 10.8: Robustness experiments with 30% Gaussian noise on a sample from the MNIST dataset. The symbol for ζ_3^2 is shown. The image level tree did not change under 30% Gaussian noise for this example.

10.6.3 Hyperbolic vs Euclidean Experiments

We hypothesised that hyperbolic embeddings will better embed a knowledge tree without distortion and hence allow for a reduction of the dimensionality d of ζ^i such that knowledge distillation would not be affected. We support this hypothesis by achieving better knowledge distillation accuracy with Poincaré embeddings highlighted in table 10.4. We note an increasingly wider margin in performance between Poincaré and Euclidean embeddings as we reduce the dimensionality of ζ^i down to 2 in all

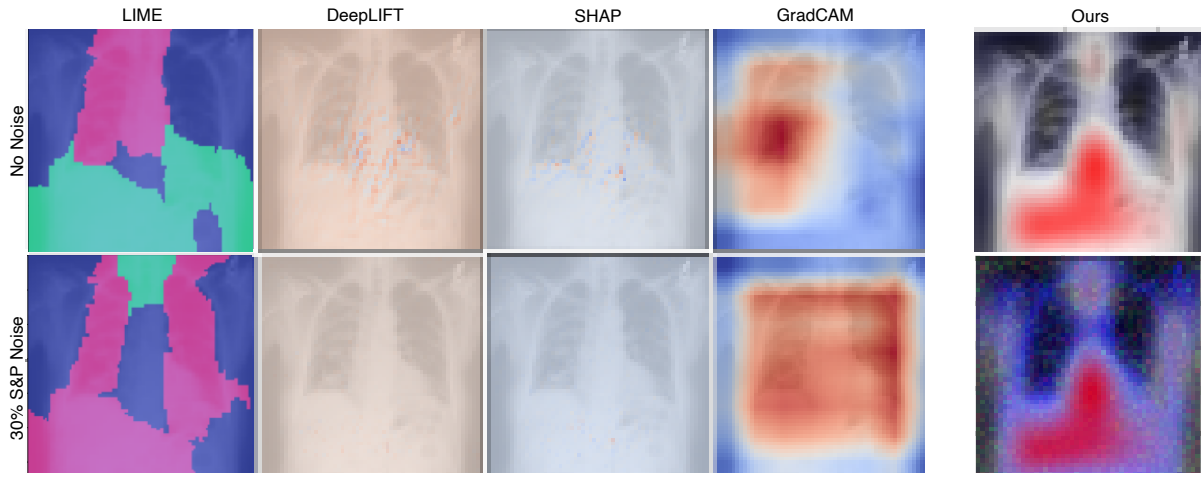


Figure 10.9: Robustness experiments with 30% salt and pepper noise on a sample from the MIMIC dataset. The symbol for ζ_0^2 is shown. The image level tree did not change under 30% S&P noise for this example.

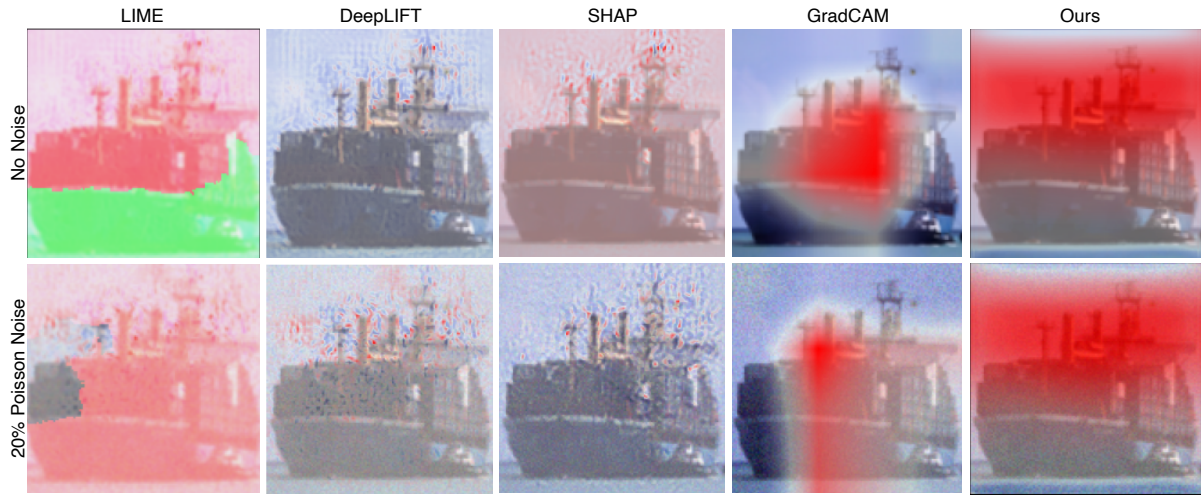


Figure 10.10: Robustness experiments with 20% Poisson noise on a sample from the STL10 dataset. The symbol for ζ_0^2 is shown. The image level tree did not change under 20% Poisson noise for this example.

4 datasets. Figure 10.11 shows the 2-dimensional embeddings on the Poincaré disk for the MNIST dataset maintaining a robust hierarchy. Furthermore, we are also simultaneously showing in table 10.4 that hyperbolic symbolic abstraction via a higher knowledge distillation accuracy leads to a more accurate explanation of the classifier.

Our hypothesis also implies that reasoning hierarchically in hyperbolic space should lead to less overcrowding of concepts at the first level of reasoning. One can show this by measuring the overlap using the Dice score between all the visual semantics in the first abstraction layer corresponding to all symbols sampled from ζ^0 for each image. Table 10.5 demonstrates less overlap (lower Dice score) between concepts in hyperbolic

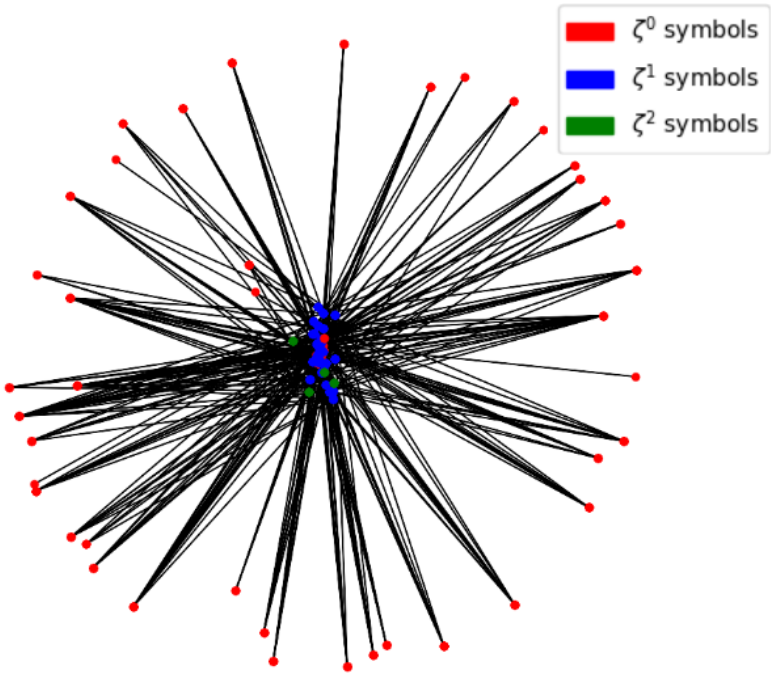


Figure 10.11: 2D Poincaré embedding of symbols obtained for MNIST. Red, blue, and green nodes indicate symbols from $\zeta^0, \zeta^1, \zeta^2$ respectively.

space which becomes more apparent as one reduces the dimensionality of embeddings compared to Euclidean space.

Emb. dim → Dataset ↓	Poincaré			Euclidean		
	2	4	16	2	4	16
MNIST	**0.90**	**0.96**	**0.99**	0.81	0.92	0.95
AFHQ	**0.90**	**0.95**	**0.98**	0.80	0.90	0.97
STL10	**0.84**	**0.87**	**0.88**	0.72	0.82	0.86
MIMIC	**0.78**	**0.83**	**0.84**	0.71	0.80	0.80

Table 10.4: Knowledge distillation accuracy of different dimensional Euclidean and Poincaré embeddings.

Emb. dim → Dataset ↓	Poincaré			Euclidean		
	2	4	16	2	4	16
MNIST	**0.23**	**0.19**	**0.20**	0.33	0.24	0.21
AFHQ	**0.46**	**0.21**	**0.17**	0.72	0.31	0.28
STL10	**0.41**	**0.17**	**0.16**	0.60	0.23	0.19
MIMIC	**0.49**	**0.30**	**0.32**	0.55	0.31	**0.32**

Table 10.5: Average Dice score overlap between the visual semantics of symbols sampled from ζ^0 for different dimensional Euclidean and Poincaré embeddings

10.6.4 Codebook Ablations

In table 10.6, we show the minimum number of codebook vectors required to achieve a knowledge distillation accuracy of at least 90%. We limit the number of codebook

vectors in ζ^n to at least $\lfloor log_2 N + 1 \rfloor$. We note in more complex datasets, a larger number of codebook vectors may be required in ζ^n, whereby a class may need to be encoded with more than one possible combination of codebook vectors in order to fit the data.

Codebook	MNIST	AFHQ	STL10	MIMIC
ζ^1	128	256	256	256
ζ^2	32	64	64	64
ζ^3	4	3	4	2

Table 10.6: Codebook ablations to develop the most sparse knowledge tree.

10.7 Conclusion

This work provides novel hierarchical explanations for deep discriminative models, demonstrated on several datasets. The proposed framework discretises the continuous latent space of classifiers into discrete features, followed by multiple layers of symbolic abstraction in hyperbolic space to form a knowledge tree which provides hierarchical chain rules as explanations. We demonstrate that hyperbolic geometry allows to embed our knowledge tree with minimal distortion and hence prevent the overcrowding of concepts compared to the Euclidean counterpart.

The results show the existence of a consistent and robust set of chain rules for each class, visualised by generating attention regions in an image which are more robust compared to traditional post-hoc methods.

For future work, our framework can be developed into a stand-alone interpretable deep discriminative neuro-symbolic model which improves generalisability. We plan to extend this method with domain experts and assign human-interpretable meaning to symbols.

Chapter 11

Conclusions

We conclude this thesis by returning to the research objectives in section 1.2 of chapter 1 which is also summarised in figure 1.1.

11.1 Research Objectives

11.1.1 Research Objective 1

We firstly fulfilled our first objective in section 1.2 by proposing and showing that one can improve the robustness of deep learning models for classification and segmentation by constraining the latent space to a finite number of components using vector quantisation.

11.1.2 Research Objective 2

We then demonstrated the importance of embedding these components in a latent space equipped with the correct curvature. Specifically, we improve the robustness of segmentation models by imposing hierarchical compositionality on a hyperbolic manifold in the latent space.

We also showed the importance of considering the relational structure between symbolic features for classification and the natural curvature best equipped to embed this structured symbolic representation. Specifically, we developed robust hierarchical explanations for deep discriminative models by generating the explanations in hyperbolic space.

11.1.3 Research Objective 3

Next we demonstrated that robustness can also be improved by imposing a discrete shape equivariant latent space for segmentation using either a constrastive based and kernel constraint method. We also then showed the dependence of robustness on the order of the group. This satisfies the third research objective.

11.1.4 Research Objective 4

We then demonstrated the importance of embedding the shape components in a latent space equipped with correct topology and geometry to improve robustness further which satisfies the fourth research objectives. We specifically explored the importance of enforcing the correct topology after sampling a finite latent space using persistent homology. We then highlight the disadvantages of persistent homology for capturing shape information and improve on this by using cellular sheaves.

11.1.5 Research Objective 5

We empirically demonstrated the advantage of our approach throughout this thesis by comparing to a number of methods used to improve model robustness in the literature including data augmentation, adversarial and self-supervised learning. We showed our methods competes favourably with these methods using simple geometric and topological constraints in the latent space which are also computationally less expensive. This satisfies the final research objective.

11.2 Summary of Contributions

We will now highlight a summary of contributions from each chapter. The following summaries are adapted from publications stated in each of the contribution chapters.

11.2.1 Chapter 4

.

- We propose a new framework to improve the robustness of vision transformer-based models for disease classification by constructing discrete representations of the data using vector quantisation. We sample a subset of the discrete representations in either Euclidean or hyper-spherical Space to form the input into a transformer-based model.
- We use cross-attention in our transformer model to combine discrete representations of T2-weighted and ADC images.
- Our approach achieves SOTA performance for the classification of lesions on prostate MRI and outperforms various other CNN and transformer-based models in terms of robustness to domain shifts and perturbations in the input space.

11.2.2 Chapter 5

- We focus on enhancing the robustness of segmentation network architectures by transforming input images into a quantised lower-dimensional embedding space.
- We substantiate this approach with a comprehensive rationale, supported by a well-defined set of assumptions. Additionally, we establish an empirically derived upper bound on the permissible shift in the latent space caused by input perturbations to ensure robustness. Finally, we validate our proposition through a series of experiments showcasing the enhanced robustness and performance of the widely used UNet architecture [RFB15] with a quantised latent space.

11.2.3 Chapter 6

- We enhance the robustness of deep learning-based segmentation models by introducing hierarchical compositionality in the latent space.
- This is the first contribution to leverage the curvature of the latent space for learning a dictionary of base components through quantisation. Specifically, we explore negative curvature (Poincare space) as the optimal choice to foster hierarchical compositionality.

11.2.4 Chapter 7

- We propose to constrain the latent space to a dictionary of shape components which is sampled to construct the segmentation output. We impose equivariance of each component in the dictionary to various discrete or continuous groups by either using a contrastive approach or imposing group equivariant constraints on the convolutional kernel itself. We hypothesise this will improve the generalisability of any segmentation model which maps the input space, to a lower dimensional embedding space, using an encoder, before mapping to the segmentation output, with a decoder.
- We showed that by increasing the density of the group equivariant constraint by increasing the order of the group, the robustness of a segmentation model improves.
- We evaluate the capability of our method to improve single domain generalisability in the task of prostate zonal segmentation with two labels (transitional and peripheral zone) and cardiac segmentation with three labels (myocardium, left ventricle and right ventricle).

11.2.5 Chapter 8

- We introduce a topological prior via persistent homology during the sampling of the learned shape dictionary. This ensures that the components are not independently sampled (as with a uniform prior), facilitating the creation of topologically meaningful segmentations.

- Additionally, we are the first to effectively employ persistent homology for providing deep supervision to deep learning-based segmentation models.

11.2.6 Chapter 9

- Our method introduces the concept of a cellular sheaf over a poset to characterise the connectivity of patches in a segmentation map. We demonstrate that this approach offers a more comprehensive geometric representation of shapes compared to existing methods that rely on persistent homology.
- We devise a loss function based on cellular sheaves to enhance the segmentation performance of the model proposed in chapter 7 which learns a shape equivariant dictionary in the latent space. This loss function ensures that shape components are sampled to generate segmentation outputs that are both geometrically and topologically coherent.

11.2.7 Chapter 10

- We present an approach to discretise the continuous latent space of a classifier model into a hierarchy of discrete vectors representing symbols. Leveraging the inherent structure of hyperbolic geometry, we model this hierarchy more effectively.
- We develop a method for learning hierarchical symbolic relationships by utilizing binary weight layers to construct a knowledge tree. This enables the extraction of a collection of robust local image-level and global class-level chain rules, serving as explanations for deep discriminative models used for image classification.

In conclusion, this thesis highlights the importance of compositionality, curvature, equivariance and topology in building robust representations for downstream tasks in medical imaging. Among the constraints we have explored in this these to build more robust representations, group equivariant constraints have proven to the most valuable and consistently reliable to constructing robust models. This is because group equivariance provides the most general notion of preserving symmetry between the input and discrete latent space and therefore the least restrictive. One

is also allowed greater flexibility in the degree of the group equivariance imposed which allows one to control the strength of the group equivariant assumption on the structure of the latent space.

11.3 Future Work

This thesis has laid the foundations for understanding the role geometry and topology of the latent space plays in building robust representations for non-adversarial robustness. We hope the research community will take inspiration from the tools developed and theorems proposed to now explore the importance of geometry and topology in adversarial robustness as this work can help provide further insight into adversarial learning from a geometric and topological perspective.

We believe this work can also form the foundations to use the notion of topology and compositionality in object-centric learning. For example, one can use the language of cellular sheaves to model compositionality and relationships between objects for object-centric learning. We foresee identifiability as being crucial for building robust models and hope this thesis will inspire the use of the language of category theory and cellular sheaves for constructing identifiable representations.

We believe power of cellular sheaf modelling can be extended to a number of other applications to medical image analysis. One specific application, is to model motion such as cardiac motion using sheaves such that both local and global and topological corrected is maintained throughout the cardiac cycle.

We find topological losses are powerful and can also be interpreted as a counting loss function if we consider the number of connected components. This is very useful in clinical practise and can used for accurate detection tumours by enforcing the correct number of tumours detected. This can serve to reduce the false positive rate of tumour detection models.

We can also extend our topological insights into robust representation learning by using persistent homology to discover the shape and intrinsic dimension of the underlying data manifold. This provides a useful tool to distinguish data generated from certain

manifolds. For example, its natural to hypothesis the manifold structure and intrinsic dimension of AI generated and natural data varies and we plan to use our approach to detect AI generated data.

Symmetric breaking between the input and latent space also serves as valuable to detect anomalies in the dataset. Therefore, we hope to develop a method using group equivariance to discover samples which breach the preservation of equivariance into the latent for anomaly detection.

Furthermore, given the significant interest in image-language models in the research community, this thesis can provide the community with tools to understand the geometry and topology of the shared representation space between images and language. For example, it is natural to assume in data manifold of image and language data varies significantly in its intrinsic dimension. Therefore, tools in this thesis can help to analyse this difference as well as help to build a robust shared representation space which takes into account the different data manifolds.

Finally, this work can be used to develop a set of robustness evaluation metrics for deep learning models used in medical imaging to serve as a measure of reliability of these models to be translated into clinical practise.

Bibliography

[ACM22] Shreya Arya, Justin Curry, and Sayan Mukherjee. A sheaf-theoretic construction of shape space. *arXiv preprint arXiv:2204.09020*, 2022. 166

[AGG83] Sharon Lee Armstrong, Lila R Gleitman, and Henry Gleitman. What some concepts might not be. *Cognition*, 13(3):263–308, 1983. 187

[ALDP20] Nader Aldoj, Steffen Lukas, Marc Dewey, and Tobias Penzkofer. Semi-automatic classification of prostate cancer on multi-parametric mr imaging using a multi-channel 3d convolutional neural network. *European radiology*, 30(2):1243–1253, 2020. 24

[AMJ18] David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, 31, 2018. 188

[ARB+22] Michela Antonelli, Annika Reinke, Spyridon Bakas, Keyvan Farahani, Annette Kopp-Schneider, Bennett A Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M Summers, et al. The medical segmentation decathlon. *Nature communications*, 13(1):4128, 2022. 62, 137

[ASA+22] Mina Ghadimi Atigh, Julian Schoep, Erman Acar, Nanne van Noord, and Pascal Mettes. Hyperbolic image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4453–4462, 2022. 109

[ASJ+21] Fatih Altay, Guillermo Ramón Sánchez, Yanli James, Stephen V Faraone, Senem Velipasalar, and Asif Salekin. Preclinical stage alzheimer's disease detection using magnetic resonance image scans. In *Proceedings of the*

AAAI Conference on Artificial Intelligence, volume 35, pages 15088–15097, 2021. 75

[Awo10] Steve Awodey. *Category theory*. Oxford university press, 2010. 50, 51, 52, 53

[BAAA+18] Mrishta Brizmohun Appayya, Jim Adshead, Hashim U Ahmed, Clare Allen, Alan Bainbridge, Tristan Barrett, Francesco Giganti, John Graham, Phil Haslam, Edward W Johnston, et al. National implementation of multi-parametric magnetic resonance imaging for prostate cancer detection–recommendations from a uk consensus meeting. *BJU international*, 122(1):13–25, 2018. 24

[BBdOBL22] Federico Barbero, Cristian Bodnar, Haitz Sáez de Ocáriz Borde, and Pietro Lio. Sheaf attention networks. In *NeurIPS 2022 Workshop on Symmetry and Geometry in Neural Representations*, 2022. 165

[BCCS23] Xiaoyu Bie, Dexiong Chen, Xiaodong Cun, and Xi SHEN. Learning discrete representation with optimal transport quantized autoencoders, 2023. 134

[BDGC+22] Cristian Bodnar, Francesco Di Giovanni, Benjamin Chamberlain, Pietro Lio, and Michael Bronstein. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in gnns. *Advances in Neural Information Processing Systems*, 35:18527–18541, 2022. 165

[Bek19] Erik J Bekkers. B-spline cnns on lie groups. In *International Conference on Learning Representations*, 2019. 120

[BG18] Gary Becigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. In *International Conference on Learning Representations*, 2018. 77

[BGSF08] Silvia Biasotti, Daniela Giorgi, Michela Spagnuolo, and Bianca Falcidieno. Reeb graphs for shape analysis and applications. *Theoretical computer science*, 392(1-3):5–22, 2008. 165

[BH21] Daniel C Burnston and Philipp Haueis. Evolving concepts of "hierarchy" in systems neuroscience. In *Neural Mechanisms*, pages 113–141. Springer, 2021. 187

[BKH16] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 14

[BLC13] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. 22

[BLV+18] Erik J Bekkers, Maxime W Lafarge, Mitko Veta, Koen AJ Eppenhof, Josien PW Pluim, and Remco Duits. Roto-translation covariant convolutional networks for medical image analysis. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part I*, pages 440–448. Springer, 2018. 126

[BMH+15a] N Bloch, A Madabhushi, H Huisman, J Freymann, J Kirby, M Grauer, A Enquobahrie, C Jaffe, L Clarke, and K Farahani. Cancer imaging archive wiki. *URL http://doi.org/10.7937/K9/TCIA.2015.zF0vlOPv*, 2015. 62, 112, 138

[BMH+15b] Nicholas Bloch, Anant Madabhushi, Henkjan Huisman, John Freymann, Justin Kirby, Michael Grauer, Andinet Enquobahrie, Carl Jaffe, Larry Clarke, and Keyvan Farahani. Nci-isbi 2013 challenge: automated segmentation of prostate structures. *The Cancer Imaging Archive*, 370:6, 2015. 62, 101, 112, 137, 138, 156, 177

[Bod23] Cristian Bodnar. *Topological Deep Learning: Graphs, Complexes, Sheaves*. PhD thesis, University of Cambridge, 2023. xxiii, xxiv, 33, 35, 54

[BPC+20] EJ Bass, A Pantovic, M Connor, R Gabe, AR Padhani, A Rockall, H Sokhi, H Tam, M Winkler, and HU Ahmed. A systematic review and meta-analysis of the diagnostic accuracy of biparametric prostate mri for prostate cancer in men at risk. *Prostate Cancer and Prostatic Diseases*, pages 1–16, 2020. 24

[Bre12] Glen E Bredon. *Sheaf theory*, volume 170. Springer Science & Business Media, 2012. 165, 166, 171

[BTC15] Tristan Barrett, Baris Turkbey, and Peter L Choyke. Pi-rads version 2: what you need to know. *Clinical radiology*, 70(11):1165–1176, 2015. 63

[ÇAL+16] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016. 161

[CBO+20] James R Clough, Nicholas Byrne, Ilkay Oksuz, Veronika A Zimmer, Julia A Schnabel, and Andrew P King. A topological loss function for deep-learning based image segmentation using persistent homology. *IEEE transactions on pattern analysis and machine intelligence*, 44(12):8766–8778, 2020. 150, 177

[CCG+21] Mathieu Carriere, Frédéric Chazal, Marc Glisse, Yuichi Ike, Hariprasad Kannan, and Yuhei Umeda. Optimizing persistent homology based functions. In *International conference on machine learning*, pages 1294–1303. PMLR, 2021. 154, 170

[CCS+21] Renato Cuocolo, Albert Comelli, Alessandro Stefano, Viviana Benfante, Navdeep Dahiya, Arnaldo Stanzione, Anna Castaldo, Davide Raffaele De Lucia, Anthony Yezzi, and Massimo Imbriaco. Deep learning whole-gland and zonal prostate segmentation on a public mri dataset. *Journal of Magnetic Resonance Imaging*, 54(2):452–459, 2021. 63

[CDB+19] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2229–2238, 2019. 85, 90, 92, 116, 159, 181

[CDF+04] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on*

statistical learning in computer vision, ECCV, volume 1, pages 1–2. Prague, 2004. 20

[CFP21] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 357–366, 2021. 78

[CGI+21] Victor M Campello, Polyxeni Gkontra, Cristian Izquierdo, Carlos Martin-Isla, Alireza Sojoudi, Peter M Full, Klaus Maier-Hein, Yao Zhang, Zhiqiang He, Jun Ma, et al. Multi-centre, multi-vendor and multi-disease cardiac segmentation: the m&ms challenge. *IEEE Transactions on Medical Imaging*, 40(12):3543–3554, 2021. 64, 137, 156, 177

[CGW18] Taco S Cohen, Mario Geiger, and Maurice Weiler. Intertwiners between induced representations (with applications to the theory of equivariant neural networks). *arXiv preprint arXiv:1803.10743*, 2018. 126

[Che21] Yurong Chen. Towards to robust and generalized medical image segmentation framework. *arXiv preprint arXiv:2108.03823*, 2021. 96

[CLJ20] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. In *Eighth International Conference on Learning Representations-ICLR 2020*, 2020. 71

[CLY+21] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021. 75, 113

[CNL11] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 66, 199

[CQQ+20] Chen Chen, Chen Qin, Huaqi Qiu, Cheng Ouyang, Shuo Wang, Liang Chen, Giacomo Tarroni, Wenjia Bai, and Daniel Rueckert. Realistic ad-

versarial data augmentation for mr image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 667–677. Springer, 2020. 29, 116, 138, 158, 181

[CSC+21] Renato Cuocolo, Arnaldo Stanzione, Anna Castaldo, Davide Raffaele De Lucia, and Massimo Imbriaco. Quality control and whole-gland, zonal and lesion annotations for the prostatex challenge public dataset. *European Journal of Radiology*, 138:109647, 2021. 63

[CSHB18] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE, 2018. 187

[CTAN+20] Jose M Castillo T, Muhammad Arif, Wiro J Niessen, Ivo G Schoots, Jifke F Veenland, et al. Automated classification of significant prostate cancer on mri: A systematic review on the performance of machine learning applications. *Cancers*, 12(6):1606, 2020. 24

[CTT+19] Benjamin Chidester, That-Vinh Ton, Minh-Triet Tran, Jian Ma, and Minh N Do. Enhanced rotation-equivariant u-net for nuclear segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 126

[Cur14] Justin Michael Curry. *Sheaves, cosheaves and applications*. University of Pennsylvania, 2014. 56, 166

[Cur15] Justin Michael Curry. Topological data analysis and cosheaves. *Japan Journal of Industrial and Applied Mathematics*, 32:333–371, 2015. 166

[CUYH20] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020. 65, 199

[CVS+13] Kenneth Clark, Bruce Vendt, Kirk Smith, John Freymann, Justin Kirby, Paul Koppel, Stephen Moore, Stanley Phillips, David Maffitt, Michael

Pringle, et al. The cancer imaging archive (tcia): maintaining and operating a public information repository. *Journal of digital imaging*, 26:1045–1057, 2013. 62, 112, 138

[CW16a] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016. 120, 121, 122, 126

[CW16b] Taco S Cohen and Max Welling. Steerable cnns. In *International Conference on Learning Representations*, 2016. 122, 126

[CW17] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017. 96

[CWC+23] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. In *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*, pages 205–218. Springer, 2023. 75

[CYRL19] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems*, 32, 2019. 45, 48, 188, 189, 193, 194, 197

[CZS+19] Ruiming Cao, Xinran Zhong, Sepideh Shakeri, Amirhossein Mohammadian Bajgiran, Sohrab Afshari Mirak, Dieter Enzmann, Steven S Raman, and Kyunghyun Sung. Prostate cancer detection and segmentation in multi-parametric mri via cnn and conditional random field. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 1900–1904. IEEE, 2019. 24

[DBK+20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. xxiii, 1, 17, 19, 70, 78, 84, 85

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 17

[DDFK16] Sander Dieleman, Jeffrey De Fauw, and Koray Kavukcuoglu. Exploiting cyclic symmetry in convolutional neural networks. In *International conference on machine learning*, pages 1889–1898. PMLR, 2016. 126

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 127

[Den12] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012. 65, 199

[DSSU21] Nathan Drenkow, Numair Sani, Ilya Shpitser, and Mathias Unberath. A systematic review of robustness in deep learning for computer vision: Mind the gap? *arXiv preprint arXiv:2112.00639*, 2021. xxiii, 2, 25, 26, 27

[DT17] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. arxiv 2017. *arXiv preprint arXiv:1708.04552*, 2017. 28, 116, 138, 158, 181, 183

[dTL+21] Stéphane d'Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *International conference on machine learning*, pages 2286–2296. PMLR, 2021. 71

[DVBK17] Finale Doshi-Velez, Ryan Budish, and Mason Kortz. The role of explanation in algorithmic trust. Technical report, Technical report, Artificial Intelligence and Interpretability Working Group . . . , 2017. 187

[DW18] Paweł Dłotko and Thomas Wanner. Rigorous cubical approximation and persistent homology of continuous functions. *Computers & Mathematics with Applications*, 75(5):1648–1666, 2018. 150

[DZ00] Benoit M Dawant and Alex P Zijdenbos. Image segmentation. *Handbook of medical imaging*, 2:71–127, 2000. 14, 15

[EH22] Herbert Edelsbrunner and John L Harer. *Computational topology: an introduction*. American Mathematical Society, 2022. 41

[EHN+04] LJ Erasmus, D Hurter, M Naudé, HG Kritzinger, and S Acho. A short overview of mri artefacts. *SA Journal of Radiology*, 8(2), 2004. 23

[ELZ00] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. In *Proceedings 41st annual symposium on foundations of computer science*, pages 454–463. IEEE, 2000. 35, 40

[ERO21] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 21

[FDC+19] AM Feldman, Zhenzhen Dai, Eric Carver, Chang Liu, Joon K Lee, Milan Pantelic, Mohamed A Elshaikh, and Ning Wen. Utilizing a deep learning-based object detection and instance segmentation algorithm for the delineation of prostate and prostate cancer segmentation. *International Journal of Radiation Oncology, Biology, Physics*, 105(1):S197–S198, 2019. 24

[FKMN20] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2020. 72

[GAG+17] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International conference on machine learning*, pages 1243–1252. PMLR, 2017. 17

[GAKRM21] Mina Ghadimi Atigh, Martin Keller-Ressel, and Pascal Mettes. Hyperbolic busemann learning with ideal prototypes. *Advances in Neural Information Processing Systems*, 34:103–115, 2021. 109

[GBH18] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in neural information processing systems*, 31, 2018. 48, 188, 189

[GENT+21] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12259–12269, 2021. 72

[GHW+22] Jianyuan Guo, Kai Han, Han Wu, Chang Xu, Yehui Tang, Chunjing Xu, and Yunhe Wang. Cmt: Convolutional neural networks meet vision transformers. 2022 ieee. In *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12165–12175, 2022. 72

[GKL+21] Asma Ghandeharioun, Been Kim, Chun-Liang Li, Brendan Jou, Brian Eoff, and Rosalind Picard. Dissect: Disentangled simultaneous explanations via concept traversals. In *International Conference on Learning Representations*, 2021. 187

[GRM+18] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2018. 73, 127, 130

[GSC00] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000. 16

[GSK+16] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232, 2016. 16

[GTR+18] Robert Geirhos, Carlos RM Temme, Jonas Rauber, Heiko H Schütt, Matthias Bethge, and Felix A Wichmann. Generalisation in humans and

deep neural networks. *Advances in neural information processing systems*, 31, 2018. 28

[GWE+19] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *International Conference on Machine Learning*, pages 2376–2384. PMLR, 2019. 187

[GWZK19] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. *Advances in neural information processing systems*, 32, 2019. 187

[GZM21] Yunhe Gao, Mu Zhou, and Dimitris N Metaxas. Utnet: a hybrid transformer architecture for medical image segmentation. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part III 24*, pages 61–71. Springer, 2021. 75

[Hal13] Paul R Halmos. *Measure theory*, volume 18. Springer, 2013. 60

[Har13] Robin Hartshorne. *Algebraic geometry*, volume 52. Springer Science & Business Media, 2013. 165

[HB17] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017. 127

[HC21] Chuan-Shen Hu and Yu-Min Chung. A sheaf and topology approach to detecting local merging relations in digital images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4396–4405, 2021. 166

[HD18] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2018. 25

[HG16] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 13

[HGW+21] Joy Hsu, Jeffrey Gu, Gong Wu, Wah Chiu, and Serena Yeung. Capturing implicit hierarchical structure in 3d biomedical images with self-supervised hyperbolic representations. *Advances in Neural Information Processing Systems*, 34:5112–5123, 2021. 109

[HKNU17] Christoph Hofer, Roland Kwitt, Marc Niethammer, and Andreas Uhl. Deep learning with topological signatures. *Advances in neural information processing systems*, 30, 2017. 150

[HLSC19] Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. Topology-preserving deep image segmentation. *Advances in neural information processing systems*, 32, 2019. 150, 156, 178, 183

[HM19] Drew Hudson and Christopher D Manning. Learning by abstraction: The neural state machine. *Advances in Neural Information Processing Systems*, 32, 2019. 188

[HNT+22] Ali Hatamizadeh, Vishwesh Nath, Yucheng Tang, Dong Yang, Holger R Roth, and Daguang Xu. Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images. In *International MICCAI Brainlesion Workshop*, pages 272–284. Springer, 2022. 75, 156

[HPCW18] Emiel Hoogeboom, Jorn WT Peters, Taco S Cohen, and Max Welling. Hexaconv. *arXiv preprint arXiv:1803.02108*, 2018. 126

[HTB+22] Xinzi He, Ee-Leng Tan, Hanwen Bi, Xuzhe Zhang, Shijie Zhao, and Baiying Lei. Fully transformer network for skin lesion analysis. *Medical Image Analysis*, 77:102357, 2022. 75

[HTN+22] Ali Hatamizadeh, Yucheng Tang, Vishwesh Nath, Dong Yang, Andriy Myronenko, Bennett Landman, Holger R Roth, and Daguang Xu. Unetr: Transformers for 3d medical image segmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 574–584, 2022. 75

[HTP+18] Truong Son Hy, Shubhendu Trivedi, Horace Pan, Brandon M Anderson, and Risi Kondor. Predicting molecular properties with covariant compositional networks. *The Journal of chemical physics*, 148(24), 2018. 126

[Hu20] Chuan-Shen Hu. A brief note for sheaf structures on posets. *arXiv preprint arXiv:2010.09651*, 2020. 55, 165, 166

[HWG+19] Koen Helwegen, James Widdicombe, Lukas Geiger, Zechun Liu, Kwang-Ting Cheng, and Roeland Nusselder. Latent weights do not exist: Rethinking binarized neural network optimization. *Advances in neural information processing systems*, 32, 2019. 192, 194, 195, 197, 198

[HXW+21] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *Advances in neural information processing systems*, 34:15908–15919, 2021. 71

[HZG+22] Lei Hu, Da-Wei Zhou, Xiang-Yu Guo, Wen-Hao Xu, Li-Ming Wei, and Jun-Gong Zhao. Adversarial training for prostate cancer classification using magnetic resonance imaging. *Quantitative Imaging in Medicine and Surgery*, 12(6):3276–3287, 2022. 85, 90, 92

[HZRS15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 87

[HZRS16a] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 14

[HZRS16b] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 100, 113, 116, 158, 198

[IJK+21] Fabian Isensee, Paul F Jaeger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein. nnu-net: a self-configuring method for deep learning-based

biomedical image segmentation. *Nature methods*, 18(2):203–211, 2021. 156, 183

[IKE+20] Md Amirul Islam, Matthew Kowal, Patrick Esser, Sen Jia, Björn Ommer, Konstantinos G Derpanis, and Neil Bruce. Shape or texture: Understanding discriminative features in cnns. In *International Conference on Learning Representations*, 2020. 130

[IMK+14] Forrest Iandola, Matt Moskewicz, Sergey Karayev, Ross Girshick, Trevor Darrell, and Kurt Keutzer. Densenet: Implementing efficient convnet descriptor pyramids. *arXiv preprint arXiv:1404.1869*, 2014. 198

[IMU+18] Junichiro Ishioka, Yoh Matsuoka, Sho Uehara, Yosuke Yasuda, Toshiki Kijima, Soichiro Yoshida, Minato Yokoyama, Kazutaka Saito, Kazunori Kihara, Noboru Numao, et al. Computer-aided diagnosis of prostate cancer on magnetic resonance imaging using a convolutional neural network algorithm. *BJU international*, 122(3):411–417, 2018. 24

[IST+19] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019. 130

[JGP16] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2016. 110, 111

[JPB+19] AEW Johnson, TJ Pollard, SJ Berkowitz, R Mark, and S Horng. Mimic-cxr database (version 2.0. 0). physionet, 2019. 66, 199

[JPG+19] Alistair EW Johnson, Tom J Pollard, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Yifan Peng, Zhiyong Lu, Roger G Mark, Seth J Berkowitz, and Steven Horng. Mimic-cxr-jpg, a large publicly available database of labeled chest radiographs. *arXiv preprint arXiv:1901.07042*, 2019. 66, 199

[JSH+19] Haozhe Jia, Yang Song, Heng Huang, Weidong Cai, and Yong Xia. Hd-net: hybrid discriminative network for prostate segmentation in mr images. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part II 22*, pages 110–118. Springer, 2019. 24

[KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 19, 87, 101, 112, 138, 156, 178

[KB20] Myeongjin Kim and Hyeran Byun. Learning texture invariant representation for domain adaptation of semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12975–12984, 2020. 127

[KBS+17] Simon Kohl, David Bonekamp, Heinz-Peter Schlemmer, Kaneschka Yaqubi, Markus Hohenfellner, Boris Hadaschik, Jan-Philipp Radtke, and Klaus Maier-Hein. Adversarial networks for the detection of aggressive prostate cancer. *arXiv preprint arXiv:1702.08014*, 2017. 24

[KCS20] Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *International Conference on Machine Learning*, pages 5275–5285. PMLR, 2020. 28

[KHLY20] Adam Kortylewski, Ju He, Qing Liu, and Alan L Yuille. Compositional convolutional neural networks: A deep architecture with innate robustness to partial occlusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8940–8949, 2020. 108

[KJJP21] Henry Kvinge, Brett Jefferson, Cliff Joslyn, and Emilie Purvine. Sheaves as a framework for understanding and interpreting model fit. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4222–4230, 2021. 166

[KMN+16] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep

learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2016. 72

[KMU+20] Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. Hyperbolic image embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6418–6428, 2020. 109

[KNSK21] Avinash Kori, Parth Natekar, Balaji Srinivasan, and Ganapathy Krishnamurthi. Interpreting deep neural networks for medical imaging using concept graphs. In *International Workshop on Health Intelligence*, pages 201–216. Springer, 2021. 187

[KPO+21] Gwanghyun Kim, Sangjoon Park, Yujin Oh, Joon Beom Seo, Sang Min Lee, Jin Hwan Kim, Sungjun Moon, Jae-Kwang Lim, and Jong Chul Ye. Severity quantification and lesion localization of covid-19 on cxr using vision transformer. *arXiv preprint arXiv:2103.07062*, 2021. 75

[Kro15] Joshua Alexander Kroll. *Accountable algorithms*. PhD thesis, Princeton University, 2015. 187

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 14

[KSP+18] Risi Kondor, Hy Truong Son, Horace Pan, Brandon Anderson, and Shubhendu Trivedi. Covariant compositional networks for learning graphs. *arXiv preprint arXiv:1801.02144*, 2018. 126

[KWG+18] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018. 187

[LB+95] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995. 11, 121

[LBBH98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 120

[LCP+20] Shaoteng Liu, Jingjing Chen, Liangming Pan, Chong-Wah Ngo, Tat-Seng Chua, and Yu-Gang Jiang. Hyperbolic visual embedding learning for zero-shot recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9273–9281, 2020. 109

[LDB+14] Geert Litjens, Oscar Debats, Jelle Barentsz, Nico Karssemeijer, and Henkjan Huisman. Computer-aided detection of prostate cancer in mri. *IEEE transactions on medical imaging*, 33(5):1083–1092, 2014. 62, 81

[LDB+17] G Litjens, O Debats, J Barentsz, N Karssemeijer, and H Huisman. Cancer imaging archive wiki. *URL https://doi. org/10.7937/K9TCIA*, 2017. 63, 183

[Lip18] Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018. 187

[LKB+17] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017. 63, 183

[LL17] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 187, 199, 201, 202

[LLC+21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 72

[LNK19] Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. *Advances in Neural Information Processing Systems*, 32, 2019. 188

[LRD+22] Andrew K Lampinen, Nicholas Roy, Ishita Dasgupta, Stephanie CY Chan, Allison Tam, James Mcclelland, Chen Yan, Adam Santoro, Neil C Rabinowitz, Jane Wang, et al. Tell me why! explanations support learning relational and causal structure. In *International Conference on Machine Learning*, pages 11868–11890. PMLR, 2022. 187

[LSGL20] Qiufu Li, Linlin Shen, Sheng Guo, and Zhihui Lai. Wavelet integrated cnns for noise-robust image classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7245–7254, 2020. 29

[LTS+22] Xiao Liu, Spyridon Thermos, Pedro Sanchez, Alison Q O'Neil, and Sotirios A Tsaftaris. vmfnet: Compositionality meets domain-generalised segmentation. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part VII*, pages 704–714. Springer, 2022. 108, 181

[LW20] Leon Lang and Maurice Weiler. A wigner-eckart theorem for group equivariant convolution kernels. In *International Conference on Learning Representations*, 2020. 122, 123, 124, 125, 126

[LWL+20] Lida Li, Kun Wang, Shuai Li, Xiangchu Feng, and Lei Zhang. Lst-net: Learning a convolutional neural network with a learnable sparse transform. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 562–579. Springer, 2020. 29

[LWU+20] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020. 2

[LWV+18] Jasper Linmans, Jim Winkens, Bastiaan S Veeling, Taco S Cohen, and Max Welling. Sample efficient semantic segmentation using rotation equivariant convolutional networks. *arXiv preprint arXiv:1807.00583*, 2018. 126

[LWW+20] Tao Lei, Risheng Wang, Yong Wan, Bingtao Zhang, Hongying Meng, and Asoke K Nandi. Medical image segmentation using deep learning: a survey. *arXiv preprint arXiv:2009.13120*, 2020. 96

[LXI+15] Bennett Landman, Zhoubing Xu, J Igelsias, Martin Styner, T Langerak, and Arno Klein. Miccai multi-atlas labeling beyond the cranial vault–workshop and challenge. In *Proc. MICCAI Multi-Atlas Labeling Beyond Cranial Vault—Workshop Challenge*, volume 5, page 12, 2015. 61, 101, 112, 156

[LXT+18] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018. 72

[MDFFF17] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017. 96

[MDS+18] Stephen Muggleton, Wang-Zhou Dai, Claude Sammut, Alireza Tamaddoni-Nezhad, Jing Wen, and Zhi-Hua Zhou. Meta-interpretive learning from noisy images. *Machine Learning*, 107(7):1097–1118, 2018. 188

[MGK+18] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *International Conference on Learning Representations*, 2018. 188

[MGTC21] Philip Müller, Vladimir Golkov, Valentina Tomassini, and Daniel Cremers. Rotation-equivariant deep learning for diffusion mri. *arXiv preprint arXiv:2102.06942*, 2021. 126

[MHRB20] Michael Moor, Max Horn, Bastian Rieck, and Karsten Borgwardt. Topological autoencoders. In *International conference on machine learning*, pages 7045–7054. PMLR, 2020. 150, 154, 170

[MJD+21] Chengzhi Mao, Lu Jiang, Mostafa Dehghani, Carl Vondrick, Rahul Sukthankar, and Irfan Essa. Discrete representations strengthen vision transformer robustness. In *International Conference on Learning Representations*, 2021. 74

[MLF+09] David Meunier, Renaud Lambiotte, Alex Fornito, Karen Ersche, and Edward T Bullmore. Hierarchical modularity in human brain functional networks. *Frontiers in neuroinformatics*, 3:37, 2009. 187

[MMS+18] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 28

[MNA16] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. Ieee, 2016. 24, 113

[MQC+22] Xiaofeng Mao, Gege Qi, Yuefeng Chen, Xiaodan Li, Ranjie Duan, Shaokai Ye, Yuan He, and Hui Xue. Towards robust vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12042–12051, 2022. 72

[MVKT17] Diego Marcos, Michele Volpi, Nikos Komodakis, and Devis Tuia. Rotation equivariant vector field networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5048–5057, 2017. 126

[MZJ+23] Chengzhi Mao, Lingyu Zhang, Abhishek Vaibhav Joshi, Junfeng Yang, Hao Wang, and Carl Vondrick. Robust perception through equivariance. In *International Conference on Machine Learning*, pages 23852–23870. PMLR, 2023. 131

[NH10] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010. 13

[NJT14] Maximilian Nickel, Xueyan Jiang, and Volker Tresp. Reducing the rank in relational factorization models by including observable patterns. *Advances in Neural Information Processing Systems*, 27, 2014. 188

[NK17] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017. 109, 193

[NQ22] Manh-Hung Nguyen and Khai Ngo Quang. A study of vision transformer for lung diseases classification. In *2022 6th International Conference on Green Technology and Sustainable Development (GTSD)*, pages 116–121. IEEE, 2022. 75

[NRK+21] Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34:23296–23308, 2021. 72, 73, 75

[PC22] Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2071–2081, 2022. 72

[PDO+22] Shuchao Pang, Anan Du, Mehmet A Orgun, Yan Wang, Quan Z Sheng, Shoujin Wang, Xiaoshui Huang, and Zhenmei Yu. Beyond cnns: exploiting further inherent symmetries in medical image segmentation. *IEEE transactions on cybernetics*, 2022. 126

[Pea09] Judea Pearl. *Causality*. Cambridge university press, 2009. 25

[PGSO21] Fernando Pérez-García, Rachel Sparks, and Sébastien Ourselin. Torchio: a python library for efficient loading, preprocessing, augmentation and patch-based sampling of medical images in deep learning. *Computer Methods and Programs in Biomedicine*, 208:106236, 2021. 86, 114

[PJS17] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017. 25

[PK21] Namuk Park and Songkuk Kim. How do vision transformers work? In *International Conference on Learning Representations*, 2021. xxiv, 72, 73, 74, 157

[PK22] Namuk Park and Songkuk Kim. Blurs behave like ensembles: Spatial smoothings to improve accuracy, uncertainty, and robustness. In *International Conference on Machine Learning*, pages 17390–17419. PMLR, 2022. 73

[QZP20] Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12556–12565, 2020. 29

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 15, 113, 156, 161, 209

[RHO+19] Jérôme Rony, Luiz G Hafemann, Luiz S Oliveira, Ismail Ben Ayed, Robert Sabourin, and Eric Granger. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4322–4330, 2019. 85

[RHW86] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986. 16, 19

[RM51] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951. 19

[RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of*

the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pages 1135–1144, 2016. 187, 199, 201, 202

[RSP17] Siamak Ravanbakhsh, Jeff Schneider, and Barnabas Poczos. Equivariance through parameter-sharing. In *International conference on machine learning*, pages 2892–2901. PMLR, 2017. 126

[RSZ+20] Evgenia Rusak, Lukas Schott, Roland S Zimmermann, Julian Bitterwolf, Oliver Bringmann, Matthias Bethge, and Wieland Brendel. A simple way to make neural networks robust against diverse image corruptions. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 53–69. Springer, 2020. 28

[RTB+18] Bastian Rieck, Matteo Togninalli, Christian Bock, Michael Moor, Max Horn, Thomas Gumbsch, and Karsten Borgwardt. Neural persistence: A complexity measure for deep neural networks using algebraic topology. In *International Conference on Learning Representations*, 2018. 150

[RYB+20] Bastian Rieck, Tristan Yates, Christian Bock, Karsten Borgwardt, Guy Wolf, Nicholas Turk-Browne, and Smita Krishnaswamy. Uncovering the topology of time-varying fmri data using cubical persistence. *Advances in neural information processing systems*, 33:6900–6912, 2020. 150

[RZL17] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017. 13, 100

[SCD+17] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 187, 199, 201, 202

[SDNT18] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*, 2018. 188

[SDSGR18] Frederic Sala, Chris De Sa, Albert Gu, and Christopher Ré. Representation tradeoffs for hyperbolic embeddings. In *International conference on machine learning*, pages 4460–4469. PMLR, 2018. 188

[SEUD10] Mathieu Salzmann, Carl Henrik Ek, Raquel Urtasun, and Trevor Darrell. Factorized orthogonal latent spaces. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 701–708. JMLR Workshop and Conference Proceedings, 2010. 152

[SGK17] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR, 2017. 187, 199, 201, 202

[SKI+00] Junji Shiraishi, Shigehiko Katsuragawa, Junpei Ikezoe, Tsuneo Matsumoto, Takeshi Kobayashi, Ken-ichi Komatsu, Mitate Matsui, Hiroshi Fujita, Yoshie Kodera, and Kunio Doi. Development of a digital image database for chest radiographs with and without a lung nodule: receiver operating characteristic analysis of radiologists' detection of pulmonary nodules. *American Journal of Roentgenology*, 174(1):71–74, 2000. 64, 101

[SLJ+15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 14

[SSK21] Wolfgang Stammer, Patrick Schramowski, and Kristian Kersting. Right for the right concept: Revising neuro-symbolic concepts by interacting with their explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3619–3629, 2021. 187

[SSP+21] Sam Sattarzadeh, Mahesh Sudhakar, Konstantinos N Plataniotis, Jongseong Jang, Yeonjeong Jeong, and Hyunwoo Kim. Integrated grad-cam: Sensitivity-aware visual explanation of deep convolutional networks via integrated gradient-based scoring. In *ICASSP 2021-2021 IEEE*

International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1775–1779. IEEE, 2021. 187

[SSY+22] Rulin Shao, Zhouxing Shi, Jinfeng Yi, Pin-Yu Chen, and Cho-jui Hsieh. On the adversarial robustness of vision transformers. In *Annual Conference on Neural Information Processing Systems*, 2022. 72, 73

[ST20] Primoz Skraba and Katharine Turner. Wasserstein stability for persistence diagrams. *arXiv preprint arXiv:2006.16824*, 2020. 154, 170

[SUV18] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics, 2018. 17

[SWS17] Dinggang Shen, Guorong Wu, and Heung-Il Suk. Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19:221–248, 2017. 23

[SZ03] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Computer Vision, IEEE International Conference on*, volume 3, pages 1470–1470. IEEE Computer Society, 2003. 20

[SZS+13] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 25

[SZY+18] Yang Song, Yu-Dong Zhang, Xu Yan, Hui Liu, Minxiong Zhou, Bingwen Hu, and Guang Yang. Computer-aided diagnosis of prostate cancer using a deep convolutional neural network from multiparametric mri. *Journal of Magnetic Resonance Imaging*, 48(6):1570–1577, 2018. 24

[TCD+21] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image trans-

formers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 71

[TT20] James Tee and Desmond P Taylor. Is information in the brain represented in continuous or discrete form? *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, 6(3):199–209, 2020. 2, 76

[TTXS19] You-Bao Tang, Yu-Xing Tang, Jing Xiao, and Ronald M Summers. Xlsor: A robust and accurate lung segmentor on chest x-rays using criss-cross attention and customized radiorealistic abnormalities generation. In *International Conference on Medical Imaging with Deep Learning*, pages 457–467. PMLR, 2019. 64, 101

[TWH19] Pavel Tokmakov, Yu-Xiong Wang, and Martial Hebert. Learning compositional representations for few-shot recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6372–6381, 2019. 2

[TWR+16] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR, 2016. 188

[TYL+22] Yucheng Tang, Dong Yang, Wenqi Li, Holger R Roth, Bennett Landman, Daguang Xu, Vishwesh Nath, and Ali Hatamizadeh. Self-supervised pre-training of swin transformers for 3d medical image analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20730–20740, 2022. 72

[UVL16] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 14, 136

[VDL+19] Ramakrishna Vedantam, Karan Desai, Stefan Lee, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Probabilistic neural symbolic models for interpretable visual question answering. In *International Conference on Machine Learning*, pages 6428–6437. PMLR, 2019. 188

[VDOV+17] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 20, 76, 97, 98, 108, 110, 151, 191, 193

[VN14] Petra Vetter and Albert Newen. Varieties of cognitive penetration in visual perception. *Consciousness and cognition*, 27:62–75, 2014. 187

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1, 15

[WC19] Maurice Weiler and Gabriele Cesa. General e (2)-equivariant steerable cnns. *Advances in neural information processing systems*, 32, 2019. 124, 125, 126, 136

[WC22] Marysia Winkels and Taco S Cohen. 3d g-cnns for pulmonary nodule detection. In *Medical Imaging with Deep Learning*, 2022. xxv, 126, 130

[WGTB17] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5028–5037, 2017. 125, 126

[WH18] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 100, 158

[WHS18] Maurice Weiler, Fred A Hamprecht, and Martin Storath. Learning steerable filters for rotation equivariant cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 849–858, 2018. 126, 136

[WLC+18] Zhiwei Wang, Chaoyue Liu, Danpeng Cheng, Liang Wang, Xin Yang, and Kwang-Ting Cheng. Automated detection of clinically significant prostate cancer in mp-mri images based on an end-to-end deep neural network. *IEEE transactions on medical imaging*, 37(5):1127–1139, 2018. 24

[WPL+17] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017. 101

[WVT+01] CM Wessinger, J VanMeter, Biao Tian, J Van Lare, James Pekar, and Josef P Rauschecker. Hierarchical organization of the human auditory cortex revealed by functional magnetic resonance imaging. *Journal of cognitive neuroscience*, 13(1):1–7, 2001. 187

[WW16] Bao Wang and Guo-Wei Wei. Object-oriented persistent homology. *Journal of computational physics*, 305:276–299, 2016. 150

[WXC+21] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22–31, 2021. 71, 72

[WXL+21] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021. 72

[XLY+20] Zhenlin Xu, Deyi Liu, Junlin Yang, Colin Raffel, and Marc Niethammer. Robust and generalizable visual representation learning via random convolutions. In *International Conference on Learning Representations*, 2020. 28, 116, 127, 128, 129, 138, 158, 181

[YCLX23] Jinyang Yuan, Tonglin Chen, Bin Li, and Xiangyang Xue. Compositional scene representation learning via reconstruction: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2

[YLW+17] Xin Yang, Chaoyue Liu, Zhiwei Wang, Jun Yang, Hung Le Min, Liang Wang, and Kwang-Ting (Tim) Cheng. Co-trained convolutional neural

networks for automated detection of prostate cancer in multi-parametric mri. *Medical Image Analysis*, 42:212–227, 2017. 24

[YWG+18] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. *Advances in neural information processing systems*, 31, 2018. 188

[YYC17] Fan Yang, Zhilin Yang, and William W Cohen. Differentiable learning of logical rules for knowledge base reasoning. *Advances in neural information processing systems*, 30, 2017. 188

[ZCDLP17] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 28

[ZCY+19] Yuyu Zhang, Xinshi Chen, Yuan Yang, Arun Ramamurthy, Bo Li, Yuan Qi, and Le Song. Efficient probabilistic logic reasoning with graph neural networks. In *International Conference on Learning Representations*, 2019. 188

[ZDY19] Qikui Zhu, Bo Du, and Pingkun Yan. Boundary-weighted domain adaptive neural network for prostate mr image segmentation. *IEEE transactions on medical imaging*, 39(3):753–763, 2019. 24

[Zha19] Richard Zhang. Making convolutional networks shift-invariant again. In *International conference on machine learning*, pages 7324–7334. PMLR, 2019. 29

[ZJRS16] Hongyi Zhang, Sashank J Reddi, and Suvrit Sra. Riemannian svrg: Fast stochastic optimization on riemannian manifolds. *Advances in Neural Information Processing Systems*, 29, 2016. 196

[ZRLS22] Huanqiu Zhang, P Dylan Rich, Albert K Lee, and Tatyana O Sharpee. Hippocampal spatial representations exhibit a hyperbolic geometry that expands with experience. *Nature Neuroscience*, pages 1–9, 2022. 108

[ZWY+20] Ling Zhang, Xiaosong Wang, Dong Yang, Thomas Sanford, Stephanie Harmon, Baris Turkbey, Bradford J. Wood, Holger Roth, Andriy Myronenko, Daguang Xu, and Ziyue Xu. Generalizing deep learning for medical image segmentation to unseen domains via deep stacked transformation. *IEEE transactions on medical imaging*, 39(7):2531–2540, 2020. xix, 28, 85, 90, 92, 116, 117, 138, 159, 181