Imperial College London
Department of Computing

Deep Learning for Cortical Surface Reconstruction

Qiang Ma

Supervised by Prof. Daniel Rueckert and Prof. Bernhard Kainz

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computing of Imperial College London

May 2025

Declaration

Originality

I, Qiang Ma, hereby declare that the work presented in this thesis is my own, unless specifically acknowledged.

Copyright

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC). Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes. Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Abstract

Cortical surface reconstruction facilitates both visualisation and quantification of the cerebral cortex, playing a pivotal role in the diagnosis of neurological disorders and the characterisation of cortical folding patterns. Due to the highly folded anatomical structure and intrinsic topological constraints, it is challenging to extract anatomically and topologically accurate cortical surfaces from brain magnetic resonance imaging (MRI) data. In this thesis, we leverage advanced geometric deep learning (DL) techniques and present a series of DL-based frameworks for fast and explicit cortical surface reconstruction from adult, fetal and neonatal brain MRI scans.

Firstly, we propose a novel deep neural network architecture for explicit and topology-preserving cortical surface reconstruction end-to-end from adult brain MRI. To prevent surface self-intersections, we further introduce neural ordinary differential equations to learn diffeomorphic surface deformations for cortical surface reconstruction. Secondly, we develop customised DL-based frameworks for fetal and neonatal subjects that undergo rapid brain development. To tackle considerable brain variations across different ages, we utilise an attention mechanism to learn neonatal cortical surface construction conditioned on the ages of neonates. Furthermore, we devise a weakly supervised framework for fetal cortical surface reconstruction supervised by brain segmentations, thereby eliminating the reliance on pseudo ground truth cortical surfaces generated by traditional neuroimage processing pipelines. Finally, we present a fast and robust DL-based pipeline for cortical surface-based structural MRI processing of developing human brains.

We demonstrate that the DL-based cortical surface reconstruction approaches proposed in this thesis achieve superior geometrical and topological accuracy, fast inference within only a few seconds, and high adaptability to brain MRI acquired from subjects in various age groups including adults, fetuses and neonates.

Acknowledgements

The journey of my PhD started during the COVID-19 pandemic. It was a difficult period for everyone, and thus I am very grateful for all the assistance I have received. First of all, I would like to express my sincere gratitude to my supervisors Prof. Daniel Rueckert and Prof. Bernhard Kainz, for their invaluable support and constant encouragement.

Daniel led me into my PhD journey and always guided me in the right direction. He unstintingly makes time for our meetings, provides his resources, and shares his knowledge to address any problems I have encountered. His unlimited help and constructive suggestions over the past four years are immeasurable treasures for my research career.

Bernhard always responds promptly and patiently whenever I need help. I have been inspired a lot by his numerous creative ideas and insightful views on research. I am thankful for not only his careful edits and polishing of every line I have written in my papers, but also his affirmation of every progress I have made.

Many thanks to all other academic staff at BioMedIA group. Prof. Ben Glocker generously gave me valuable feedback and suggestions in my early and late stage PhD assessments. Dr. Wenjia Bai kindly accepted my application for a postdoctoral position and greatly helped me to prepare for the start of my new research journey. In addition, I would like to thank my colleagues at BioMedIA (in alphabetical order): Amir Alansary, Benjamin Hou, Chen Chen, Chen Qin, Cheng Ouyang, Hadrien Reynaud, Huaqi Qiu, Jeremy Tan, Liu Li, Matthew Baugh, Mengyun Qiao, Michael Tanzer, Qingjie Meng, Weitong Zhang, Zeju Li, and all others. Special thanks go to Dr. Amir Alansary, who acted as my third supervisor at the early stage of my PhD, offering considerable help as well as technical and practical advice that benefits me a lot. I would also like to thank Dr. Emma C. Robinson and her team members at King's College London for their close collaboration.

I deeply appreciate the financial support I received from the President's PhD Scholarship awarded by Imperial College London. I am also thankful to the staff of the Department of Computing, including but not limited to Amani El-Kholy, Bridget Gundry, Geoff Bruce, Hassan Patel, Katherine Bellenie, and Lloyd Kamara, for their prompt help, sustained support, and generous funding.

Finally, I am forever grateful to my family for their unreserved trust and love. My mother, Nianhong Xue, and my father, Jian Ma, provided me with mental, emotional and financial support selflessly at every stage of my life. They always care about me, stand by me, and firmly support my every decision. My grandfather, Keli Xue (†2021), always encouraged and inspired me to get through the most difficult times in my life. My wife, Jiayin Li, accompanied me to a new country bravely and unwaveringly during the pandemic. She took excellent care of my daily life, raised me up when I felt down, and brought me happiness every day throughout my PhD journey.

This thesis is dedicated to my family with my deepest gratitude for everything they have done. Their love and encouragement have always been my greatest motivation.

To my family.

Contents

Abstract **ii**

Acknowledgements **iii**

1 Introduction **1**

1.1 Motivation and Objectives 1

1.2 Contributions and Thesis Overview 7

1.3 Publications 10

2 Background **12**

2.1 Mathematical Preliminaries 13

2.1.1 Topology and Manifold 13

2.1.2 Ordinary Differential Equations 15

2.2 Geometric Deep Learning 20

2.2.1 Deep Learning Foundations 20

2.2.2 Geometric Domains 22

2.2.3 Geometric Deep Neural Network Models 24

2.2.4 Deep Learning for Surface Reconstruction 30

2.3 Cortical Surface Reconstruction 34

2.3.1 Problem Formulation 34

2.3.2 Methodology 35

2.3.3 Neuroimage Processing Pipeline 40

3 Learning Explicit and Diffeomorphic Cortical Surface Reconstruction 47

3.1 Learning Explicit Cortical Pial Surface Reconstruction 49

3.1.1 Introduction 49

3.1.2 Related Work 51

3.1.3 Method 52

3.1.3.1 Deformation Block 52

3.1.3.2 Smoothing and Training 54

3.1.4 Experiments 55

3.1.5 Discussion 59

3.2 Learning Diffeomorphic Cortical Surface Reconstruction by Neural ODEs . 60

3.2.1 Introduction 60

3.2.2 CortexODE Framework 65

3.2.2.1 Neural ODE-based Framework 66

3.2.2.2 Deformation Network 67

3.2.2.3 Existence and Uniqueness of Solutions 69

3.2.2.4 CortexODE Discretisation 70

3.2.3 CortexODE-based Processing Pipeline 72

3.2.3.1 White Matter Segmentation 72
3.2.3.2 Signed Distance Transformation 73
3.2.3.3 Topology Correction 74
3.2.3.4 Cortical Surface Reconstruction 75
3.2.4 Experiments 76
3.2.4.1 Experiment Details 76
3.2.4.2 Comparative Results 78
3.2.4.3 ODE Integration 83
3.2.4.4 Clinical Analysis 85
3.2.4.5 Ablation Study 88
3.2.5 Discussion 92
3.2.6 Proof of Theorems 93

4 Learning Cortical Surface Reconstruction for Developing Human Brain 96

4.1 Conditional Temporal Attention Networks for Neonatal Cortical Surface Reconstruction 98
4.1.1 Introduction 98
4.1.2 Method 101
4.1.2.1 Diffeomorphic Surface Deformation 101
4.1.2.2 Conditional Temporal Attention Network 101
4.1.2.3 Neonatal Cortical Surface Reconstruction 103

4.1.3 Experiments . . . 105
4.1.3.1 Implementation Details . . . 105
4.1.3.2 Comparative Results . . . 106
4.1.3.3 Ablation Study . . . 109
4.1.3.4 Attention Maps . . . 110
4.1.4 Discussion . . . 111
4.2 Weakly Supervised Fetal Cortical Surface Reconstruction Guided by Segmentations . . . 113
4.2.1 Introduction . . . 113
4.2.2 Method . . . 115
4.2.2.1 Temporal Attention Networks . . . 115
4.2.2.2 Initial Surface Generation . . . 116
4.2.2.3 White Matter Surface Reconstruction . . . 117
4.2.2.4 Pial Surface Reconstruction . . . 117
4.2.3 Experiments . . . 119
4.2.3.1 Experiments on Adult Brain MRI . . . 119
4.2.3.2 Experiments on Fetal Brain MRI . . . 125
4.2.4 Discussion . . . 126

5 A Deep Learning-based Pipeline for Neonatal Cortical Surface Reconstruction 128

5.1 Introduction . . . 130

5.2 Deep Learning-based dHCP Neonatal Pipeline . . . 135
5.2.1 Data Acquisition . . . 135
5.2.2 Pipeline Overview . . . 136
5.3 Method . . . 137
5.3.1 Dataset Preparation . . . 137
5.3.2 Preprocessing . . . 139
5.3.3 Cortical Surface Reconstruction . . . 140
5.3.3.1 Diffeomorphic Surface Deformation . . . 140
5.3.3.2 Multiscale Deformation Network . . . 141
5.3.3.3 Model Training . . . 143
5.3.4 Cortical Surface Inflation . . . 145
5.3.5 Cortical Feature Estimation . . . 146
5.3.6 Spherical Projection . . . 147
5.4 Comparison to Existing Neuroimage Pipelines . . . 150
5.4.1 Comparison to Original dHCP Pipeline . . . 150
5.4.2 Comparison to DL-based Neuroimage Pipelines . . . 152
5.5 Quality Control . . . 153
5.5.1 Cortical Surface Quality . . . 153
5.5.1.1 Visual Inspection . . . 153
5.5.1.2 Brain Development . . . 154

5.5.1.3 Surface Topology 156

5.5.1.4 Generalisability 158

5.5.2 Sphere Quality 159

5.6 Discussion 162

6 Conclusion **165**

6.1 Summary of Thesis Achievements 165

6.2 Future Work 169

Bibliography **171**

List of Tables

2.1 Topological requirements of cortical surfaces and corresponding meshes . . 35

3.1 Comparison of geometric errors for pial surface reconstruction 56

3.2 Comparative results of cortical surface reconstruction on geometric accuracy and self-intersections 79

3.3 The metrics of ODE solvers for training 84

3.4 Comparative results of test-retest reliability 86

3.5 The results of cortical feature estimation 87

3.6 Ablation study for WM segmentation in CortexODE-based pipeline 88

3.7 Comparison on different implementations of the topology correction algorithm 90

4.1 Comparative results of neonatal cortical surface reconstruction on the dHCP dataset 106

4.2 Comparative results of runtime, GPU memory cost, and the number of model parameters for both training and testing 108

4.3 The results of ablation experiments for CoTAN on white matter surface reconstruction 108

4.4 Comparison with existing learning-based approaches on surface type, supervision, and loss function 120

4.5 Comparison with implicit approaches on the HCP dataset at 1mm/2mm resolutions 122

4.6 Comparison with explicit learning-based approaches and ablation studies of CoSeg on the HCP dataset 123

5.1 Comparisons of different approaches between original dHCP pipeline and our DL-based dHCP pipeline 150

5.2 The implementation and runtime of the proposed dHCP DL-based pipeline 151

5.3 Comparisons of functionality and runtime across DL-based neuroimage processing pipelines 152

5.4 The number of self-intersecting faces on cortical surfaces 158

5.5 Comparisons of metric distortions between spherical projection approaches 160

List of Figures

1.1 Illustration of cortical surface reconstruction 2

1.2 Visualisation of cortical surfaces from different perspectives 3

1.3 Visualisation of neonatal brain MRI and cortical surfaces for subjects scanned at 29 to 44 weeks 4

2.1 Surfaces with different topology and genus 14

2.2 Non-Euclidean geometric domains for 3D shape representation 22

2.3 Categories of non-manifold geometries 24

2.4 The architecture of the PointNet 25

2.5 The architecture of the Pixel2Mesh and Voxel2Mesh frameworks 27

2.6 The architecture of the Neural Mesh Flow 28

2.7 The architecture of the DeepSDF framework 29

2.8 The architecture of the NeRF framework 30

2.9 Illustration of implicit and explicit deep learning-based surface reconstruction approaches 31

2.10 Comparison between adult and neonatal T1w and T2w brain MRI 34

2.11 Visualisation of common topological defects 35

2.12 Illustration of cortical surface reconstruction procedure 36

2.13 The workflow of a conventional automated neuroimage processing pipeline. 40

3.1 The architecture of PialNN for pial surface reconstruction 53

3.2 Visualisation of pial surface meshes reconstructed by different approaches . 57

3.3 Qualitative comparison between predicted pial surfaces 58

3.4 Error maps of pial surfaces in ablation study 59

3.5 Runtime of learning-based approaches for pial surface reconstruction 59

3.6 A diffeomorphic flow modelled by CortexODE 63

3.7 The architecture of the deformation network in the CortexODE framework 68

3.8 The CortexODE-based cortical surface reconstruction pipeline 72

3.9 Visualisation of a volumetric SDF 73

3.10 Visualisation of reconstructed cortical surfaces on dHCP, HCP and ADNI datasets 81

3.11 Visualisation of cortical surfaces projected on MRI slices 82

3.12 The runtime of cortical surface reconstruction on dHCP, HCP and ADNI datasets 83

3.13 Comparisons on different ODE solvers 85

3.14 Cortical surfaces extracted by CortexODE-based pipeline based on imperfect segmentations 89

3.15 Ablation study for isosurface extraction of the initial surface 90

3.16 Ablation study for surface smoothing of the initial surface 91

3.17 Ablation study for CortexODE architecture 92

4.1 Neonatal cortical surfaces at different post-menstrual ages 99

4.2 Comparison between adult and neonatal brain MRI 99

4.3 The architecture of the proposed CoTAN framework 102

4.4 Diffeomorphic deformation from an initial mesh to cortical surfaces 105

4.5 Visualisation of neonatal cortical surfaces generated by the dHCP structural pipeline and CoTAN for different ages 109

4.6 Visualisation of attention maps predicted by CoTAN for cortical surface reconstruction 110

4.7 Visualisation of conditional time-varying velocity fields 111

4.8 The architecture of the TA-Net 116

4.9 Visualisation for the computation of different loss functions 118

4.10 Cortical surfaces predicted by implicit approaches on the HCP dataset . . . 121

4.11 Pial surfaces predicted by explicit approaches on the HCP dataset (2mm) . . 123

4.12 Cortical thickness and sulcal depth of cortical surfaces predicted by explicit learning-based approaches 124

4.13 The error maps of cortical thickness and sulcal depth for explicit learning-based approaches 124

4.14 Qualitative comparison of cortical surfaces on the dHCP fetal dataset 126

5.1 The distribution of PMA of neonates at birth and scan 136

5.2 The workflow of our DL-based dHCP neonatal structural brain MRI processing pipeline 137

5.3 The generation of initial image, surface and sphere templates 138

5.4 The deep learning-based approach for brain extraction 139

5.5 The architecture of the multiscale deformation network 141

5.6 The deformed surfaces generated by the multiscale deformation network . 142

5.7 Unsupervised learning-based spherical projection framework 148

5.8 The results of manual cortical surface quality control 154

5.9 The qualitative comparison between original and our DL-based dHCP pipeline on different test samples of the dHCP dataset 155

5.10 The visualisation of cortical surfaces and cortical features of different neonatal subjects 156

5.11 The cortical morphological features across 265 test neonatal subjects with different PMA at scan 157

5.12 The qualitative comparison between original and our DL-based dHCP pipeline on two neonatal subjects in the ePrime dataset 159

5.13 The visualisation of projected spheres for all spherical mapping approaches 161

Chapter 1

Introduction

1.1 Motivation and Objectives

Cortical surface reconstruction aims to extract the inner and outer surfaces of the cerebral cortex from brain structural magnetic resonance imaging (MRI) data. As illustrated in Figure 1.1, the inner cortical surface, also called the white matter surface, is the boundary between the white matter (WM) and the cortical grey matter (cGM). The outer cortical surface, also called the pial surface, is the boundary between the cGM and the cerebrospinal fluid (CSF). Cortical surfaces are normally represented by three-dimensional (3D) polygon meshes, which provide both 3D visualisation (Figure 1.2) and quantitative characterisation for the anatomical structure of the cerebral cortex. The reconstructed cortical surfaces facilitate the measurement of cortical morphological features such as cortical thickness, volume, and surface area (Dale et al., 1999; Fischl et al., 1999a; Fischl and Dale, 2000; Dahnke et al., 2013). The abnormalities of these features are important criteria for the detection and diagnosis of neurological (Aël Chetelat and Baron, 2003; Lerch et al., 2008; Querbes et al., 2009; McEvoy et al., 2009) and psychiatric disorders (Rimol et al., 2012; Madre et al., 2020; Patel et al., 2021). The morphological properties, such as mean curvature and sulcal depth, also play crucial roles in characterising cortical growth and gyrification as well as identifying atypical cortical folding patterns for developing fetal and neonatal brains (Nie et al., 2012; Wright et al., 2014; Garcia et al., 2018a,b; Edwards et al., 2022). Additionally, the reconstruction of cortical surfaces is an essential prerequisite for downstream tasks, for example, the parcellation of cortical regions (Fischl et al.,

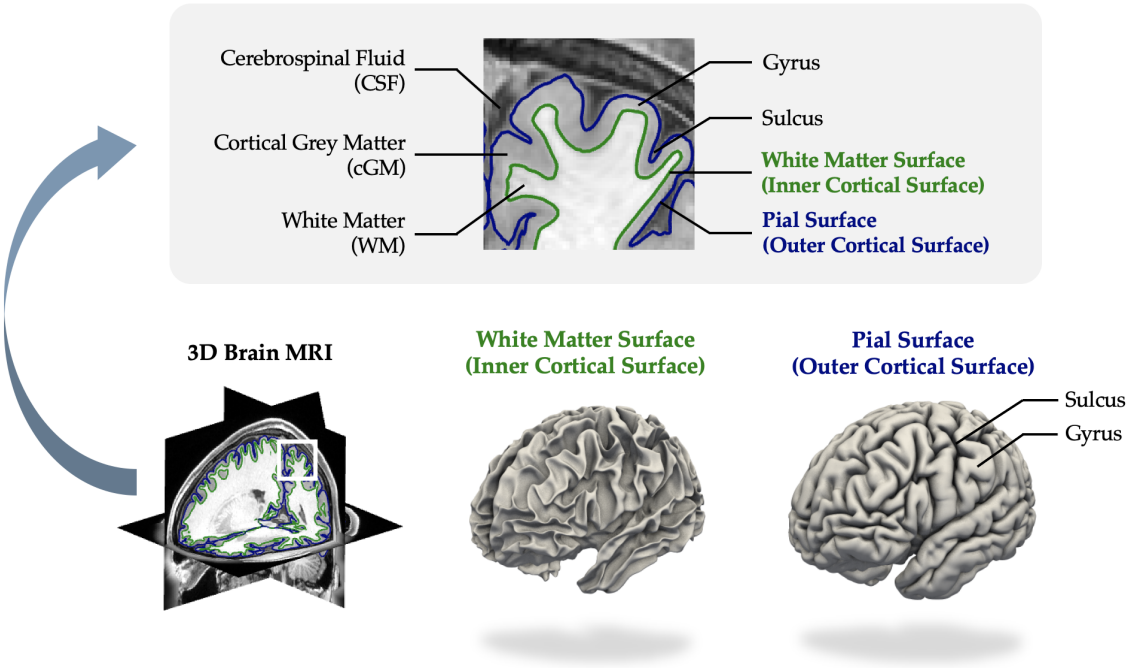


Figure 1.1: Illustration of cortical surface reconstruction. The cortical white matter (inner) and pial (outer) surfaces are extracted from a 3D brain MRI volume. The white matter surface (green) is the interface between the WM and the cGM. The pial surface (blue) is the interface between the cGM and the CSF. A *gyrus* is a ridge, and a *sulcus* is a groove in the cerebral cortex. The *cortical folds* are characterised by cortical gyri and sulci.

2004; Desikan et al., 2006; Destrieux et al., 2010; Glasser et al., 2016), the visualisation of functional activation (Sereno et al., 1995; Tootell et al., 1997; Dale et al., 1999), and the registration of cortical surfaces for cross-sectional and longitudinal studies (Fischl et al., 1999b; Lyttelton et al., 2007; Wright et al., 2014; Somerville et al., 2018; Garcia et al., 2018b).

Cortical surface reconstruction is challenging due to the complicated geometric structure of the highly folded and convoluted cerebral cortex (Figure 1.2). Besides, the cortical surfaces of each brain hemisphere are required to be closed and topologically equivalent to a 2-sphere, *i.e.*, no holes and handles (Dale et al., 1999; Fischl et al., 1999a,b, 2001). The deep and narrow cortical sulci, as shown in Figure 1.1, further underscore the necessity to prevent self-intersections and collisions on both inner and outer cortical surfaces. In addition to the inherent geometric and topological properties, the anatomical accuracy of cortical surfaces also relies on the quality of brain MRI. The partial volume effects, occurring when multiple tissue types are assigned to a single voxel in volumetric medical

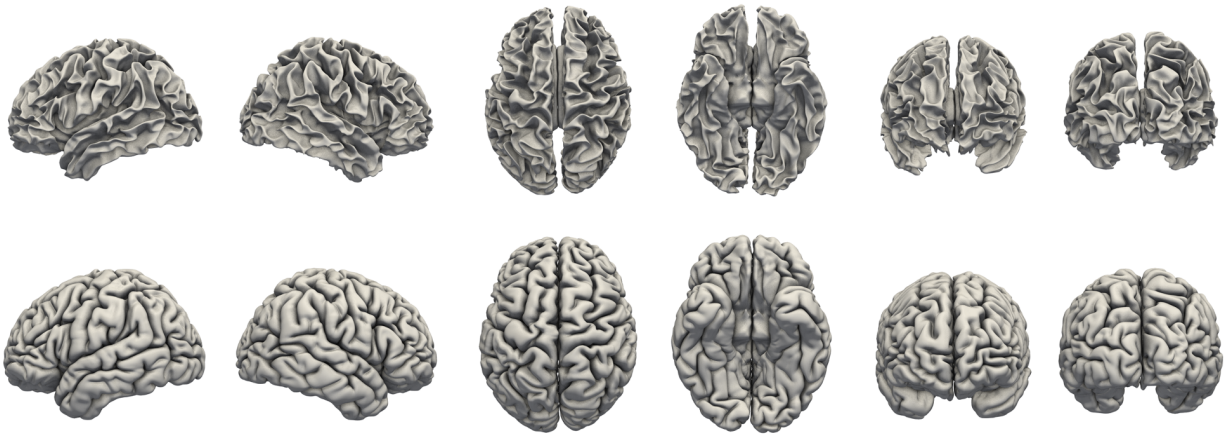


Figure 1.2: Visualisation of cortical surfaces from different perspectives. Top: white matter surface (inner cortical surface). Bottom: pial surface (outer cortical surface).

images (Ballester et al., 2002; Van Leemput et al., 2003; Tohka et al., 2004), pose extra difficulties in accurately fitting the subvoxel boundary of the cortex in structural brain MRI with limited image resolution. These challenges are especially pronounced in cortical surface reconstruction for fetal and neonatal brain MRI scans (Makropoulos et al., 2018; Howell et al., 2019; Price et al., 2019; Edwards et al., 2022), which have relatively low image resolution and contrast as a consequence of fast MRI sequences (Hughes et al., 2017) and restricted region of interests (ROI) caused by the small head size (Orasanu et al., 2014; Makropoulos et al., 2016). Moreover, the rapid development and folding of the cerebral cortex during the perinatal period lead to considerable variations in brain appearance and size for fetuses and neonates across different gestational and post-menstrual ages (Bethlehem et al., 2022; Edwards et al., 2022), as shown in Figure 1.3.

As an indispensable component in surface-based neuroimage analysis, cortical surface reconstruction has been tackled and integrated into existing automated brain MRI processing pipelines, *e.g.*, FreeSurfer (Dale et al., 1999; Fischl et al., 1999a; Fischl, 2012), CIVET (MacDonald et al., 2000; Kim et al., 2005, 2015), CRUISE (Xu et al., 1999; Han et al., 2004; Huo et al., 2016), BrainSuite (Shattuck and Leahy, 2002), the Human Connectome Project (HCP) pipeline (Glasser et al., 2013), and the developing HCP (dHCP) pipeline (Makropoulos et al., 2018). Typical cortical surface-based brain MRI processing pipelines

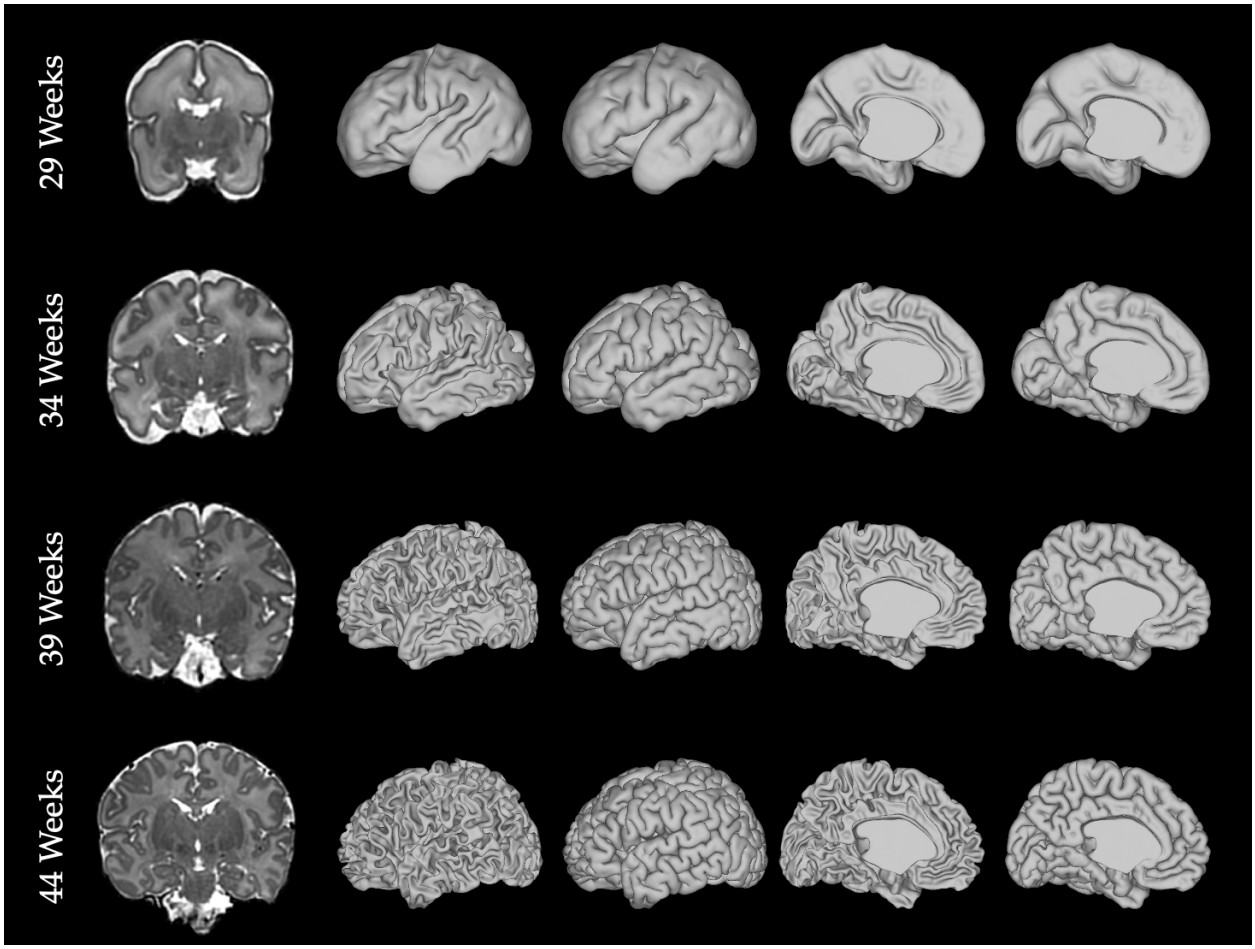


Figure 1.3: Visualisation of T2-weighted neonatal brain MRI and corresponding cortical surfaces for subjects scanned at 29 to 44 weeks.

employ a fully automated workflow consisting of skull stripping, intensity normalisation, brain segmentation, cortical surface reconstruction, cortical feature estimation, spherical projection, and cortical parcellation (Dale et al., 1999; Fischl et al., 1999a). These pipelines have offered invaluable services to the research community of neuroimaging. However, due to computationally intensive processing steps, traditional pipelines usually require several hours to process a single brain MRI scan. This makes it expensive to extract cortical surfaces for large-scale neuroimaging studies, for instance, the Alzheimer's Disease Neuroimaging Initiative (ADNI) (Jack Jr et al., 2008), the Open Access Series of Imaging Studies (OASIS) (Marcus et al., 2007), UK Biobank (Sudlow et al., 2015; Alfaro-Almagro et al., 2018), HCP (Van Essen et al., 2013), Baby Connectome Project (BCP) (Howell et al., 2019), and dHCP (Edwards et al., 2022) databases. Furthermore, considering the substan-

tial differences between adult and infant brain MRI, it is also difficult to develop a universal pipeline generalisable to subjects across different ages. These limitations of traditional pipelines highlight the demand for fast and versatile cortical surface reconstruction approaches for brain MRI processing in all age groups.

The advent and development of deep learning over the past years provide an alternative solution for the cortical surface reconstruction problem (LeCun et al., 2015; Goodfellow et al., 2016). Deep learning can be rapidly developed and deployed as a consequence of its complete open-source software libraries (Abadi et al., 2016; Paszke et al., 2019) and fast parallel computation supported by modern Graphics Processing Units (GPU) (Owens et al., 2008; Keckler et al., 2011). Deep learning techniques have been extensively applied in computer vision (LeCun et al., 1995; Deng et al., 2009; Krizhevsky et al., 2012; He et al., 2016) and medical image analysis (Ronneberger et al., 2015; Litjens et al., 2017; Shen et al., 2017; Zhou et al., 2023), providing a promising avenue for cortical surface reconstruction. Previous deep learning-based approaches tackled 3D shape modelling (Mescheder et al., 2019; Park et al., 2019; Peng et al., 2020) and cortical surface reconstruction (Henschel et al., 2020; Cruz et al., 2021) by learning implicit surface representations, *e.g.*, segmentations, occupancy fields, and signed distance functions. The explicit triangular meshes are reconstructed from the implicit surfaces via isosurface extraction algorithms such as Marching Cubes (Lorensen and Cline, 1998). These implicit surface reconstruction approaches do not enforce the topology of underlying surfaces, enabling the representation of complicated real-world scenes and objects. For cortical surface reconstruction, however, the spherical topology should be guaranteed by time-consuming topology correction algorithms (Fischl et al., 2001; Shattuck and Leahy, 2001; Han et al., 2002; Bazin and Pham, 2005, 2007; Ségonne et al., 2007; Yotter et al., 2011).

Recent advances of geometric deep learning have demonstrated its powerful representation learning ability in non-Euclidean domains such as point clouds, graphs, meshes and manifolds (Kipf and Welling, 2016; Gilmer et al., 2017; Qi et al., 2017a; Wang et al., 2018;

Veličković et al., 2018; Bronstein et al., 2017, 2021). Instead of learning implicit surface representations, an increasing number of geometric deep learning frameworks have been proposed for fast and explicit 3D shape reconstruction (Wang et al., 2018; Wen et al., 2019; Wickramasinghe et al., 2020; Gupta and Chandraker, 2020). These approaches generate explicit 3D meshes within only a few seconds by iteratively deforming an initial template mesh to the target shape. Such geometric deep learning approaches offer great potential for explicit cortical surface reconstruction, which is expected to be orders of magnitude faster compared to traditional neuroimage processing pipelines (MacDonald et al., 2000; Shattuck and Leahy, 2002; Han et al., 2004; Fischl, 2012; Glasser et al., 2013; Dai et al., 2013; Huo et al., 2016; Makropoulos et al., 2018).

The objective of this thesis is to provide and present accurate, efficient and versatile cortical surface reconstruction approaches, leveraging advanced 3D geometric deep learning techniques. In this thesis, we aim to address the following research questions:

- Learning explicit cortical surface reconstruction end-to-end from brain MRI.
- Learning time and memory efficient cortical surface reconstruction.
- Learning topology-preserving cortical surface reconstruction without self-intersections.
- Learning cortical surface reconstruction for developing fetal and neonatal brains.
- Learning cortical surfaces from noisy and limited ground truth data.

1.2 Contributions and Thesis Overview

In this thesis, we present a series of fast and robust geometric deep learning frameworks for explicit cortical surface reconstruction from adult, fetal and neonatal brain MRI scans. Specifically, we first propose a novel 3D geometric deep neural network model (Ma et al., 2021), which extracts cortical surfaces with spherical topology end-to-end from adult brain MRI within one second, supervised by pseudo ground truth (pGT) cortical surfaces generated by traditional neuroimage processing pipelines (Fischl, 2012; Glasser et al., 2013; Makropoulos et al., 2018). Based on the proposed neural network architecture (Ma et al., 2021), we further introduce diffeomorphic surface deformations (Ma et al., 2022) to prevent self-intersections in cortical surfaces, of which the deformations are modelled as neural ordinary differential equations (ODE) parameterised by deep neural networks (Coddington et al., 1956; Chen et al., 2018; Gupta and Chandraker, 2020).

In contrast to adult cortical surface reconstruction (Ma et al., 2021, 2022), it is more challenging to extract cortical surfaces from fetal and neonatal brains that experience rapid development during the perinatal period. To capture the large deformations and variations in neonatal brains scanned at different ages, we devise a customised deep learning-based approach for neonatal cortical surface reconstruction (Ma et al., 2023). Compared to neonatal brain MRI, the fetal brains were scanned *in utero*, resulting in lower image resolution, and thus traditional pipelines (Makropoulos et al., 2018) fail to provide acceptable pGT fetal cortical surfaces. To avoid the dependence on pGT cortical surfaces, we present a novel weakly supervised learning framework for fetal cortical surface reconstruction (Ma et al., 2024) supervised only by brain segmentation maps.

Apart from end-to-end cortical surface reconstruction (Ma et al., 2023, 2024), a fast and complete neuroimage processing pipeline, incorporating brain MRI preprocessing and downstream cortical surface-based analysis, is in demand for fetal and neonatal brain MRI analysis. Inspired by existing brain MRI processing pipelines (Fischl, 2012; Glasser

et al., 2013; Makropoulos et al., 2018), we develop a deep learning-based cortical surface reconstruction pipeline (Ma et al., 2025) to process and analyse the structural MRI of developing human brains, which only needs 24 seconds to process a single brain MRI scan accelerated by a modern GPU.

Compared to traditional neuroimage processing pipelines (MacDonald et al., 2000; Shattuck and Leahy, 2002; Han et al., 2004; Kim et al., 2005; Fischl, 2012; Dai et al., 2013; Glasser et al., 2013; Huo et al., 2016; Makropoulos et al., 2018) and existing implicit deep learning-based cortical surface reconstruction approaches (Henschel et al., 2020; Cruz et al., 2021), we utilise geometric deep learning techniques to accomplish fast and explicit cortical surface reconstruction, in particular with the following advantages:

- **Accuracy.** Proposed deep learning-based cortical surface reconstruction approaches extract anatomically accurate and topologically correct cortical surfaces end-to-end from brain MRI, with theoretical prevention of surface self-intersections.

- **Efficiency.** Proposed deep learning-based cortical surface reconstruction approaches achieve fast inference and reduce the processing time for a single brain MRI scan from several hours required by traditional pipelines to a few seconds only.

- **Versatility.** Proposed deep learning-based cortical surface reconstruction approaches can be trained and adapted to various age groups, including adult, fetal and neonatal cortical surface reconstruction.

The remaining contents of this thesis are organised as follows:

- In Chapter 2, we introduce necessary preliminaries and background of this thesis. We first provide basic concepts and definitions for topology, manifold, and ODE. Then, we review the fundamentals of geometric deep learning and its applications on surface reconstruction. Furthermore, we describe the problem formulation and methodology for cortical surface reconstruction.

- In Chapter 3, we propose a fast and novel 3D deep neural network architecture to learn iterative deformations for explicit cortical pial surface reconstruction from 3D brain MRI volumes. The proposed architecture is further extended to learn diffeomorphic surface deformations via Neural ODEs (Chen et al., 2018) for both inner and outer cortical surface reconstruction. Both theoretical guarantees and numerical conditions are provided to prevent surface self-intersections. The performance is evaluated on large public brain MRI datasets of newborn infants, young and old adults.

- In Chapter 4, we focus on cortical surface reconstruction for developing fetal and neonatal brains. We introduce an attention mechanism to learn coarse-to-fine diffeomorphic deformations for neonatal cortical surface reconstruction. The deformations are also conditioned on the post-menstrual ages of neonates to capture the vast differences in neonatal brain MRI scanned at different ages. In addition, we devise a weakly supervised cortical surface reconstruction framework for fetal brain MRI, supervised by cortical ribbon segmentations when traditional neuroimage processing pipelines (Makropoulos et al., 2018) fail to produce acceptable pGT cortical surfaces.

- In Chapter 5, we present a fast and robust deep learning-based processing pipeline for neonatal cortical surface reconstruction. Following the workflow of dHCP neonatal brain MRI processing pipeline (Makropoulos et al., 2018), our pipeline replaces current processing steps with novel deep learning-based approaches and GPU-accelerated re-implementations. Compared to the original dHCP pipeline (Makropoulos et al., 2018), our learning-based pipeline achieves similar functionality and superior surface quality, while being nearly 1000 times faster.

- In Chapter 6, we conclude and summarise the achievements of this thesis. We also provide future directions to address current limitations and improve deep learning-based cortical surface reconstruction.

1.3 Publications

This thesis draws upon the content of the following publications produced during my PhD studies:

1. **Ma, Q.**, Robinson, E.C., Kainz, B., Rueckert, D. and Alansary, A. (2021). PialNN: A Fast Deep Learning Framework for Cortical Pial Surface Reconstruction. In *International Workshop on Machine Learning in Clinical Neuroimaging*, pages 73-81. Springer. https://doi.org/10.1007/978-3-030-87586-2_8

2. **Ma, Q.**, Li, L., Robinson, E.C., Kainz, B., Rueckert, D. and Alansary, A. (2022). CortexODE: Learning Cortical Surface Reconstruction by Neural ODEs. *IEEE Transactions on Medical Imaging*, 42(2):430-443. https://doi.org/10.1109/TMI.2022.3206221

3. **Ma, Q.**, Li, L., Kyriakopoulou, V., Hajnal, J.V., Robinson, E.C., Kainz, B. and Rueckert, D. (2023). Conditional Temporal Attention Networks for Neonatal Cortical Surface Reconstruction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 312-322. Springer. https://doi.org/10.1007/978-3-031-43901-8_30

4. **Ma, Q.**, Li, L., Robinson, E.C., Kainz, B. and Rueckert, D. (2024). Weakly Supervised Learning of Cortical Surface Reconstruction from Segmentations. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 766-777. Springer. https://doi.org/10.1007/978-3-031-72120-5_71

5. **Ma, Q.**, Liang, K., Li, L., Masui, S., Guo, Y., Nosarti, C., Robinson, E.C., Kainz, B. and Rueckert, D. (2024). The Developing Human Connectome Project: A Fast Deep Learning-based Pipeline for Neonatal Cortical Surface Reconstruction. *Medical Image Analysis*, 100:103394. https://doi.org/10.1016/j.media.2024.103394

The following publications, which are not presented in detail, are also relevant to the objective of this thesis and focus on topology-preserving brain MRI segmentation:

1. Li, L., **Ma, Q.**, Li, Z., Ouyang, C., Zhang, W., Price, A., Kyriakopoulou, V., Grande, L.C., Makropoulos, A., Hajnal, J. and Rueckert, D., Kainz, B., Alansary, A. (2022). Fetal Cortex Segmentation with Topology and Thickness Loss Constraints. In *MICCAI Workshop on Topological Data Analysis for Biomedical Imaging*, pages 123-133. Springer.

2. Li, L., **Ma, Q.**, Ouyang, C., Li, Z., Meng, Q., Zhang, W., Qiao, M., Kyriakopoulou, V., Hajnal, J.V., Rueckert, D. and Kainz, B. (2023). Robust Segmentation via Topology Violation Detection and Feature Synthesis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 67-77. Springer.

3. Li, L., Wang, H., Baugh, M., **Ma, Q.**, Zhang, W., Ouyang, C., Rueckert, D. and Kainz, B. (2024). Universal Topology Refinement for Medical Image Segmentation with Polynomial Feature Synthesis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 670-680. Springer.

Chapter 2

Background

In this chapter, we provide relevant background knowledge for this thesis. We first introduce the necessary definitions and mathematical preliminaries of topology, manifold, ordinary differential equations and flow. Next, we review the foundations of geometric deep learning, the non-Euclidean geometric domains, and associated deep neural network models, with emphasis on applications to surface reconstruction. Lastly, we formulate the cortical surface reconstruction problem, describe the methodology, and review existing automated surface-based neuroimage processing pipelines.

2.1 Mathematical Preliminaries

2.1.1 Topology and Manifold

In this section, we introduce fundamental concepts of general topology and manifold. These concepts are necessary for the characterisation of geometric and topological properties of cortical surfaces. Firstly, we introduce basic definitions related to a topological space (Hatcher, 2002; Edelsbrunner and Harer, 2010; Younes, 2010; Lee, 2012).

Topological Space. A *topology* on a set $\mathcal{X}$ is a collection $\mathcal{T}$ of subsets of $\mathcal{X}$, which are called open sets. A topology $\mathcal{T}$ satisfies:
(1) $\mathcal{X}, \varnothing \in \mathcal{T}$, (2) if $U_1, U_2 \in \mathcal{T}$, then $U_1 \cap U_2 \in \mathcal{T}$, and (3) if $U_i \in \mathcal{T}$, then $\bigcup_i U_i \in \mathcal{T}$.
A *topological space* is defined as a pair $(\mathcal{X}, \mathcal{T})$, which can be simplified as a topological space $\mathcal{X}$ when the topology $\mathcal{T}$ is understood.

Homeomorphism. For topological spaces $\mathcal{X}$ and $\mathcal{Y}$, a bijection $f : \mathcal{X} \to \mathcal{Y}$ is called a *homeomorphism* if f and its inverse f^{-1} are continuous.

Two topological spaces $\mathcal{X}$ and $\mathcal{Y}$ are *topologically equivalent*, if $\mathcal{X}$ is *homeomorphic* to $\mathcal{Y}$, *i.e.*, there exists a homeomorphism $f : \mathcal{X} \to \mathcal{Y}$. A *topological invariant* or *topological property* is defined as the property of a topological space that is preserved under homeomorphisms. According to the definitions of the topological space and homeomorphism, we further introduce the definitions of a manifold (Do Carmo, 1992; Younes, 2010; Lee, 2012).

Manifold. A topological space $\mathcal{X}$ is called a *topological manifold of dimension n* or *topological n-manifold*, if $\mathcal{X}$ is Hausdorff, second-countable, and locally Euclidean of dimension n.
• *Hausdorff space*: for any distinct points $u, v \in \mathcal{X}$, there exist open sets $U, V \subseteq \mathcal{X}$ satisfying $u \in U$, $v \in V$ and $U \cap V = \varnothing$.
• *Second-countable*: there exists a countable basis for the topology $\mathcal{T}$ of space $\mathcal{X}$.
• *Locally Euclidean of dimension n*: each point $x \in \mathcal{X}$ has an open neighbourhood $U \subseteq \mathcal{X}$ that is homeomorphic to an open set $V \subseteq \mathbb{R}^n$.

A *differentiable (or smooth) manifold* is a topological manifold with a *differentiable (or smooth) structure*.

Diffeomorphism. For differentiable (or smooth) manifolds X and Y, a bijection $f : X \to Y$ is called a *diffeomorphism* if f and its inverse f^{-1} are differentiable (or smooth).

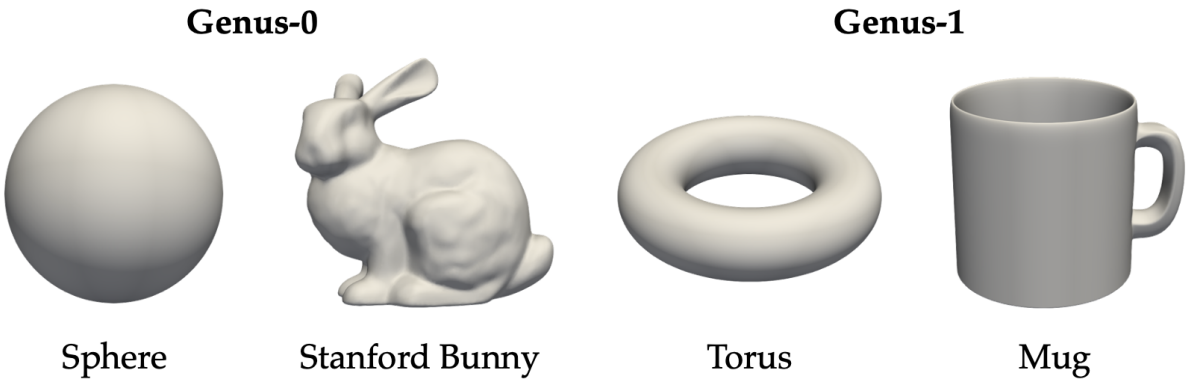


Figure 2.1: Surfaces with different topology and genus. Genus-0: a 2-sphere and a Stanford bunny. Genus-1: a torus and a mug.

In the context of differential geometry and topology (Do Carmo, 1992; Edelsbrunner and Harer, 2010; Younes, 2010; Lee, 2012), a *surface* defined in a three-dimensional (3D) Euclidean space can be considered as a *2-manifold*. The *genus* g of a closed, connected, and orientable surface $\mathcal{S}$ is a topologically invariant, defined as the maximum number of cuttings along disjoint closed curves that do not disconnect the surface $\mathcal{S}$. Equivalently, the genus of a closed surface is the number of holes or handles. For example, a sphere or a Stanford bunny has genus 0, and a torus or a mug has genus 1 as illustrated in Figure 2.1.

A surface can be discretised by a *triangular mesh* $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$, where $\mathcal{V}, \mathcal{E}, \mathcal{F}$ are the sets of *vertices*, *edges*, and *faces*. Another topological invariant of a discretised closed surface is called *Euler characteristic number*, which is defined as:

$$\chi = |\mathcal{V}| - |\mathcal{E}| + |\mathcal{F}| = 2 - 2g, \tag{2.1}$$

where $|\mathcal{V}|, |\mathcal{E}|, |\mathcal{F}|$ are the numbers of vertices, edges, and faces. The Euler characteristic number χ is also associated with the genus g of a surface.

2.1.2 Ordinary Differential Equations

In this section, we introduce the basic theory and numerical methods for *ordinary differential equations* (ODE), which is important to the construction of diffeomorphic surface deformations (Ma et al., 2022, 2023).

Initial Value Problem. A first order ODE is defined as (Coddington et al., 1956; Walter, 2013; Perko, 2013):

$$\frac{d}{dt}x(t) = f(x(t), t),\ t \in \mathcal{I}, \tag{2.2}$$

where $\mathcal{I} \in \mathbb{R}$ is a time interval, $x(t) \in \Omega \subseteq \mathbb{R}^n$ is a *solution* of the ODE (2.2), and function $f : \Omega \times \mathcal{I} \to \Omega$ is called the *right-hand side* (RHS) *function*. Furthermore, given an *initial condition* $x(t_0) = x_0$, an *initial value problem* of the ODE (2.2) is defined as:

$$\frac{d}{dt}x(t) = f(x(t), t),\ x(t_0) = x_0. \tag{2.3}$$

Existence and Uniqueness of Solutions. The sufficient conditions for the existence and uniqueness of solutions for the ODE initial value problem (2.3) are provided by the following theorem (Coddington et al., 1956; Walter, 2013):

Theorem 2.1 (Existence and Uniqueness Theorem)**.** *If the RHS function f is continuous on $\mathcal{D} = \Omega \times [t_0 - \delta, t_0 + \delta]$ and satisfies a Lipschitz condition in $\mathcal{D}$ with respect to x, i.e.,*

$$\|f(x, t) - f(\tilde{x}, t)\| \leq L\|x - \tilde{x}\|, \tag{2.4}$$

with Lipschitz constant $L \geq 0$, then there exists an unique solution $x(t)$ in $[t_0 - \delta, t_0 + \delta]$ for the initial value problem (2.3).

Theorem 2.1 is also known as *Picard-Lindelöf Theorem*, which can be proved by *Picard's method of successive approximations*. Theorem 2.1 can be extended to the entire integration interval $\mathcal{I}$ if the RHS function f is globally Lipschitz continuous for all $t \in \mathcal{I}$. The solution

is represented as $x(t) = x_0 + \int_0^t f(x(s), s)ds$. The Existence and Uniqueness Theorem guarantees that for any distinct initial values $x_0, y_0 \in \Omega$, the trajectories of the solutions $x(t), y(t)$ will never intersect with each other in the interval $\mathcal{I}$. Next, we introduce an autonomous ODE and its associated flow (Lee, 2012; Varadarajan, 2013; Perko, 2013).

Autonomous ODE. For $t_0 = 0$, an autonomous ODE is defined as:

$$\frac{d}{dt}x(t) = f(x(t)),\ x(0) = x_0, \tag{2.5}$$

where the RHS function $f : \Omega \to \Omega$ does not explicitly depend on time t. Hence, the RHS f is also called a *vector field*, or a *velocity field* for $\Omega \subseteq \mathbb{R}^3$.

Flow. For $s, t \in \mathbb{R}$ and $x \in \Omega$, a *flow* is a continuous mapping $\phi : \Omega \times \mathbb{R} \to \Omega$ satisfying

$$\phi(\phi(x, s), t) = \phi(x, t + s),\ \phi(x, 0) = x. \tag{2.6}$$

Let $\phi_t(x) := \phi(x, t)$ for any $t \in \mathbb{R}$. According to the definition (2.6), the flow ϕ constructs a *one-parameter group* (Lee, 2012; Varadarajan, 2013) with group laws:

$$\phi_t \circ \phi_s = \phi_{t+s},\ \phi_0 = Id_\Omega, \tag{2.7}$$

where $\circ$ is a group operation, ϕ_0 is the identity element of the group, and Id_Ω is an identity mapping on Ω. For any group element ϕ_t, its inverse mapping $(\phi_t)^{-1} = \phi_{-t}$ is also continuous. Therefore, ϕ_t is a homeomorphism for any $t \in \mathbb{R}$, and ϕ is called a *one-parameter group of homeomorphisms*. Furthermore, if ϕ_t is a diffeomorphism, *i.e.*, ϕ_t is differentiable (or smooth) with respect to $x \in \Omega$, then ϕ is called a *one-parameter group of diffeomorphisms*.

Define $\varphi_t(x_0) = \varphi(x_0, t) := x(t)$ for a solution $x(t)$ of the autonomous ODE (2.5). It can be verified that φ_t is a flow satisfying the definition (2.6). The flow φ_t is called a flow associated with or induced by the vector field f. The corresponding autonomous ODE

(2.5) can be rewritten as a flow ODE:

$$\frac{d}{dt}\varphi_t = f \circ \varphi_t,\ \varphi_0 = Id_\Omega. \tag{2.8}$$

If the vector field f is sufficiently smooth, then φ is a one-parameter subgroup of diffeomorphisms and f is an element of a *Lie algebra*. Based on Lie group theory (Arsigny, 2004; Lee, 2012; Varadarajan, 2013), the solution of the flow ODE (2.8), also called an *integral curve* of the vector field f, can be expressed by an exponential map $\varphi_t = \exp(tf)$.

Numerical Methods. We introduce classic numerical methods (Süli and Mayers, 2003; Butcher, 1987, 2016) for solving the ODE initial value problem defined in Equation (2.3).

- *Forward Euler Method.* A *forward Euler* or *explicit Euler method* is defined as the following iteration scheme:

$$x_{n+1} = x_n + hf\left(x_n, t_n\right), \tag{2.9}$$

  for $n = 0, ..., N-1$, where h is called a *step size*, $t_n = t_0 + nh$ is the current integration time step, and x_n is the approximation for the solution $x(t_n)$. The forward Euler method is an explicit scheme. Thus, given any initial value $x(t_0) = x_0$, we can iteratively calculate the numerical solutions x_n for $n = 1, ..., N$ according to Equation (2.9).

- *Backward Euler Method.* A *backward Euler* or *implicit Euler* method is defined as:

$$x_{n+1} = x_n + hf\left(x_{n+1}, t_{n+1}\right). \tag{2.10}$$

  Unlike the forward Euler method, the backward Euler is an implicit method as the unknown x_{n+1} appears on both sides of Equation (2.10), which needs to be solved to obtain the numerical solution x_{n+1}.

- *Midpoint Method.* Euler methods are first-order numerical methods, of which the global errors $e_n = x(t_n) - x_n$ are of order $\mathcal{O}(h)$ with respect to the step size h. Hence, Euler methods require a very small step size to reduce numerical errors, which increases

the computational burden. To achieve fast convergence, we introduce a second-order *midpoint method*:

$$x_{n+1} = x_n + hf\left(x_n + \frac{h}{2} f\left(x_n, t_n\right), t_n + \frac{h}{2}\right). \tag{2.11}$$

The midpoint method has second-order accuracy $\mathcal{O}(h^2)$ with respect to the step size h. It first computes the midpoint value $x_{n+1/2} = x_n + \frac{h}{2} f\left(x_n, t_n\right)$ and then approximate the solution x_{n+1}.

- *Runge-Kutta Method.* Both Euler and midpoint methods can be viewed as members of the family of Runge-Kutta (RK) methods (Butcher, 1987; Süli and Mayers, 2003). The general form of s-stage RK methods for the ODE initial value problem (2.3) is defined by the following scheme:

$$x_{n+1} = x_n + h \sum_{i=1}^{s} \beta_i k_i, \tag{2.12}$$

where k_i is defined as follows:

$$k_i = f\Big(x_n + h \sum_{j=1}^{s} \alpha_{ij} k_j, t_n + \gamma_i h\Big). \tag{2.13}$$

where coefficients α_{ij}, β_i and γ_i are determined according to a Butcher tableau (Butcher, 1963). The fourth-order RK (RK4) method is one of the most frequently used RK methods, which is defined as:

$$\begin{aligned} x_{n+1} &= x_n + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right), \\ k_1 &= f\left(x_n, t_n\right), \\ k_2 &= f\left(x_n + \frac{h}{2} k_1, t_n + \frac{h}{2}\right), \\ k_3 &= f\left(x_n + \frac{h}{2} k_2, t_n + \frac{h}{2}\right), \\ k_4 &= f\left(x_n + h k_3, t_n + h\right). \end{aligned} \tag{2.14}$$

- *Linear Multistep Method.* The RK family, including Euler, midpoint, and RK4 methods, is one-step method that computes the approximate solution x_{n+1} only based on one

previous step x_n. In contrast, a *linear k-step method* is formulated by:

$$\sum_{i=0}^{k} a_i x_{n+i} = h \sum_{i=0}^{k} b_i f(x_{n+i}, t_{n+i}), \tag{2.15}$$

where a_i and b_i are predefined coefficients. The solution x_{n+i} depends on the values of all k steps $x_n, ..., x_{n+k}$.

In addition to general numerical methods for first-order ODEs, we introduce the scaling and squaring method (Higham, 2005; Arsigny et al., 2006; Ashburner, 2007) to solve the flow ODE (2.8), which has been widely adopted in deformable medical image registration and computational anatomy (Arsigny, 2004; Ashburner, 2007, 2009; Sotiras et al., 2013; Dalca et al., 2018, 2019)

- *Scaling and Squaring.* Scaling and squaring method is originally devised to solve matrix exponential problems, which can be also applied to solve the flow ODE (2.8) by approximating the exponential map $\varphi_t = \exp(tf)$. The theoretical foundation of the scaling and squaring method is the group law of a flow $\varphi_{t+s} = \varphi_t \circ \varphi_s$, or equivalently the property of an exponential map $\exp((t+s)f) = \exp(tf)\exp(sf)$. For an integration interval $\mathcal{I} = [0, T]$, the solution φ_T can be approximated by recursively computing

$$\varphi_{T/2^n} = \varphi_{T/2^{n+1}} \circ \varphi_{T/2^{n+1}}, \tag{2.16}$$

for $n = 0, ..., N-1$, where N is the number of steps. The starting point is $\varphi_{T/2^N} = Id_\Omega + hf$ with step size $h = T/2^N$. The scaling and squaring method is effective and efficient in solving an autonomous ODE recurrently. In fact, a scaling and squaring method with N integration steps is equivalent to 2^N steps of forward Euler iteration.

2.2 Geometric Deep Learning

As an important subfield of deep learning, geometric deep learning has demonstrated its efficacy and efficiency in learning informative representations from non-Euclidean domains (Kipf and Welling, 2016; Bronstein et al., 2017, 2021; Wu et al., 2020). A growing number of geometric deep learning approaches have been developed and applied to a wide range of areas, for instance, exploring underlying correlations in social recommend systems (Ying et al., 2018; Fan et al., 2019; Wu et al., 2022), designing the molecular graphs for drug discovery (Duvenaud et al., 2015; Gilmer et al., 2017; Sun et al., 2020; Gaudelet et al., 2021), solving combinatorial optimisation problems (Khalil et al., 2017; Nazari et al., 2018; Cappart et al., 2023), and generating polygon meshes for shape modelling (Chang et al., 2015; Groueix et al., 2018; Wang et al., 2018; Wickramasinghe et al., 2020). In this section, we will introduce the fundamentals of geometric deep learning, in particular focusing on its applications to 3D shape reconstruction and analysis.

2.2.1 Deep Learning Foundations

Deep learning is a branch of machine learning using deep neural network models to automatically learn underlying representations from large datasets (LeCun et al., 2015; Goodfellow et al., 2016). In this part, we provide a brief introduction to the foundations of machine learning and deep learning.

Machine Learning. We first review the basics of machine learning (Jordan and Mitchell, 2015; Deisenroth et al., 2020). Given an annotated dataset $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$ with input data $\mathcal{X} = \{x_i\}_{i=1}^N$ and the ground truth (GT) labels $\mathcal{Y} = \{y_i\}_{i=1}^N$, we divide the dataset into training $\mathcal{D}_{train}$, validation $\mathcal{D}_{val}$, and test sets $\mathcal{D}_{test}$. A supervised machine learning task aims to fit or train a model $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ based on the training set $\mathcal{D}_{train}$ to make predictions $\hat{y} = f_\theta(x)$ for the test data $x \in \mathcal{X}_{test}$. Here θ are learnable parameters, which can be learned by minimising a loss function $\hat{\theta} = \min_{\theta \in \Theta} \mathcal{L}(f_\theta(\mathcal{X}_{train}), \mathcal{Y}_{train})$ via various machine

learning algorithms. The validation dataset $\mathcal{D}_{val}$ is employed to assess the performance of the learned model during training. While supervised machine learning tasks involve classification and regression such as Logistic regression, decision tree and support vector machine (SVM), unsupervised machine learning involves clustering and dimension reduction such as K-means clustering and principal component analysis (PCA) (Jordan and Mitchell, 2015; Deisenroth et al., 2020).

Deep Learning. To capture effective representations for accurate prediction, machine learning usually requires laborious feature engineering to extract meaningful features from input data. In contrast, deep learning leverages deep neural network models to facilitate automatic feature extractions, leading to considerable improvement compared to conventional machine learning techniques (LeCun et al., 2015; Goodfellow et al., 2016). One basic class of deep neural networks is the multi-layer perceptron (MLP). Each layer of a MLP is formulated as $g(x) = \phi(Wx + b)$, where W and b are the learnable weight and bias, and ϕ is a non-linear activation function, for example, Sigmoid, hyperbolic tangent (Tanh), and rectified linear unit (ReLU) functions (Goodfellow et al., 2016). Besides vanilla MLPs, typical deep neural network models consist of convolutional neural networks (CNN) (LeCun et al., 1995) such as AlexNet (Krizhevsky et al., 2012), Inception (Szegedy et al., 2015), ResNet (He et al., 2016) and U-Net (Ronneberger et al., 2015), recurrent neural networks (RNN) such as LSTM (Hochreiter and Schmidhuber, 1997) and GRU (Cho et al., 2014), attention models such as Transformers (Vaswani et al., 2017; Dosovitskiy et al., 2021), as well as graph neural networks (Kipf and Welling, 2016; Gilmer et al., 2017; Veličković et al., 2018) designed for geometric deep learning.

A deep neural network model f_θ is trained by a backpropagation algorithm (LeCun et al., 1989), which computes and tracks the gradients $\partial\mathcal{L}/\partial\theta$ of a loss function $\mathcal{L}(\theta)$ with respect to all learnable parameters θ via the chain rule. The parameters are updated by gradient descent-based optimisation, *e.g.*, stochastic gradient descent (SGD) and Adam (Goodfellow et al., 2016; Kingma and Ba, 2015), to minimise the loss function. After training, we

select the learned parameters $\hat{\theta}$ with the best performance on the validation dataset for the model $f_{\hat{\theta}}$ to make predictions and inferences.

2.2.2 Geometric Domains

Unlike standard deep learning frameworks that extract features from Euclidean space, geometric deep learning is known for its powerful representation learning in non-Euclidean domains (Bronstein et al., 2017, 2021). In this part, we introduce general non-Euclidean geometric domains (Bronstein et al., 2021) in computer vision and graphics, with particular emphasis on those commonly used for 3D shape representation. These geometric domains are visualised in Figure 2.2.

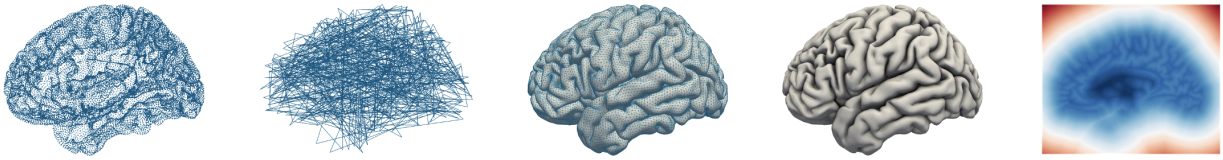


Figure 2.2: Non-Euclidean geometric domains for 3D shape representation. From left to right: point cloud, graph, mesh, manifold, and signed distance function (SDF).

Point Cloud. A *point cloud* $\mathcal{P}$ is defined a set of 3D data points $p \in \mathcal{P} \subset \mathbb{R}^3$. Each point has a coordinate representation $p = (x, y, z)$ with an optional normal direction $n(p) \in \mathbb{R}^3$. Point clouds are normally generated by 3D scanners or LiDAR sensors to depict the surface of the scanned object and scene (Guo et al., 2020). For 3D point cloud processing, the common geometric deep learning tasks include point cloud classification (Qi et al., 2017a,b; Qian et al., 2022), segmentation (Wang et al., 2019; Zhao et al., 2021c; Guo et al., 2021), completion (Yuan et al., 2018; Wang et al., 2020; Yu et al., 2021b, 2022), registration (Wang and Solomon, 2019; Lu et al., 2019; Aoki et al., 2019), and surface reconstruction from point clouds (Hanocka et al., 2020; Peng et al., 2021; Vahdat et al., 2022). Point clouds are able to characterise the shape of 3D objects or real-world scenes, but cannot represent their topological structures.

Graph. A *graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined by a set of *vertices* (or *nodes*) $\mathcal{V}$ and a set of *edges* (or *links*) $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Each edge $e = (i, j) \in \mathcal{E}$ connects the vertices $v_i, v_j \in \mathcal{V}$ to model their relations and interactions. A *directed graph* is a graph with ordered edges (i, j), and a *undirected graph* is a graph with unordered edges $\{i, j\}$. In other words, for an edge $(i, j) \in \mathcal{E}$, it holds $(j, i) \in \mathcal{E}$ for an undirected graph. The set of all adjacent vertices of the i-th vertex v_i is defined as $\mathcal{N}(i) = \{v_j : (i, j) \in \mathcal{E}\}$. An *adjacency matrix* $\mathcal{A}$ is a $n \times n$ matrix with each entry $a_{ij} = 1$ for $(i, j) \in \mathcal{E}$ and $a_{ij} = 0$ for $(i, j) \notin \mathcal{E}$, where $n = |\mathcal{V}|$ denotes the number of vertices. An adjacency matrix of an undirected graph is symmetric. For an undirected graph, a *degree matrix* $\mathcal{D}$ is a $n \times n$ diagonal matrix with diagonal entry $d_{ii} = |\mathcal{N}(i)|$, where $|\mathcal{N}(i)|$ is the number of adjacent vertices of a vertex v_i, also called the degree of v_i. Then, a graph *Laplacian matrix* is defined by $\mathcal{L} = \mathcal{D} - \mathcal{A}$.

Mesh. A *polygon mesh* $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ is defined by sets of *vertices* $\mathcal{V}$, *edges* $\mathcal{E}$, and *faces* $\mathcal{F}$. On the basis of a graph, a mesh further introduces the concept of a face, *i.e.*, a polygon defined by vertices and edges. In a triangular mesh, for instance, a face $f = (i, j, k) \in \mathcal{F}$ is a triangle defined by vertices $v_i, v_j, v_k \in \mathcal{V}$ and edges $(i, j), (j, k), (k, i) \in \mathcal{E}$. Generally, a polygon mesh is embedded in 3D Euclidean space as a tessellation on the surface of any 3D shapes. Each vertex v_i has a 3D coordinate $x_i \in \mathbb{R}^3$ and a unit normal vector $n_i \in \mathbb{R}^3$. Each face f also has a normal vector $n(f)$ which is orthogonal to the face. As the discrete representation of an explicit surface, a polygon mesh provides not only the geometric structure, but also the topological property of the surface via an Euler characteristic number defined in Equation (2.1). Since a mesh can be viewed as a special case of an undirected graph, we can also define a Laplacian matrix based on mesh connectivity, which is crucial to mesh deformation and smoothing (Taubin, 1995a,b; Desbrun et al., 1999; Sorkine et al., 2004; Nealen et al., 2006; Botsch and Sorkine, 2007).

Manifold. As defined in Section 2.1.1, a *topological n-manifold* Ω (Do Carmo, 1992; Lee, 2012) is a second-countable Hausdorff space that is locally Euclidean of dimension n, *i.e.*, for each point $x \in \Omega$, there exists an open neighbourhood U of x homeomorphic to an

open set $V \subset \mathbb{R}^n$. According to the definition of a manifold, a *discrete manifold* or *manifold mesh* is defined as a polygon mesh without non-manifold vertices, non-manifold faces, and self-intersections (Crane, 2018; Gupta and Chandraker, 2020; Bronstein et al., 2017, 2021) as illustrated in Figure 2.3. Such discretisation of a manifold enables the approximation of Riemannian metrics, curvatures and geodesics for the description of intrinsic geometric properties (Do Carmo, 1992; Crane, 2018; Bronstein et al., 2021).

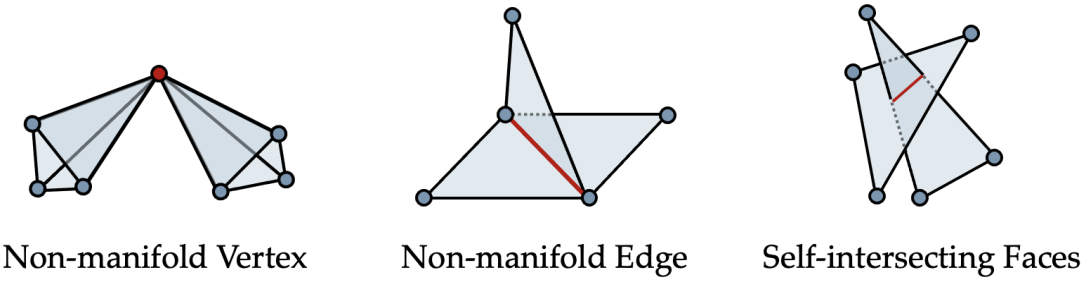


Figure 2.3: Categories of non-manifold geometries. Non-manifold vertex: a vertex shared by non-adjacent groups of faces. Non-manifold edge: an edge shared by more than two faces. Self-intersecting faces: two faces intersect with each other.

Signed Distance Function. Let $\mathcal{S} \in \mathbb{R}^n$ be a closed surface. A *signed distance function* (SDF) (Malladi et al., 1995; Curless and Levoy, 1996; Park et al., 2019) is defined as $f(x) = I_S(x)d(x, S)$, where $d(x, S)$ is the distance from a point $x \in \mathbb{R}^n$ to the surface S, and $I_S(x)$ is an indicator function. It holds that $I_S(x) = -1$ if $x \in \text{int } S$, *i.e.*, x is inside of the surface S, and $I_S(x) = 1$ if $x \notin \text{int } S$. For any constant $\lambda \in \mathbb{R}$, a SDF represents an isosurface as a level set $S(\lambda) = \{x : f(x) = \lambda\}$. Particularly, since the SDF $f(x) = 0$ holds when $x \in S$, the implicit representation of a surface S is naturally defined by $S = S(0) = \{x : f(x) = 0\}$. The surface normal of an implicit surface is defined by the gradient $n(x) = \nabla f(x)$.

2.2.3 Geometric Deep Neural Network Models

The outstanding feature learning ability of geometric deep learning in non-Euclidean space can be attributed to the advanced architecture of geometric deep neural network models (Duvenaud et al., 2015; Defferrard et al., 2016; Kipf and Welling, 2016; Gilmer

et al., 2017; Veličković et al., 2018; Qi et al., 2017a,b; Cohen et al., 2018; Wang et al., 2018; Park et al., 2019; Wickramasinghe et al., 2020; Gupta and Chandraker, 2020; Mildenhall et al., 2021; Bronstein et al., 2017, 2021). We briefly review representative deep neural network models for geometric deep learning in this section.

PointNet. PointNet (Qi et al., 2017a) and its variants (Qi et al., 2017b; Qian et al., 2022; Aoki et al., 2019) are popular neural network models to tackle 3D point clouds problems. As shown in Figure 2.4, given an unordered input point cloud, the PointNet (Qi et al., 2017a) first learns a spatial transformation applying to the input points via matrix multiplication. Subsequently, a shared MLP layer for all points is employed to extract features from the transformed points. These features are further transformed by another learned matrix. The transformed features are passed to a sequence of MLP layers, followed by a max pooling layer to extract a global feature vector for the whole point cloud. Such a global feature vector is used for object classification or concatenated to the transformed features for point cloud segmentation.

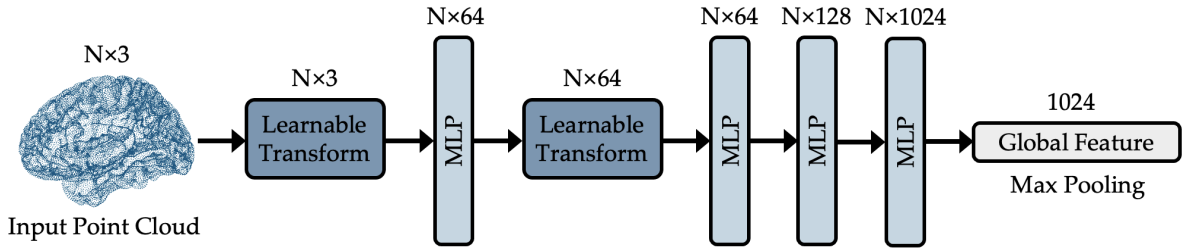


Figure 2.4: The architecture of the PointNet (Qi et al., 2017a). A global feature vector is learned by the PointNet from an input 3D point cloud. N is the number of points.

Graph Neural Networks. Graph Neural Networks (GNN) play a vital role in geometric deep learning (Duvenaud et al., 2015; Defferrard et al., 2016; Gilmer et al., 2017; Bronstein et al., 2017; Hamilton et al., 2017; Veličković et al., 2018; Xu et al., 2018; Wu et al., 2019a, 2020; Bronstein et al., 2021). The novel architectures of GNNs extend the domains of deep learning from Euclidean grids to arbitrary non-Euclidean graphs and mesh-based surfaces. GNNs have been extensively applied to various fields *e.g.*, recommendation systems and social networks (Ying et al., 2018; Fan et al., 2019; Wu et al., 2022), as well

as molecular chemistry and biology (Duvenaud et al., 2015; Gilmer et al., 2017; Sun et al., 2020; Gaudelet et al., 2021). Overall, there are three main categories of GNNs (Kipf and Welling, 2016; Veličković et al., 2018; Gilmer et al., 2017; Bronstein et al., 2021):

- *Graph Convolutional Network*. For any graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a Graph Convolutional Network (GCN) (Kipf and Welling, 2016) can be abstracted as:

$$y_i = f_\theta \left(x_i, \bigoplus_{j \in \mathcal{N}(i)} w_{ij} g_\theta(x_j) \right), \tag{2.17}$$

  where x_i is the input feature vector of the i-th vertex, and y_i is the output feature. f_θ and g_θ are learnable neural networks such as MLPs. w_{ij} are the weights of the edge $(i, j) \in \mathcal{E}$. The symbol $\bigoplus$ denotes the aggregation operator, such as summation, mean, maximum, and multiplication, over all adjacent vertices. The GCN is a pioneer geometric deep learning model, generalising the convolutional operation from regular grids in Euclidean space to any non-Euclidean graphs. Since a grid can be viewed as a graph, a CNN can be considered as a special case of a GCN in Euclidean space.

- *Graph Attention Network*. A Graph Attention Network (GAT) (Veličković et al., 2018) is defined as:

$$y_i = f_\theta \left(x_i, \bigoplus_{j \in \mathcal{N}(i)} a_\theta(x_i, x_j) g_\theta(x_j) \right), \tag{2.18}$$

  where a_θ is a learnable self-attention module. Unlike GCNs that use fixed edge weights, GATs parameterise the importance between neighbouring vertices via a learnable self-attention mechanism.

- *Message Passing*. Message Passing network is a more general GNN (Gilmer et al., 2017) formulated as:

$$y_i = f_\theta \left(x_i, \bigoplus_{j \in \mathcal{N}(i)} g_\theta(x_i, x_j) \right), \tag{2.19}$$

  where the interactions between any pairs of adjacent vertices are learned directly by the message function g_θ. The messages are passed by the aggregation operator.

Pixel2Mesh. Pixel2Mesh (Wang et al., 2018; Wen et al., 2019) and Voxel2Mesh (Wickramasinghe et al., 2020) are geometric deep learning frameworks for explicit 3D mesh reconstruction from 2D/3D images and point clouds. The basic architecture is illustrated in Figure 2.5. Pixel2Mesh or Voxel2Mesh iteratively subdivides and deforms an input template mesh, *e.g.*, a sphere or an ellipsoid, into the target shape by multiple GCN layers. The inputs of each GCN layer consist of the coordinates of mesh vertices as well as the features extracted from the input 2D/3D images by convolutional layers, while the output is the displacements of mesh vertices. The final prediction is a high-resolution 3D mesh corresponding to the input image. Pixel2Mesh and Voxel2Mesh frameworks incorporate multiple graph unpooling and deformation blocks to learn coarse-to-fine surface deformations, and thus achieve superior geometric accuracy in 3D shape reconstruction.

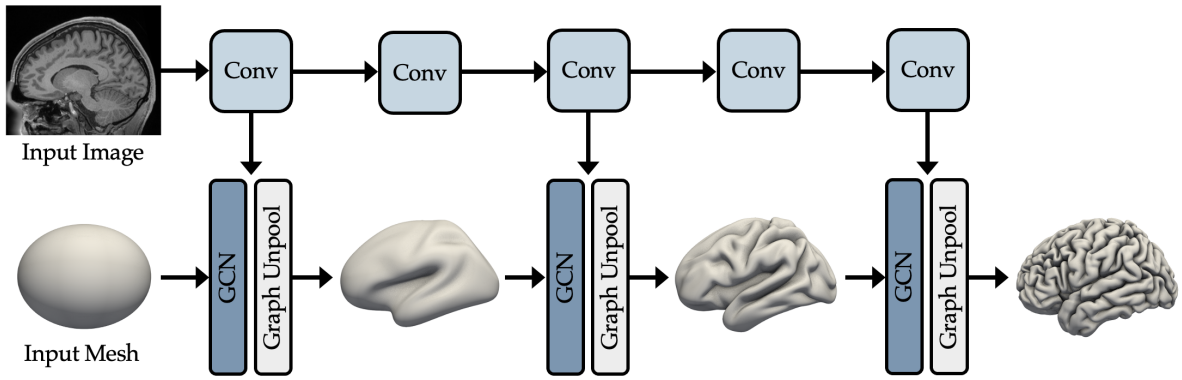


Figure 2.5: The architecture of the Pixel2Mesh (Wang et al., 2018) and Voxel2Mesh (Wickramasinghe et al., 2020) frameworks. An input 2D/3D image is encoded by a series of convolutional layers. The extracted features are passed into multiple GCN layers to deform an initial template mesh into the target shape.

Neural Mesh Flow. Neural Mesh Flow (Gupta and Chandraker, 2020) is devised to learn the 3D reconstruction of high-quality manifold meshes for physically based simulation and rendering. It trains multiple Neural ODE modules (Chen et al., 2018) to learn diffeomorphic flows that deform a template ellipsoidal mesh to the desired shape. As the architecture shown in Figure 2.6, Neural Mesh Flow is composed of three deformation blocks, each of which comprises two Neural ODE modules as well as an instance normalisation layer such that the Neural ODE can model shapes with different variances. Each

Neural ODE block f_θ uses a MLP layer to extract features from each input vertex $x_0 \in \mathbb{R}^3$. The extracted features are multiplied by a global latent vector z learned from input point clouds or 2D/3D images as a shape descriptor, and then passed to residual MLP layers to predict the mesh deformation. Such a Neural ODE f_θ learns a diffeomorphic conditional flow defined by the following ODE initial value problem (Coddington et al., 1956; Younes, 2010; Walter, 2013; Perko, 2013):

$$\frac{d}{dt}x(t) = f_\theta(x(t), z),\ x(0) = x_0,\ t \in [0, T], \tag{2.20}$$

where $x(t) \in \mathbb{R}^3$ is the trajectory of vertices. The mesh is deformed by integrating the ODE (2.20) for each vertex $x(T) = x_0 + \int_0^T f_\theta(x(t), z)dt$. According to the existence and uniqueness of ODE solutions (Coddington et al., 1956; Walter, 2013; Chen et al., 2018), if f_θ is Lipschitz continuous with respect to $x(t)$, the trajectories of two distinct vertices will never intersect with each other, which means Neural ODE-based deformations can theoretically prevent surface self-intersections. Therefore, Neural Mesh Flow is able to reconstruct manifold meshes without self-intersecting faces.

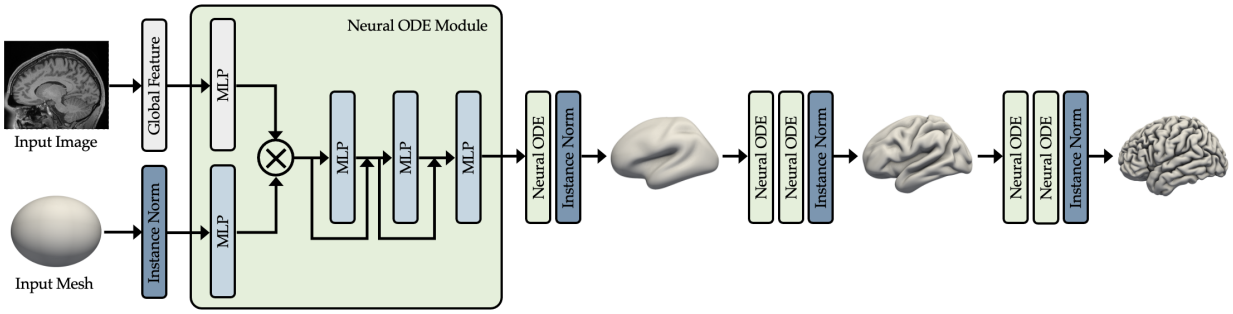


Figure 2.6: The architecture of the Neural Mesh Flow (Gupta and Chandraker, 2020). An input mesh is deformed into a target surface via several Neural ODE modules, based on both vertex coordinates and global feature vectors learned from an input 2D/3D image.

Deep Signed Distance Function. Different from explicit mesh deformation approaches (Wang et al., 2018; Wen et al., 2019; Wickramasinghe et al., 2020), Deep Signed Distance Function (DeepSDF) (Park et al., 2019) learns a SDF parameterised by a deep neural network f_θ to represent an implicit surface. For any 3D point $x \in \mathbb{R}^3$, an implicit surface

is defined as $f_\theta(x) = 0$, where $f_\theta(x) < 0$ for the interior points x of the implicit surface and $f_\theta(x) > 0$ for the exterior points. The surface representation of DeepSDF is enriched by an input global latent vector z as a conditioning variable. Such a latent vector can be learned from input point clouds or images depending on specific tasks. Figure 2.7 shows that DeepSDF predicts the SDF value $f_\theta(x, z)$ from the input location x and a latent variable z using a series of MLP layers. The predicted SDF can be rendered and visualised directly by raycasting, or be reconstructed to an explicit 3D mesh by the Marching Cubes algorithm (Lorensen and Cline, 1998), a well-known isosurface extraction method.

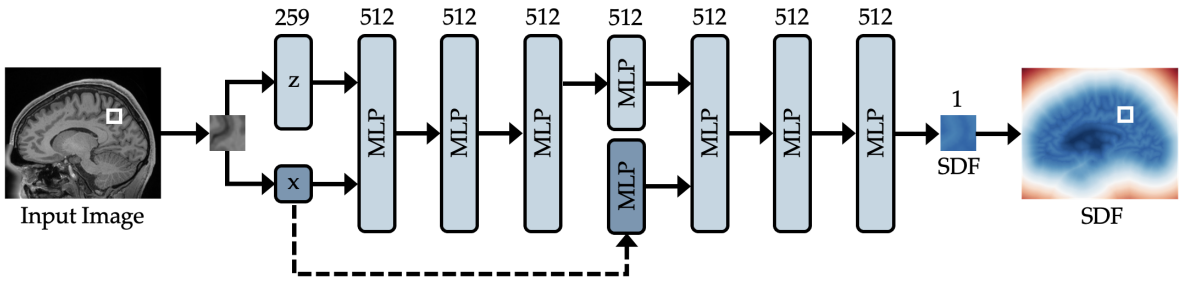


Figure 2.7: The architecture of the DeepSDF framework (Park et al., 2019). The signed distance value at any position x is learned by several MLP layers from the 3D coordinate and corresponding image intensity at the given position.

Neural Radiance Field. Neural Radiance Field (NeRF) (Mildenhall et al., 2021) and its variants (Park et al., 2021; Yu et al., 2021a; Pumarola et al., 2021) have received a wide range of attention in recent years. Rather than learning a 3D shape representation, NeRF (Mildenhall et al., 2021) leverages deep neural networks to model real-world 3D scenes with complicated geometry, colour and illumination. Given a set of multi-view 2D images of a target scene, NeRF learns a neural representation f_θ to predict the RGB colour $c \in \mathbb{R}^3$ and volume density $\sigma \in \mathbb{R}$ for any input position $x \in \mathbb{R}^3$ and viewing direction $d \in \mathbb{R}^3$, *i.e.*, $[c, \sigma] = f_\theta(x, d)$. As depicted in Figure 2.8, similar to DeepSDF (Park et al., 2019), NeRF uses multiple MLP layers to predict the volume density σ from the positional encoding $p(x)$ of the input coordinate x. Then, the RGB colours are predicted from the extracted features and the positional encoding $p(d)$ of the input direction d. Based on predicted colours and densities, a volume rendering technique is adopted to enable the visualisation of synthetic target 3D scenes at different viewing directions.

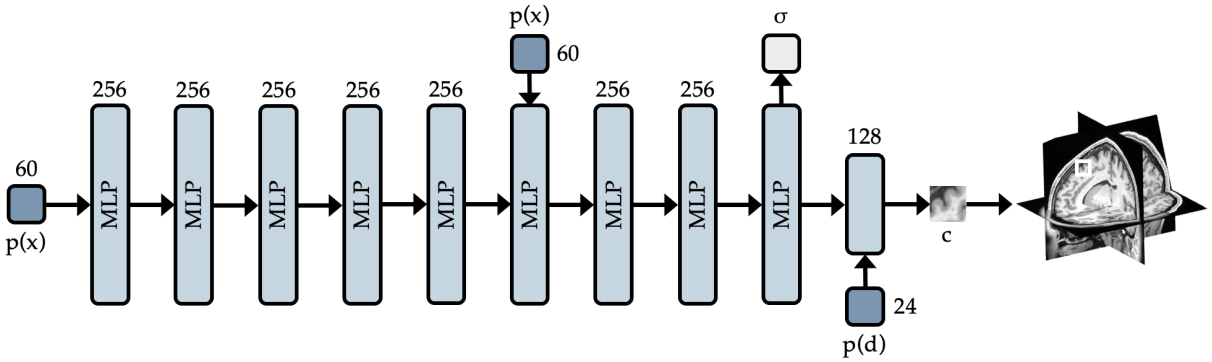


Figure 2.8: The architecture of the NeRF framework (Mildenhall et al., 2021). Given any position x, its volume density σ and colour c are predicted based on the positional encoding $p(x)$ and $p(d)$ of the input position x and view direction d.

2.2.4 Deep Learning for Surface Reconstruction

One primary application of geometric deep learning is 3D shape modelling and analysis. In this thesis, we are interested in 3D surface reconstruction and pay attention to non-Euclidean domains including explicit meshes and implicit surfaces. In computer vision, a large amount of deep learning-based approaches have been proposed to reconstruct 3D meshes from input point clouds, single- or multi-view 2D images (Groueix et al., 2018; Wang et al., 2018; Mescheder et al., 2019; Park et al., 2019; Wickramasinghe et al., 2020; Gupta and Chandraker, 2020; Mildenhall et al., 2021; Peng et al., 2021; Vahdat et al., 2022). Surface reconstruction also has broad applications in medical image analysis, *e.g.*, the 3D reconstructions of teeth models (Wu et al., 2016; Liang et al., 2020), endoscopic scenes (Bergen and Wittenberg, 2014; Mahmoud et al., 2018; Zha et al., 2023), cardiac chambers (Kong et al., 2021; Kong and Shadden, 2022; Meng et al., 2023), and cortical surfaces (Fischl, 2012; Cruz et al., 2021; Henschel et al., 2020). The extraction of surface meshes from medical images enables both 3D visualisation and quantitative analysis of corresponding anatomical structures. Generally, deep learning-based 3D surface reconstruction approaches can be divided into implicit (Mescheder et al., 2019; Park et al., 2019; Peng et al., 2020; Jiang et al., 2020; Niemeyer et al., 2020; Sitzmann et al., 2020; Remelli et al., 2020; Mildenhall et al., 2021; Mittal et al., 2022; Peng et al., 2021; Vahdat et al., 2022) and explicit

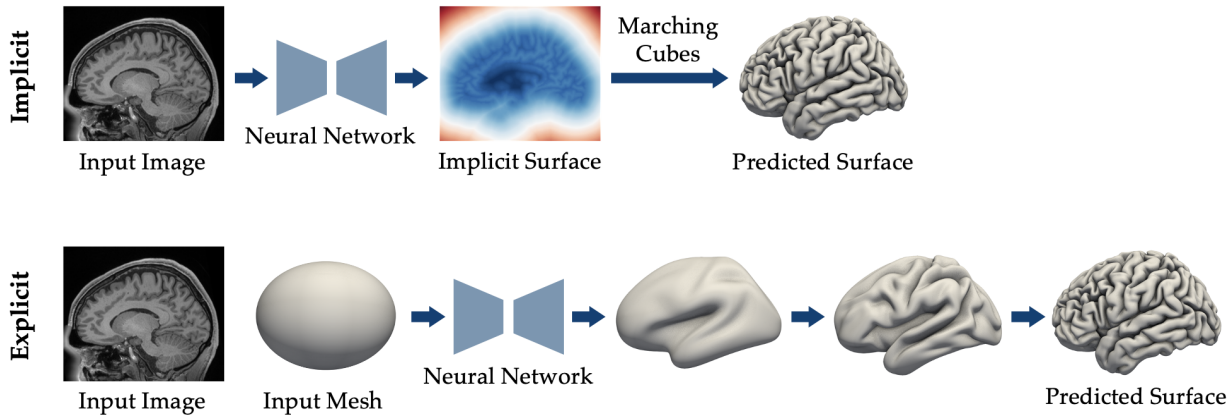


Figure 2.9: Illustration of implicit and explicit deep learning-based surface reconstruction approaches. A deep neural network model is trained to either predict an implicit surface or deform an input template mesh explicitly.

approaches (Groueix et al., 2018; Wang et al., 2018; Wen et al., 2019; Wickramasinghe et al., 2020; Gupta and Chandraker, 2020; Tóthová et al., 2020; Hui et al., 2022; Sun et al., 2022).

Implicit Surface Reconstruction. Implicit deep learning-based surface reconstruction approaches represent a surface implicitly by a binary segmentation (Henschel et al., 2020), an occupancy field (Mescheder et al., 2019; Peng et al., 2020), or a SDF (Malladi et al., 1995; Curless and Levoy, 1996; Park et al., 2019; Remelli et al., 2020; Cruz et al., 2021; Mittal et al., 2022). As illustrated in Figure 2.9, for any 3D coordinate $x \in \mathbb{R}^3$, a deep neural network model f_θ learns to predict a corresponding implicit value $y = f_\theta(x, \mathcal{I})$ conditioned on input shape information $\mathcal{I}$ such as a point cloud or a 2D/3D image. The output is a binary label $y \in \{0, 1\}$ or a signed distance $y \in \mathbb{R}$ to the implicit surface. The neural network f_θ is trained by minimising a binary cross entropy (BCE) loss between the prediction and GT for binary segmentation, or by optimising a L1/L2 loss for SDF prediction. For the inference, the neural network model f_θ predicts the value of each voxel in a 3D grid at arbitrary resolution. Afterwards, an explicit 3D polygon mesh can be extracted from the volumetric representation of the implicit surface by Marching Cubes algorithm (Lorensen and Cline, 1998). Implicit surface reconstruction does not enforce the topology of the extracted surface, and thus it is an ideal choice to model real-world scenes and objects with various topological structures.

Explicit Surface Reconstruction. Instead of learning implicit surface representations, explicit learning-based approaches (Groueix et al., 2018; Wang et al., 2018; Wen et al., 2019; Wickramasinghe et al., 2020; Gupta and Chandraker, 2020; Tóthová et al., 2020; Hui et al., 2022; Sun et al., 2022) learn from input shape information to deform an initial template mesh explicitly to the target surface. More specifically, as shown in Figure 2.9, explicit surface reconstruction approaches require an initial template mesh $\mathcal{M}_0 = \{\mathcal{V}_0, \mathcal{E}_0, \mathcal{F}_0\}$ such as a sphere, an ellipsoid, a convex hull, or surface atlases for medical images, as well as underlying shape information $\mathcal{I}$ such as a point cloud and 2D/3D images as the inputs. For each mesh vertex $x_0 \in \mathcal{V}_0$, a sequence of deep neural network models f_θ^i are employed to learn vertex displacements Δx^i from input underlying shape representation $\mathcal{I}$ and vertex information, *e.g.*, the coordinates, normals, and connectivity. Each layer of neural network models can be a GNN to learn vertex-wise displacement, a CNN to learn a deformation field, or a combination of various deep neural network modules (Wang et al., 2018; Wickramasinghe et al., 2020; Gupta and Chandraker, 2020; Tóthová et al., 2020). The final prediction of explicit learning-based approaches is a deformed surface mesh $\widehat{\mathcal{M}}$ with vertices $\hat{x} = x_0 + \sum_i \Delta x^i$.

The neural network models f_θ are trained by minimising the distance between the predicted mesh $\widehat{\mathcal{M}}$ and the GT mesh $\mathcal{M}_*$. One of widely used distances in explicit learning-based surface reconstruction is the bidirectional Chamfer distance (Fan et al., 2017; Wang et al., 2018; Wickramasinghe et al., 2020; Ravi et al., 2020), which is defined as the average distance between two point clouds $\widehat{\mathcal{P}}$ and $\mathcal{P}_*$:

$$\mathcal{L}_{cd}(\widehat{\mathcal{P}}, \mathcal{P}_*) = \frac{1}{|\widehat{\mathcal{P}}|} \sum_{\hat{p} \in \widehat{\mathcal{P}}} \min_{p_* \in \mathcal{P}_*} \|\hat{p} - p_*\|^2 + \frac{1}{|\mathcal{P}_*|} \sum_{p_* \in \mathcal{P}_*} \min_{\hat{p} \in \widehat{\mathcal{P}}} \|p_* - \hat{p}\|^2, \tag{2.21}$$

where $\widehat{\mathcal{P}}$ and $\mathcal{P}_*$ are the sets of mesh vertices or randomly sampled point clouds from the predicted and GT meshes. $|\widehat{\mathcal{P}}|$ and $|\mathcal{P}_*|$ are the numbers of points. Additionally, mesh regularisation terms such as edge length loss, mesh Laplacian loss, and normal consistency loss, are needed to enhance the quality of the predicted mesh $\widehat{\mathcal{M}}$ (Wang et al.,

2018; Wickramasinghe et al., 2020; Ravi et al., 2020). The edge length loss regularises the length of mesh edges:

$$\mathcal{L}_{edge}(\widehat{\mathcal{M}}) = \frac{1}{|\widehat{\mathcal{E}}|} \sum_{(i,j)\in\widehat{\mathcal{E}}} \|\hat{x}_i - \hat{x}_j\|^2, \tag{2.22}$$

which avoids over-length mesh edges. To encourage the mesh smoothness, the mesh Laplacian loss (Desbrun et al., 1999; Nealen et al., 2006) guides a vertex to move towards the centroid of its neighbourhoods:

$$\mathcal{L}_{lap}(\widehat{\mathcal{M}}) = \frac{1}{|\widehat{\mathcal{V}}|} \sum_{i=1}^{|\widehat{\mathcal{V}}|} \left\| \hat{x}_i - \frac{1}{|\mathcal{N}(i)|} \sum_{j\in\mathcal{N}(i)} \hat{x}_j \right\|^2, \tag{2.23}$$

where $\mathcal{N}(i)$ is the adjacency list of the i-th vertex. The normal consistency loss constraints the similarity between the normal directions of any pairs of adjacent faces:

$$\mathcal{L}_{nc}(\widehat{\mathcal{M}}) = 1 - \frac{1}{|\widehat{\mathcal{E}}|} \sum_{\hat{f}_i \cap \hat{f}_j \in \widehat{\mathcal{E}}} \hat{n}_i \cdot \hat{n}_j, \tag{2.24}$$

where two faces $\hat{f}_i, \hat{f}_i$ are adjacent and intersect at an edge. $\hat{n}_i, \hat{n}_j$ are the unit normal vectors of the faces.

Explicit surface reconstruction is topology-preserving. The topology of the reconstructed surface is determined by the input template surface. Hence, it is well-suited for surface reconstruction with known topological prior, for example, in medical image analysis (Wickramasinghe et al., 2020; Tóthová et al., 2020; Kong et al., 2021).

2.3 Cortical Surface Reconstruction

2.3.1 Problem Formulation

Cortical surfaces refer to the inner (white matter) and outer (pial) surfaces of the cerebral cortex as shown in Figure 1.1. The white matter surface is the interface between the cortical grey matter (cGM) and white matter (WM), and the pial surface is the interface between cGM and cerebrospinal fluid (CSF). The aim of cortical surface reconstruction is to extract 3D polygon meshes, in particular triangular meshes of inner and outer cortical surfaces from a 3D structural brain magnetic resonance imaging (MRI) volume. Cortical surface reconstruction is primarily performed on T1-weighted (T1w) brain MRI for adults and T2w MRI for fetal and neonatal subjects, due to the inverted MRI contrast caused by the incomplete myelination process (Prastawa et al., 2005) as illustrated in Figure 2.10.

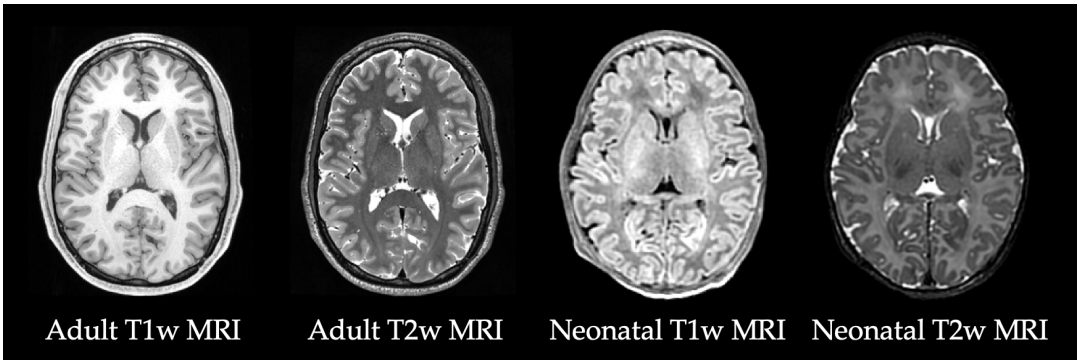


Figure 2.10: Comparison between adult and neonatal T1w and T2w brain MRI.

In addition to accurately depict the geometric structure of the cerebral cortex, cortical surfaces are required to be topologically equivalent to a sphere, *i.e.*, both inner and outer cortical surfaces are homeomorphic to a 2-sphere (Dale et al., 1999; Fischl et al., 1999a, 2001). More specifically, the cortical surfaces should be closed surfaces with genus-0 topology, *i.e.*, no holes and handles. The homeomorphism also implies that there are no singularities and crosses in cortical surfaces (Edelsbrunner and Harer, 2010), especially for pial surfaces on which two adjacent banks of a cortical sulcus are close and prone to intersect with each other. Numerically, a discretised polygon mesh $\mathcal{M} = \{\mathcal{V}, \mathcal{E}, \mathcal{F}\}$ of a cortical

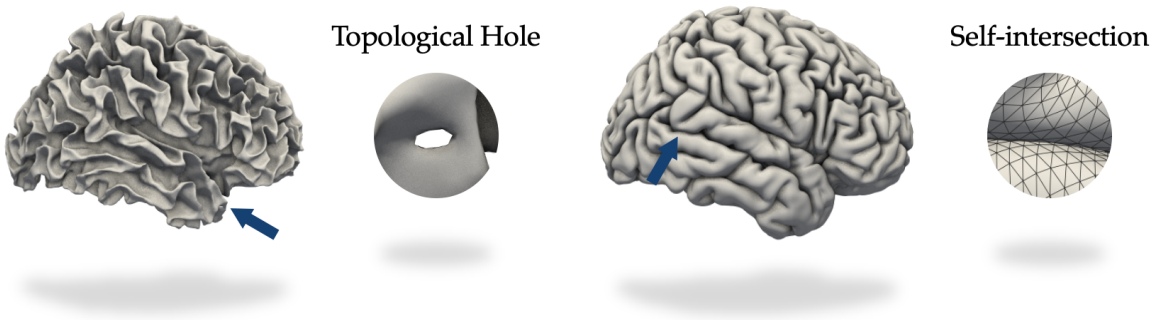


Figure 2.11: Visualisation of common topological defects. Left: a topology hole. Right: mesh self-intersections.

surface should be watertight with Euler characteristic number $\chi = |\mathcal{V}| - |\mathcal{E}| + |\mathcal{F}| = 2$. The mesh $\mathcal{M}$ should be a manifold mesh without any self-intersecting faces (Crane, 2018; Bronstein et al., 2017). The topological requirements are summarised in Table 2.1 and common topological defects are visualised in Figure 2.11.

Cortical Surface	**Polygon Mesh**
Closed surface	Watertight mesh
Genus $g = 0$	Euler characteristic $\chi = 2$
No singularities and crosses	No self-intersections

Table 2.1: Topological requirements of cortical surfaces and corresponding meshes.

2.3.2 Methodology

In this section, we introduce the fundamental methodology of cortical surface reconstruction (Davatzikos and Bryan, 1996; Dale et al., 1999; Xu et al., 1999; MacDonald et al., 2000; Shattuck and Leahy, 2002; Han et al., 2004; Kim et al., 2005; Xue et al., 2007; Liu et al., 2008; del Fresno et al., 2009; Fischl, 2012; Dai et al., 2013; Kim et al., 2015; Huo et al., 2016; Schuh et al., 2017; Makropoulos et al., 2018; Zöllei et al., 2020), as illustrated in Figure 2.12. Conventionally, cortical surface reconstruction requires both brain MRI intensity and tissue segmentation. One primary approach for cortical surface reconstruction is deformable surface model (Davatzikos and Bryan, 1996; Dale et al., 1999; Xu et al., 1999; MacDonald et al., 2000; Kim et al., 2005; Liu et al., 2008; Schuh et al., 2017).

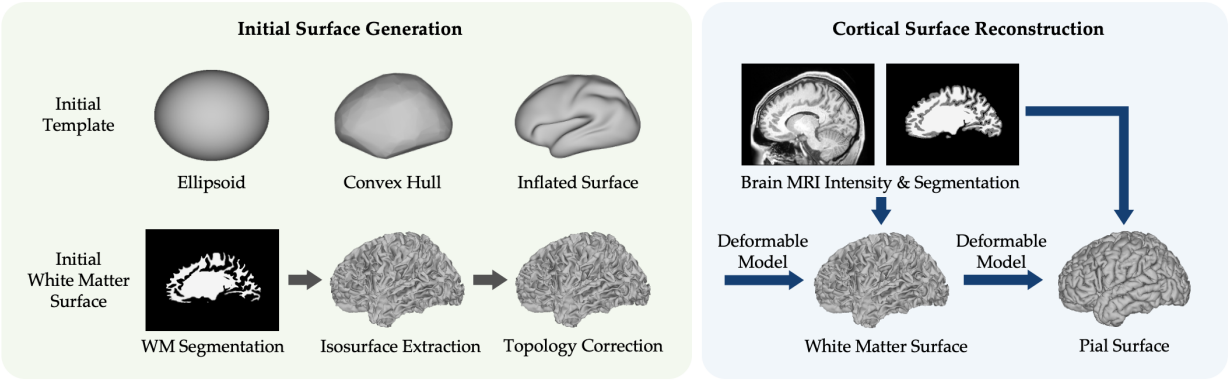


Figure 2.12: Illustration of cortical surface reconstruction procedure. An initial surface is generated, and then cortical white matter and pial surfaces are reconstructed via deformable surface models based on brain MRI intensity and segmentation.

Deformable Surface Model. Deformable model was originally devised for image segmentation and surface evolution (Kass et al., 1988; Terzopoulos and Fleischer, 1988; McInerney and Terzopoulos, 1996; Caselles et al., 1997; Xu and Prince, 1998). It iteratively deforms a surface or propagates a level set to delineate the boundary and cover the target region. For cortical surface reconstruction, a deformable surface model intends to deform an initial surface, *e.g.*, a convex hull or an inflated surface, into cortical white matter and pial surfaces. The surface deformations are guided by a combination of multiple forces including external and internal forces (Davatzikos and Bryan, 1996; Dale et al., 1999; Xu et al., 1999; MacDonald et al., 2000; Kim et al., 2005; Schuh et al., 2017). When cortical segmentations are available with known WM/cGM and cGM/CSF boundaries, the external force can be defined as a boundary distance force, which deforms the mesh vertices towards the tissue boundary along the normal direction, while the displacement is determined by the distance from the vertices to the boundary (Davatzikos and Bryan, 1996; Xu et al., 1999; MacDonald et al., 2000; Kim et al., 2005; Schuh et al., 2017). An intensity-based force is commonly used as an external force to refine the cortical surface, encouraging the surface to fit the tissue boundary with maximal image contrast or intensity gradient of the brain MRI (Dale et al., 1999; Xu et al., 1999; Schuh et al., 2017). The internal forces are regularisation terms such as a surface smoothness force to constrain the mesh smoothness, or a self-proximity force to reduce self-collision and intersection.

The surface deformations are computed by integrating a Lagrange's equation of motion using forward Euler discretisation (Schuh et al., 2017), or alternatively minimising an objective energy function via gradient descent (Dale et al., 1999; Xu et al., 1999; MacDonald et al., 2000; Kim et al., 2005). In each step of integration or optimisation, self-intersecting faces of the cortical surface mesh are detected and resolved to prevent surface self-intersections (Möller, 1997; Dale et al., 1999; Park et al., 2001; Schuh et al., 2017). Therefore, the deformable surface model preserves the topology of reconstructed cortical surfaces. In addition to constructing a deformable surface model comprising multiple external and internal forces, classic non-rigid point matching approaches in computer vision and graphics, such as non-rigid iterative closest point (ICP) algorithms (Arun et al., 1987; Besl and McKay, 1992; Zhang, 1994; Chui and Rangarajan, 2000; Liang et al., 2018; Zhang et al., 2021), can be adapted to tackle cortical surface reconstruction problems as well. Such approaches iteratively find the optimal non-rigid correspondence and alignment between the mesh vertices of the initial surface and the boundary points defined by cortical segmentations or level sets.

White Matter Surface Reconstruction. Given both brain MRI intensity and segmentation, a routine of cortical surface reconstruction is to first reconstruct a white matter surface and then inflate it into a pial surface (Dale et al., 1999), since the cGM labels are more susceptible to partial volume effects (Ballester et al., 2002; Van Leemput et al., 2003; Tohka et al., 2004) compared to WM segmentations, resulting in an increased number of topological errors. To extract a white matter surface from brain MRI, an initial surface is indispensable for applying a deformable surface model. As shown in Figure 2.12, there are primarily two types of initial surfaces. The first type is a smooth surface template such as an ellipsoid and a convex hull, which are adopted in CIVET (MacDonald et al., 2000; Kim et al., 2005) and the dHCP pipeline (Schuh et al., 2017; Makropoulos et al., 2018) respectively. The initial surface template is deformed into a white matter surface explicitly without changing the predefined surface topology. Nevertheless, owing to complicated geometric appearance of cortical surfaces, the deformable surface model inevitably re-

quires considerable deformations to evolve a smooth template to white matter surfaces. The optimisation procedure can be stuck in local minima, leading to suboptimal results.

Apart from using a smooth template, existing cortical surface reconstruction methods (Dale et al., 1999; Xu et al., 1999; Shattuck and Leahy, 2002; Han et al., 2004; Fischl, 2012; Huo et al., 2016) tend to extract an initial white matter surface from the WM segmentation via surface tessellation or isosurface extraction (Dale et al., 1999; Lorensen and Cline, 1998). However, since the isosurface extraction does enforce the topology of extracted surfaces, topology correction algorithms (Fischl et al., 2001; Shattuck and Leahy, 2001; Han et al., 2002; Bazin and Pham, 2005, 2007; Ségonne et al., 2007) are demanded to guarantee the initial surfaces are topologically equivalent to a 2-sphere.

- *Isosurface Extraction.* Surface tessellation and isosurface extraction extract an explicit surface mesh directly from a segmentation (Dale et al., 1999; Lorensen and Cline, 1998). Given a WM segmentation, surface tessellation approaches tessellate triangles on outer voxels of the segmentation, which produces a triangular mesh representation as an initial cortical white matter surface. Similarly, isosurface extraction approaches, such as the Marching Cube algorithm (Lorensen and Cline, 1998), extract a surface mesh from a volumetric level set or a segmentation. Since the WM segmentation is a discrete volume, the extracted initial white matter surface is further smoothed to improve the mesh quality (Dale and Sereno, 1993; Taubin, 1995b; Desbrun et al., 1999).

- *Topology Correction.* Although surface tessellation and the Marching Cubes algorithm extract initial white matter surfaces rapidly from WM segmentations, the extracted initial surfaces are not guaranteed to be topologically equivalent to a sphere with genus-0 topology, *i.e.*, without any holes and handles. Therefore, topology correction algorithms (Fischl et al., 2001; Shattuck and Leahy, 2001; Han et al., 2002; Bazin and Pham, 2005, 2007; Ségonne et al., 2007) are required to fix the topological defects of the initial white matter surfaces by filling the holes and cutting the handles. Topology correction can be performed on either a surface mesh or a segmentation. Surface-based topol-

ogy correction (Fischl et al., 2001; Ségonne et al., 2007) has been extensively integrated into automated brain MRI processing pipelines such as FreeSurfer (Dale et al., 1999; Fischl et al., 1999a; Fischl, 2012). It projects a cortical surface onto a sphere to identify topological defects, which are eliminated by manipulating the surface mesh directly. For segmentation-based topology correction, Bazin and Pham (2005, 2007) developed a fast marching algorithm, which iteratively finds the critical voxels and corrects the topological errors in the segmentation. Besides, Shattuck and Leahy (2001) and Han et al. (2002) used a graph to represent different components of a binary segmentation. The topology correction is accomplished by modifying the voxels in the segmentation, such that the graph does not contain any circles. Although topology correction algorithms are effective in removing topological errors without significantly changing the anatomical structure of cortical surfaces, they are usually time-consuming depending on the number and severity of topology defects.

As depicted in Figure 2.12, a preliminary white matter surface with spherical topology is obtained after surface tessellation, topology correction, and subsequent mesh smoothing. In order to further refine the initial white matter surface and eliminate potential errors introduced by imprecise segmentations, a deformable surface model with intensity-based terms is employed such that the white matter surface fits the WM/cGM boundary in brain MRI with high image contrast (Dale et al., 1999; Xu et al., 1999; Fischl, 2012). The preliminary white matter surface provides a good initialisation for the deformable model, which avoids local minima and consequently achieves superior geometric accuracy.

Pial Surface Reconstruction. Since two adjacent banks of a sulcus are very close to each other, the topology of cGM segmentations is severely affected by partial volume effects (Ballester et al., 2002; Van Leemput et al., 2003; Tohka et al., 2004). The misclassification of a single voxel can create a handle connecting the neighbouring banks of cortical sulci, leading to large topological errors (Dale et al., 1999). Therefore, it is difficult to extract pial surfaces directly from brain MRI and cGM segmentations like the reconstruction of

white matter surfaces. Existing approaches (Dale et al., 1999; Xu et al., 1999; MacDonald et al., 2000; Shattuck and Leahy, 2002; Kim et al., 2005; Schuh et al., 2017) use another deformable surface model to iteratively extend a white matter surface into a pial surface, until it reaches the cGM/CSF boundary determined by the image contrast or gradient of brain MRI. The self-intersections of the pial surface are detected and prevented during the surface deformation (Möller, 1997; Dale et al., 1999; Park et al., 2001; Schuh et al., 2017).

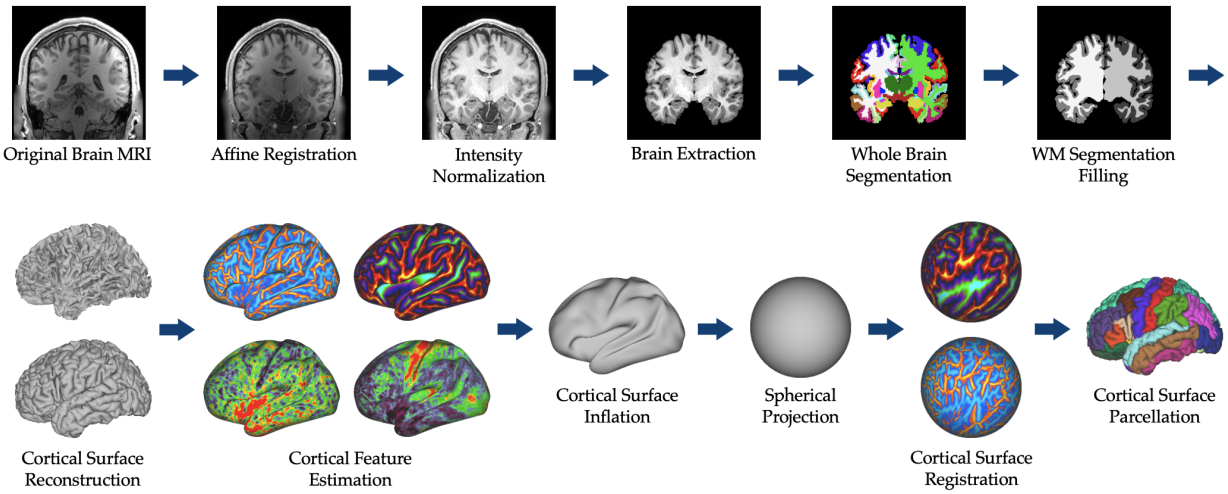


Figure 2.13: The workflow of a conventional automated neuroimage processing pipeline.

2.3.3 Neuroimage Processing Pipeline

Cortical surface reconstruction is associated with automated brain MRI processing and analysis pipelines (Dale et al., 1999; Fischl et al., 1999a; MacDonald et al., 2000; Shattuck and Leahy, 2002; Han et al., 2004; Kim et al., 2005; Fischl, 2012; Dai et al., 2013; Glasser et al., 2013; Kim et al., 2015; Huo et al., 2016; Makropoulos et al., 2018; Zöllei et al., 2020). As shown in Figure 2.13, the processing flow of existing pipelines usually consists of image preprocessing, brain segmentation, cortical surface reconstruction, inflation, registration and parcellation. In this section, we review the motivation and methodology of each processing step in a standard cortical surface-based brain MRI analysis pipeline, based on the workflow of FreeSurfer, one of the most well-known neuroimage processing software (Dale et al., 1999; Fischl et al., 1999a; Fischl, 2012).

Image Preprocessing. The preprocessing of a 3D structural brain MRI volume is a prerequisite for downstream brain segmentation and cortical surface reconstruction, which require the brain MRI to be located in a standard coordinate space with uniformly distributed image intensity and contrast (Dale et al., 1999). The preprocessing steps include affinely aligning the MRI to a template space, normalising the inhomogeneous image intensity and contrast, and extracting the region of brain tissues.

- *Affine Registration.* The objective of affine registration is to find an optimal affine transformation that aligns a brain MR image to a template image space, *e.g.*, the Talairach (Talairach, 1988; Collins et al., 1994) and the MNI space (Fonov et al., 2011). It transforms all brain MRI to the same coordinate space such that the brains have consistent scale, position and rotation across different scans and subjects. Affine registration has been implemented in medical image analysis toolkits such as Advanced Normalisation Tools (ANTs) (Avants et al., 2009), Insight Toolkit (ITK) (Avants et al., 2014), Medical Image Registration ToolKit (MIRTK) (Schuh, 2016), and NiftyReg (Modat et al., 2014).

- *Intensity Normalisation.* Intensity normalisation, also known as bias field correction, aims to correct the intensity and contrast inhomogeneities, *i.e.*, the bias fields, in a brain MR image. The imbalanced MRI intensity and contrast can cause subsequent errors in intensity-based brain segmentation and cortical surface reconstruction. The most frequently used intensity normalisation approaches include the nonparametric nonuniform normalisation (N3) algorithm (Sled et al., 1998) and its variant N4 bias correction algorithm (Tustison et al., 2010).

- *Brain Extraction.* The purpose of brain extraction, also called skull stripping, is to remove the skull from a head MRI scan and preserve the region of brain tissues. It produces a binary brain mask by deformaing a template surface mesh, for example, a sphere (Smith, 2002) or an ellipsoid (Dale et al., 1999), to fit the outer boundary of the brain. One of the frequently used skull stripping algorithms is the Brain Extraction Tool (BET) (Smith, 2002) in the FMRIB Software Library (FSL) (Jenkinson et al., 2012).

Whole Brain Segmentation. Whole brain segmentation of a brain MRI intends to classify and label different brain regions and tissue types (Pham et al., 2000; Despotović et al., 2015). Conventional brain MRI segmentation approaches incorporate intensity-based segmentation (Wells et al., 1996; Pham and Prince, 1999b; Cocosco et al., 2003), atlas-based segmentation (Collins et al., 1995; Fischl et al., 2002; Heckemann et al., 2006; Cabezas et al., 2011), and active contour models (Kass et al., 1988; McInerney and Terzopoulos, 1996; Chan and Vese, 2001; Yushkevich et al., 2006).

- *Intensity-based Segmentation.* Intensity-based segmentation assigns a label to each voxel of a greyscale brain MRI volume based on its intensity distribution and image contrast. Intensity-based segmentation approaches can be further divided into intensity thresholding (Sahoo et al., 1988; Teo et al., 1997; Pham et al., 1997), classification (Wells et al., 1996; Warfield et al., 2000; Cocosco et al., 2003) and clustering (Pham and Prince, 1999b,a; Chen and Zhang, 2004; Wang et al., 2009). However, intensity-based segmentation is sensitive to the quality of brain MRI and prone to be affected by noises, intensity inhomogeneity, and partial volume effects, *i.e.*, a voxel may contain multiple types of tissues (Ballester et al., 2002). Moreover, it is difficult for intensity-based approaches to delineate different subcortical structures with similar image intensity.

- *Atlas-based Segmentation.* Atlas-based segmentation performs deformable non-rigid registration from the intensity of a single or multiple brain MRI atlases to an individual brain MR image (Collins et al., 1995; Fischl et al., 2002; Heckemann et al., 2006; Pohl et al., 2006; Han and Fischl, 2007; Bazin and Pham, 2008; Aljabar et al., 2009; Lötjönen et al., 2010; Cabezas et al., 2011). The template segmentation of the atlas is then propagated and transformed accordingly to create individual brain segmentation maps. Such an atlas-based segmentation is less susceptible to intensity shifts, and the topology of the segmentation is preserved during the registration. However, atlas-based segmentation needs substantial deformations due to the large individual variations in the anatomical structures of brains, which may result in imprecise segmentation.

- *Active Contour Model.* Active contour model (McInerney and Terzopoulos, 1996), also known as deformable model as described in Section 2.3.2, is composed of edge-based (Kass et al., 1988; Caselles et al., 1997; Xu and Prince, 1998) and region-based (Chan and Vese, 2001; Vese and Chan, 2002; Li et al., 2008, 2011) models. To segment the brain MRI, an active contour model minimises an energy functional to either deform a template contour to delineate the boundary with high contrast (edge-based), or grow an initial region or level set to cover the corresponding brain tissues (region-based). Unlike atlas-based registration, active contour models are able to capture topological variations in segmentations.

To enhance the accuracy of brain MRI segmentations, existing brain MRI processing pipelines (Han et al., 2004; Ashburner, 2009; Fischl, 2012; Makropoulos et al., 2018) adopted hybrid whole brain segmentation approaches (Pham and Prince, 1999a; Dale et al., 1999; Shattuck et al., 2001; Ashburner and Friston, 2005; Fischl et al., 2002; Makropoulos et al., 2014; Huo et al., 2016), utilising and combining above-mentioned approaches followed by adaptation and improvement. Apart from traditional segmentation approaches, recent deep learning-based models, such as CNN (Moeskops et al., 2016), U-Net (Ronneberger et al., 2015), nnU-Net (Isensee et al., 2021), QuickNAT (Roy et al., 2019), and FastSurfer-CNN (Henschel et al., 2020), have achieved rapid inference and impressive accuracy on whole brain segmentation tasks.

After brain MRI segmentation, a cutting plane is estimated to split the segmentation maps into left and right hemispheres. To provide cortical segmentations for the following cortical surface reconstruction (Dale et al., 1999), the cerebellum region is removed, and the WM segmentation is filled by combining the label maps of other subcortical structures including deep grey matter (dGM) and ventricles. The connected components are also examined to remove isolated voxels and fill the holes inside the segmentations.

Cortical Surface Reconstruction. Cortical surfaces play an essential role in visualising and quantifying the anatomical structure of the cerebral cortex. Given preprocessed brain

MRI volumes and corresponding brain tissue segmentations, cortical white matter and pial surfaces can be reconstructed according to Section 2.3.2. The reconstructed cortical surfaces facilitate the estimation of cortical morphological features, such as cortical thickness, volume, surface area, sulcal depth, and curvature, as visualised in Figure 2.13. In this thesis, we present deep learning-based approaches for explicit and end-to-end cortical surface reconstruction (Ma et al., 2021, 2022, 2023, 2024, 2025).

Spherical Inflation and Projection. To fully visualise cortical features on highly folded cortical surfaces as well as represent cortical surfaces in a unified spherical coordinate system (Fischl et al., 1999a,b; Angenent et al., 1999; Tosun and Prince, 2001), it is necessary to inflate the reconstructed cortical surfaces and projected the surfaces onto a 2-sphere.

- *Cortical Surface Inflation.* The highly folded cerebral cortex makes it difficult to visualise the entire morphological or functional maps on cortical surfaces, especially in the deep cortical sulci of pial surfaces. This underscores the necessity to unfold the reconstructed cortical surfaces by iteratively inflating and smoothing the surface meshes. In order to produce a meaningful inflated surface for both visualisation and spherical projection, the metric distortions between the original cortical surface and the inflated surface should be minimised to preserve the edge length, face area, and geodesic distances. Existing brain MRI processing pipelines normally optimise an energy functional (Fischl et al., 1999a) to inflate and unfold a white matter or midthickness surface into an inflated surface with minimal metric distortions. The sulcal depth, or mean convexity/concavity of the cortical surface, is estimated during inflation by tracking and integrating the displacement along the normal direction (Fischl et al., 1999a).

- *Spherical Projection.* After surface inflation and unfolding, the inflated cortical surfaces can be further projected onto a 2-sphere. The large metric distortions introduced by the projection are reduced by optimising an energy functional (Fischl et al., 1999a; Elad et al., 2005). The spherical projection establishes mappings from all cortical surfaces to a sphere, providing a 2D parameterisation such that any cortical surfaces are repre-

sented in a unified spherical coordinate system (Fischl et al., 1999a,b). Such a spherical coordinate system allows us to resample or reparameterise the projected spheres of different cortical surfaces to a standard icosphere via barycentric interpolation (Fischl et al., 1999b; Robinson et al., 2014). After resampling, all cortical surfaces have the same mesh connectivity with vertex-to-vertex correspondence, enabling downstream surface registration (Fischl et al., 1999b; Yeo et al., 2009; Robinson et al., 2014), parcellation and atlas construction (Fischl et al., 2004; Desikan et al., 2006; Shattuck et al., 2008; Destrieux et al., 2010; Van Essen et al., 2012; Glasser et al., 2016; Bozek et al., 2018).

Surface Registration and Parcellation. Given the spherical mapping of cortical surfaces, surface registration is operated in the spherical domain to align any pairs of cortical surfaces (Fischl et al., 1999b; Yeo et al., 2008, 2009; Robinson et al., 2014, 2018). The inter- and intra-subject cortical surface registration facilitates cross-sectional (Fischl et al., 1999b; Marcus et al., 2007; Lyttelton et al., 2007; Rimol et al., 2012; Van Essen et al., 2013; Madre et al., 2020) and longitudinal (Jack Jr et al., 2008; Marcus et al., 2010; Wright et al., 2014; Garcia et al., 2018b; Somerville et al., 2018; LaMontagne et al., 2019; Edwards et al., 2022) surface-based brain MRI analysis respectively. The registration of cortical surfaces to an atlas space is also a prerequisite for cortical surface parcellation, which delineates different cortical regions based on geometric and functional properties (Fischl et al., 2004; Desikan et al., 2006; Destrieux et al., 2010; Glasser et al., 2016; Eickhoff et al., 2018).

- *Cortical Surface Registration.* Due to the highly curved and folded geometric structure, cortical surface registration is performed in the spherical space or 2D parametric space instead of the 3D Euclidean space. The cortical surfaces are projected onto spheres, which are then deformed and aligned to match cortical folding patterns described by the sulcal depth and mean curvature (Fischl et al., 1999b). Such a registration procedure finds the correspondence between two cortical surfaces without changing the anatomical geometry of original surfaces. Typical cortical surface registration frameworks include Spherical Demons (Yeo et al., 2008, 2009), which operates on spherical images,

i.e., the 2D parametric representation of spheres, and Multimodal Surface Matching (MSM) (Robinson et al., 2014, 2018), which operates on the sphere directly. It should be noted that deep learning-based approaches such as Spherical U-Net (Zhao et al., 2019a, 2021a), SphereMorph (Cheng et al., 2020), and GeoMorph (Suliman et al., 2022a,b) have also shown great potential for fast and accurate cortical surface registration.

- *Cortical Surface Parcellation.* Cortical surface parcellation is a segmentation task on the surface domain, which assigns a regional label to each mesh vertex of a cortical surface. Traditional cortical parcellation approaches (Fischl et al., 2004; Desikan et al., 2006; Destrieux et al., 2010) first perform surface registration to transform the cortical surface into an atlas space. Then, each vertex is classified based on the geometric features of the surface, *i.e.*, mean curvature and sulcal depth, as well as the prior information from a probabilistic parcellation atlas and the local spatial relationship between different regions of interest (ROI) (Fischl et al., 2004). The probabilistic atlas is constructed by aligning and averaging manually annotated parcellation maps of multiple subjects (Fischl et al., 1999b, 2004). There are several widely used cortical surface-based atlases with different parcellation protocols, for instance, the Desikan-Killiany atlas (Desikan et al., 2006), the Destrieux atlas (Fischl et al., 2004; Destrieux et al., 2010), the DKT atlas (Klein and Tourville, 2012), the Conte-69 atlas (Van Essen et al., 2012), the HCP-MMP1 atlas (Glasser et al., 2016), and the dHCP atlas (Bozek et al., 2018). Motivated by classic approaches, the advent of geometric deep learning has also provided powerful tools for cortical surface parcellation (Wu et al., 2019b; Zhao et al., 2019b, 2021b; Williams et al., 2021; Fawaz et al., 2021).

Chapter 3

Learning Explicit and Diffeomorphic Cortical Surface Reconstruction

The content of this chapter is based on the following publications:

- **Ma, Q.**, Robinson, E.C., Kainz, B., Rueckert, D. and Alansary, A. (2021). PialNN: A Fast Deep Learning Framework for Cortical Pial Surface Reconstruction. In *International Workshop on Machine Learning in Clinical Neuroimaging*, pages 73-81. Springer. https://doi.org/10.1007/978-3-030-87586-2_8

- **Ma, Q.**, Li, L., Robinson, E.C., Kainz, B., Rueckert, D. and Alansary, A. (2022). CortexODE: Learning Cortical Surface Reconstruction by Neural ODEs. *IEEE Transactions on Medical Imaging*, 42(2):430-443. https://doi.org/10.1109/TMI.2022.3206221

Traditional cortical surface reconstruction pipelines have made significant contributions to neuroimaging studies, while these pipelines are usually time-consuming and reconstruct cortical surfaces relying on implicit brain segmentations. Inspired by recent progress in geometric deep learning (Kipf and Welling, 2016; Wang et al., 2018; Wickramasinghe et al., 2020), in this chapter, we present an explicit 3D geometric deep neural network model for fast and end-to-end cortical pial surface reconstruction from brain magnetic resonance imaging (MRI) data. To further prevent self-intersections of cortical surfaces, we leverage Neural Ordinary Differential Equations (ODEs) (Chen et al., 2018; Gupta and Chandraker, 2020) to learn diffeomorphic surface deformations for the reconstruction of both cortical white matter and pial surfaces. Such an approach is evaluated on large-scale

brain MRI datasets in various age groups including neonates (25-45 weeks), young adults (22-36 years) and elderly subjects (55-90 years). Our approaches achieve state-of-the-art geometric accuracy while being orders of magnitude faster compared to traditional neuroimage processing pipelines (Fischl, 2012; Glasser et al., 2013; Makropoulos et al., 2018).

3.1 Learning Explicit Cortical Pial Surface Reconstruction

3.1.1 Introduction

Cortical surface reconstruction aims to visualise and analyse the 3D structure of the cerebral cortex by extracting its inner and outer surfaces from brain MRI scans (Dale et al., 1999; Fischl et al., 1999a). The inner cortical surface (or white matter surface) lies between the cortical grey matter (cGM) and white matter (WM) of the brain, and the outer surface (or pial surface) lies between the cerebrospinal fluid (CSF) and the cGM. It is difficult to extract a geometrically accurate and topologically correct cortical surface due to its highly curved and folded geometric shape. Typical cortical surface reconstruction approaches (Dale et al., 1999; Xu et al., 1999), which are integrated into existing neuroimage processing pipelines (Shattuck and Leahy, 2002; Han et al., 2004; Fischl, 2012; Glasser et al., 2013), consists of two main steps. Firstly, an initial white matter surface mesh is created by applying surface tessellation or Marching Cubes algorithm (Lorensen and Cline, 1998) to the WM segmentation mask of the brain MRI scan, along with topology fixing to guarantee the spherical topology (Fischl et al., 2001; Ségonne et al., 2007). The initial mesh is further refined and smoothed to produce the final white matter surface (Dale and Sereno, 1993). Secondly, the pial surface is produced by expanding the white matter surface iteratively until it reaches the boundary between the cGM and CSF, or causes self-intersections. One limitation of such approaches is the high computational cost. These widely used brain MRI analysis tools (Han et al., 2004; Fischl, 2012; Glasser et al., 2013) usually require several hours to extract cortical surfaces from the brain MRI scan of a single subject.

As a fast and end-to-end alternative approach, deep learning has shown its advantages in surface reconstruction for general shape objects (Chang et al., 2015; Groueix et al., 2018; Wang et al., 2018; Wen et al., 2019; Mescheder et al., 2019; Park et al., 2019) and medical images (Henschel et al., 2020; Tóthová et al., 2020; Wickramasinghe et al., 2020; Cruz et al., 2021). Given brain MRI scans, existing deep learning-based frameworks (Henschel et al.,

2020; Cruz et al., 2021) are able to predict cortical surfaces within 30 minutes, which is much faster than traditional neuroimage processing pipelines (Han et al., 2004; Fischl, 2012; Glasser et al., 2013). However, although the white matter surfaces can be extracted accurately, the reconstruction of pial surfaces is still challenging. Due to the highly folded and curved geometry, the pial surfaces predicted by previous deep learning approaches tend to be oversmooth to reduce self-intersections (Wickramasinghe et al., 2020), or fail to reconstruct the deep and narrow sulcal regions (Cruz et al., 2021).

In Section 3.1, we propose a fast and accurate deep neural network (DNN) architecture for pial surface reconstruction, called Pial Neural Network (PialNN). Given an input white matter surface and its corresponding brain MR image, PialNN reconstructs the pial surface mesh using a sequence of learnable deformation blocks. In each deformation block, we introduce a local convolutional operation, which applies a 3D convolutional neural network (CNN) to a small cube containing the MRI intensity at the locations of a vertex and its neighbourhood. Our method can process brain MRI scans at arbitrary resolution without increasing the complexity. PialNN establishes a one-to-one correspondence between the vertices in white matter and pial surfaces such that a point-to-point loss can be minimised directly without any regularisation terms or point matching. The performance is evaluated on the publicly available Human Connectome Project (HCP) young adult dataset (Van Essen et al., 2013). PialNN shows superior geometric accuracy compared to existing deep learning-based surface reconstruction approaches (Wickramasinghe et al., 2020; Cruz et al., 2021). Our codes are publicly available at `https://github.com/m-qiang/PialNN`. The main contributions and advantages of the PialNN can be summarised as:

- **Fast:** PialNN can be trained end-to-end to reconstruct the 3D mesh of cortical pial surface within one second.

- **Memory-efficient:** The local convolutional operation enables PialNN to process a high resolution mesh (>150k vertices) based on input brain MRI at arbitrary resolution.

- **Accurate:** The point-to-point loss, without additional vertex matching or mesh regularisation, effectively improves the geometric accuracy of the reconstructed surfaces.

3.1.2 Related Work

Deep learning-based 3D surface reconstruction approaches can be divided into implicit (Mescheder et al., 2019; Park et al., 2019) and explicit methods (Groueix et al., 2018; Wang et al., 2018; Wen et al., 2019; Gkioxari et al., 2019; Wickramasinghe et al., 2020). The former use a DNN to learn an implicit surface representation such as an occupancy field (Mescheder et al., 2019) and a signed distance function (Park et al., 2019). A triangular mesh is then extracted using isosurface extraction method such as the Marching Cubes (Lorensen and Cline, 1998). For explicit methods, a DNN is trained end-to-end to deform an initial mesh into a target surface, producing an explicit mesh directly.

Previous deep learning-based frameworks (Henschel et al., 2020; Cruz et al., 2021) for cortical surface reconstruction mainly adopted implicit methods. Henschel et al. (2020) proposed a FastSurfer pipeline, which improved FreeSurfer (Fischl, 2012) by introducing a fast CNN for whole brain segmentation instead of atlas-based approaches. The cortical surface is then extracted similarly to FreeSurfer. Cruz et al. (2021) proposed DeepCSR framework to predict the implicit representation of both inner and outer cortical surfaces. Explicit surfaces are extracted by the Marching Cubes algorithm (Lorensen and Cline, 1998). While implicit methods require time-consuming topology correction (Fischl et al., 2001; Bazin and Pham, 2005, 2007; Ségonne et al., 2007), explicit methods can pre-define an initial mesh with spherical topology to achieve fast inference. Wickramasinghe et al. (2020) presented an explicit learning-based framework, called Voxel2Mesh, to extract 3D meshes from medical images. Voxel2Mesh incorporated 3D U-Nets (Ronneberger et al., 2015) for feature extraction and graph convolutional networks (Kipf and Welling, 2016) for surface deformation. It employed a series of deformation and unpooling layers to deform an initial mesh while increasing the number of vertices iteratively. Regularisation

terms are utilised to improve the mesh quality and reduce self-intersections, whereas these terms tend to oversmooth the output mesh. Conversely, our PialNN uses explicit methods to learn the pial surface reconstruction without any regularisation terms.

3.1.3 Method

We first introduce necessary notations to formulate the problem. Let $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ be a 3D triangular mesh, where $\mathcal{V} \subset \mathbb{R}^3$, $\mathcal{E}$ and $\mathcal{F}$ are the sets of vertices, edges and faces of the mesh. The corresponding coordinates and normal of the vertices are represented by $\mathbf{x}, \mathbf{n} \in \mathbb{R}^{|\mathcal{V}|\times 3}$, where $|\mathcal{V}|$ is the number of vertices. Given an initial white matter surface $\mathcal{M}_0 = (\mathcal{V}_0, \mathcal{E}_0, \mathcal{F}_0)$ and a target pial surface $\mathcal{M}_* = (\mathcal{V}_*, \mathcal{E}_*, \mathcal{F}_*)$, we assume that $\mathcal{M}_0$ and $\mathcal{M}_*$ have the same connectivity, *i.e.* $\mathcal{E}_0 = \mathcal{E}_*$ and $\mathcal{F}_0 = \mathcal{F}_*$.

Given a brain MRI volume $\mathbf{I} \in \mathbb{R}^{L\times W\times H}$, the goal of deep learning-based pial surface reconstruction is to learn a neural network model g such that the vertex coordinates $\mathbf{x}_* = g(\mathbf{x}_0, \mathbf{n}_0, \mathbf{I})$. As illustrated in Figure 3.1, the PialNN framework aims to learn a series of deformation blocks f_{θ_l} for $1 \leq l \leq L$ to iteratively deform the white matter surface $\mathcal{M}_0$ to match the target pial surface $\mathcal{M}_*$, where θ_l represents the learnable parameter of the neural network.

3.1.3.1 Deformation Block

Let $\mathcal{M}_l$ be the l-th intermediate deformed mesh. The vertices of $\mathcal{M}_l$ can be computed as:

$$\mathbf{x}_l = \mathbf{x}_{l-1} + \Delta\mathbf{x}_{l-1} = \mathbf{x}_{l-1} + f_{\theta_l}(\mathbf{x}_{l-1}, \mathbf{n}_{l-1}, \mathbf{I}), \tag{3.1}$$

for $1 \leq l \leq L$, where f_{θ_l} is the l-th deformation block represented by a neural network. The purpose of PialNN is to learn the optimal f_{θ_l}, such that the final predicted mesh $\mathcal{M}_L$ matches the target mesh $\mathcal{M}_*$, *i.e.*, $\mathcal{M}_L = \mathcal{M}_*$. The architecture of the deformation block is

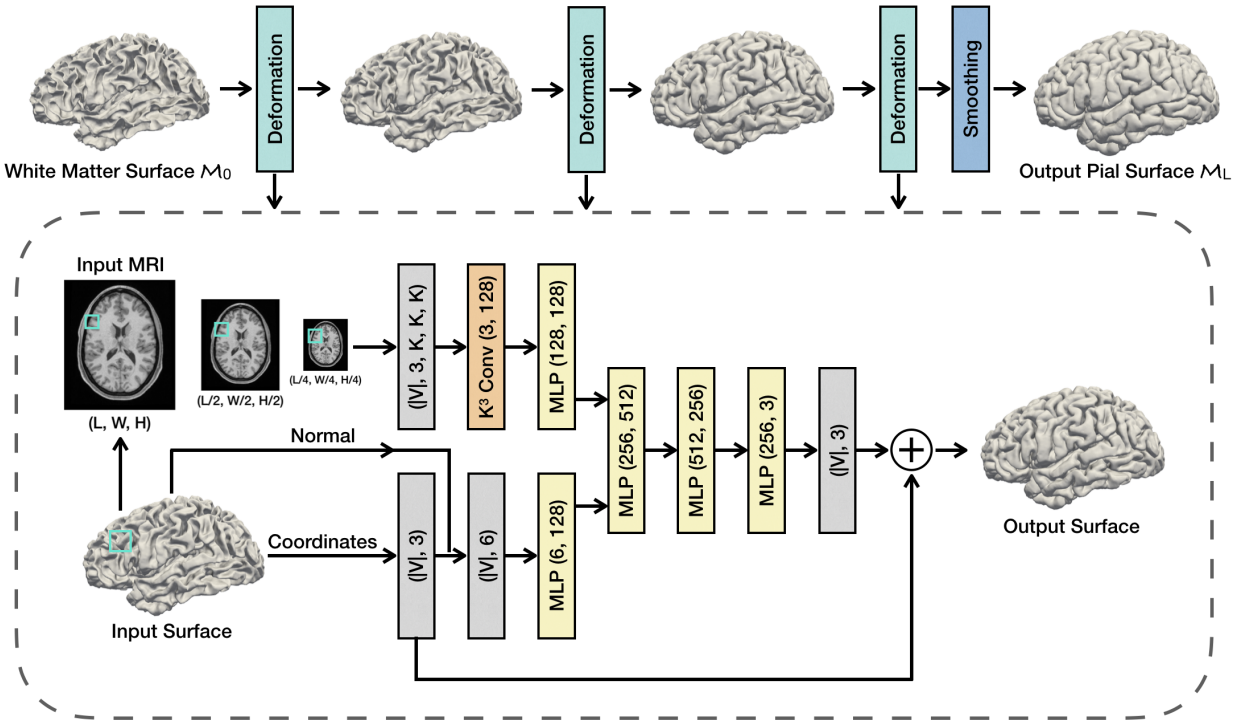


Figure 3.1: The architecture of PialNN for pial surface reconstruction. The input white matter surface is deformed by three deformation blocks to predict a pial surface. Each deformation block incorporates two types of features: point features from the white matter surface vertices and local features from the brain MRI. Finally, the output mesh is refined using Laplacian smoothing.

shown in Figure 3.1. In this approach, the deformation block predicts a displacement $\Delta\mathbf{x}$ based on the *point feature* and *local feature* of the vertex $\mathbf{x}$.

Point Feature. The point feature of a vertex is defined as the feature extracted from its coordinate $\mathbf{x}$ and normal $\mathbf{n}$, which includes the spacial and orientation information. We extract the point feature using a multi-layer perceptron (MLP).

Local Feature. We adopt a local convolutional operation to extract the local feature of a vertex from brain MRI scans. Rather than using a memory-intensive 3D CNN on the entire MRI volume (Wickramasinghe et al., 2020), this method only employs a CNN on a cube containing MRI intensity of each vertex and its neighbourhood. As illustrated in Figure 3.1, for each vertex, we find the corresponding position in the brain MRI volume. Then a K^3 grid is constructed based on the vertex to exploit its neighbourhood informa-

tion. The voxel value of each point in the grid is sampled from the MRI volume. Such a cube sampling approach extracts a K^3 voxel cube containing the MRI intensity of each vertex and its neighbourhood.

Furthermore, we build a 3D image pyramid including 3 scales $\{1, 1/2, 1/4\}$ and use cube sampling on the different scales. Therefore, each vertex is represented by a K^3 local cube with 3 channels containing multi-scale information. A 3D CNN with kernel size K is then applied to each local cube, which converts the cube to a local feature vector of its corresponding vertex. An MLP layer is followed to further refine the local feature. Such a local convolutional operation is memory- and time-efficient. As there are total $|\mathcal{V}|$ cubes with 3 channels, it only executes the convolution operators $3|\mathcal{V}|$ times, which are far less than $L \times W \times H$ times for running a 3D CNN on the full MRI. Since the complexity only relies on the number of vertices $|\mathcal{V}|$, the local convolutional operation can process MRI volumes at arbitrary resolution without increasing the complexity.

The point and local features are concatenated as the input of several MLP layers followed by leaky ReLU activation, which predict a 3D displacement $\Delta \mathbf{x}_{l-1}$. The new vertices $\mathbf{x}_l$ are updated according to Equation (3.1) and passed to the next deformation block.

3.1.3.2 Smoothing and Training

Laplacian Smoothing. After three deformation blocks, a Laplacian smoothing is used to further smooth the surface (Desbrun et al., 1999; Nealen et al., 2006). For each vertex $x^i \in \mathcal{V} \subset \mathbb{R}^3$, the smoothing is defined as $\bar{x}^i = (1-\lambda)x^i + \lambda \sum_{j \in \mathcal{N}(i)} x^j / |\mathcal{N}(i)|$, where λ controls the degree of smoothness and $\mathcal{N}(i)$ is the adjacency list of the i-th vertex. The smoothing layer is incorporated in both training and testing.

Loss Function. The bidirectional Chamfer distance (Fan et al., 2017) is commonly used as the loss function for training explicit surface reconstruction models (Wang et al., 2018; Wickramasinghe et al., 2020). It measures the distance from a vertex in one mesh to the

closest vertex in the other mesh bidirectionally. For PialNN, since the input and target mesh have the same connectivity, we can directly compute a point-to-point mean squared error (MSE) between each pair of vertices. Therefore, the loss function is defined as:

$$\mathcal{L}(\mathcal{M}_L, \mathcal{M}_*) = \mathcal{L}(\mathbf{x}_L, \mathbf{x}_*) = \frac{1}{|\mathcal{V}|}\|\mathbf{x}_L - \mathbf{x}_*\|^2. \tag{3.2}$$

Rather than computing the loss for all intermediate meshes $\mathcal{M}_l$, we only compute the loss between the final predicted pial surface $\mathcal{M}_L$ and the ground truth $\mathcal{M}_*$, because the gradient can be backpropagated to all deformation blocks f_{θ_l} for $1 \leq l \leq L$. The parameters θ_l are learned by minimising the MSE loss.

It is noted that no explicit regularisation term is required in the loss function, as the vertex will learn from the point-to-point supervision to move to a correct location. Such loss function effectively improves the geometric accuracy of the output mesh. Besides, we use an additional Laplacian smoothing after training to improve the mesh quality.

3.1.4 Experiments

Dataset. The proposed framework is evaluated using the WU-Minn Human Connectome Project (HCP) Young Adult dataset (Van Essen et al., 2013). We use 300 subjects, each of which has T1-weighted brain MRI scans with $1mm^3$ isotropic resolution. Each brain MRI is cropped to size of $192 \times 224 \times 192$. The 300 subjects are split into 200/50/50 for training/validation/testing. The input white matter surface and ground truth pial surface are generated by FreeSurfer (Fischl, 2012). The surface has approximately 150k vertices and 300k faces for each hemisphere. Note that the input white matter surfaces can be generated by other faster tools (Shattuck and Leahy, 2002; Henschel et al., 2020).

Implementation Details. The PialNN model consists of $L = 3$ layers of deformation blocks. We set the smoothing coefficient $\lambda = 1$ and kernel size $K = 5$ for 3D CNN. The Adam optimiser with learning rate 10^{-4} is used to train the PialNN model for 200

Table 3.1: Comparison of geometric errors for pial surface reconstruction. The results include the comparison with existing deep learning baselines and the ablation study. CD (mm), ASSD (mm) and HD90 (mm) are computed for both left and right hemispheres. A lower distance means a better result.

Method	Left Pial Surface			Right Pial Surface		
	CD	ASSD	HD90	CD	ASSD	HD90
DeepCSR-SDF	0.72±0.07	0.45±0.06	1.23±0.36	0.78±0.11	0.49±0.09	1.58±0.54
DeepCSR-OCC	0.66±0.04	0.42±0.04	0.87±0.13	0.65±0.05	0.40±0.04	0.88±0.20
Voxel2Mesh	0.58±0.03	0.34±0.04	0.82±0.09	0.57±0.02	0.31±0.02	0.80±0.07
PialNN (Ours)	**0.39±0.01**	**0.21±0.02**	**0.45±0.04**	**0.39±0.02**	**0.20±0.02**	**0.44±0.04**
Single Scale	0.42±0.02	0.23±0.02	0.50±0.05	0.43±0.02	0.23±0.02	0.51±0.05
Point Sampling	0.56±0.03	0.40±0.04	0.87±0.09	0.57±0.03	0.41±0.05	0.91±0.11
GCN	0.39±0.02	0.21±0.02	0.46±0.04	0.40±0.01	0.21±0.01	0.46±0.04

epochs with batch size 1. Experiments compare the performance of PialNN with state-of-the-art deep learning baselines, such as Voxel2Mesh (Wickramasinghe et al., 2020) and DeepCSR (Cruz et al., 2021). Since Voxel2Mesh uses iterative mesh unpooling, the input white matter surface is simplified to a mesh with 5120 faces using quadric error metric decimation (Garland and Heckbert, 1997). For DeepCSR, we train two different models using occupancy fields (DeepCSR-OCC) and signed distance functions (DeepCSR-SDF) as ground truth. The size of the implicit representation for DeepCSR is set to $192 \times 224 \times 192$ in order to have a reasonable number of vertices for a fair comparison. All models are trained on an NVIDIA GeForce RTX 3080 GPU.

Geometric Accuracy. We evaluate the geometric accuracy of the PialNN framework by computing the error between the predicted pial surfaces and FreeSurfer ground truth. We utilise three distance-based metrics to measure the geometric error, namely, Chamfer distance (CD) (Fan et al., 2017; Wang et al., 2018), average symmetric surface distance (ASSD) and 90th percentile of Hausdorff distance (HD90) (Cruz et al., 2021). The CD measures the mean distance between two sets of vertices. ASSD and HD90 compute the average and maximum distance between two sets of 150k sampled points from surface meshes. All distances are computed bidirectionally in millimetres (mm). A lower distance means a better result. Table 3.1 shows that PialNN achieves the best geometric accuracy

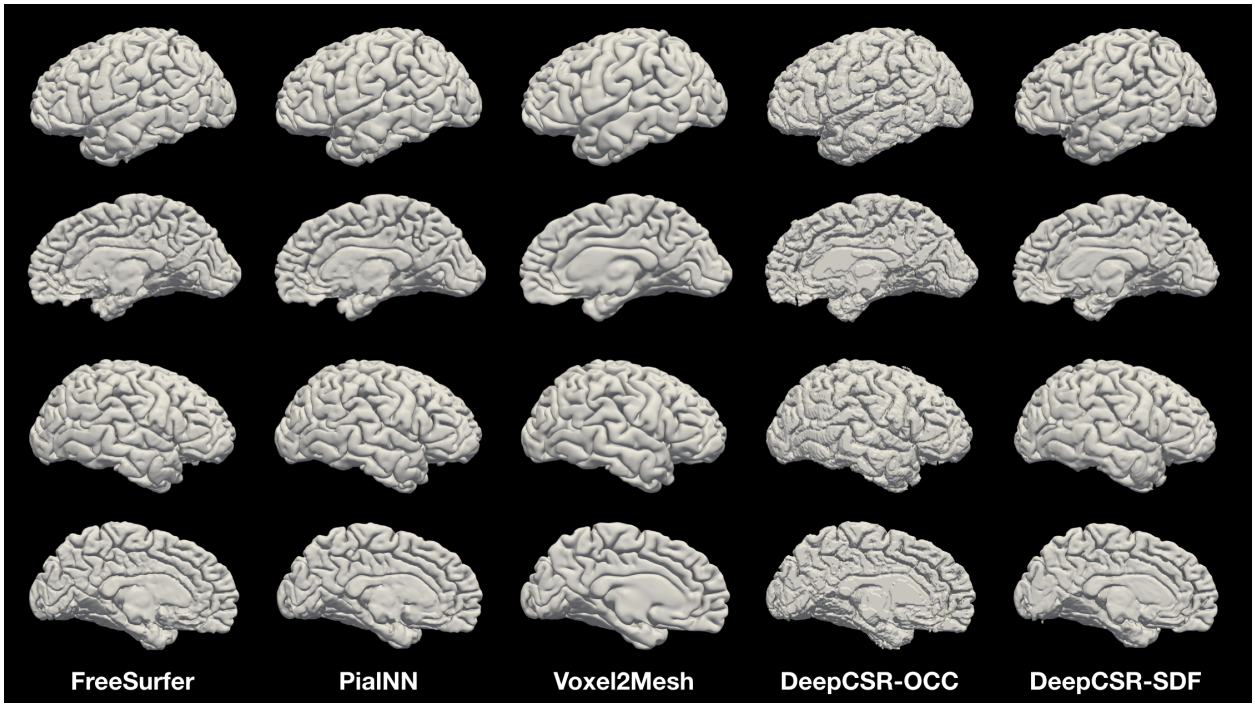


Figure 3.2: Visualisation of pial surface meshes reconstructed by different approaches.

compared with existing deep learning baselines. It reduces the geometric error by more than 30% compared to Voxel2Mesh and DeepCSR in all three distances (mm). In addition, the quality of the predicted pial surface mesh is visualised in Figure 3.2.

Figure 3.3 provides a detailed qualitative comparison between different approaches. The DeepCSR-SDF-2x represents DeepCSR-SDF with input size of $384 \times 448 \times 384$. We focus on the areas highlighted by the blocks in different colours. In the red block, the DeepCSR frameworks fail to distinguish two separate regions on the surface. The issue remains unsolved after increasing the input size. The yellow block indicates an inaccurate Voxel2Mesh prediction since the mesh is oversmoothed. In the orange block, only FreeSurfer and PialNN reconstruct the deep and narrow cortical sulci accurately. The green block indicates the error of Voxel2Mesh and DeepCSR-OCC in a sulcus region. It is noted that PialNN makes accurate reconstruction in all highlighted areas.

Figure 3.3 further visualizes that the predicted pial surface from Voxel2Mesh is oversmoothed, which can be a result of the mesh regularisation terms used in the loss function. Besides, it loses the geometric prior provided by the input white matter surface due

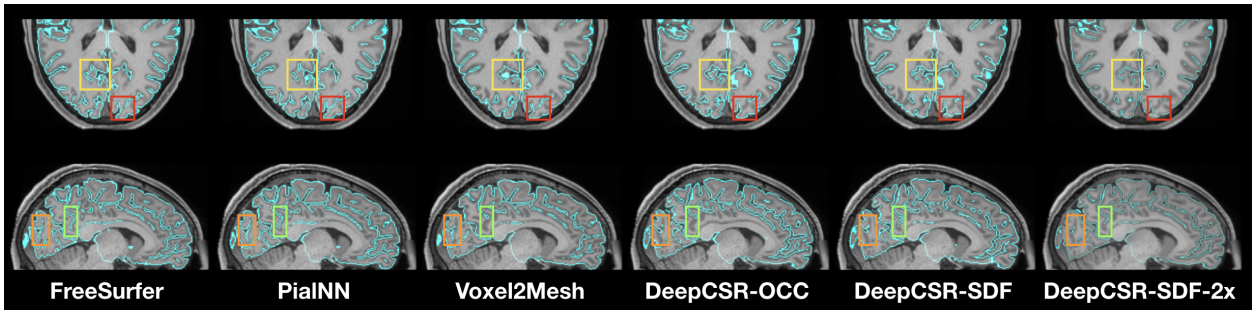


Figure 3.3: Qualitative comparison between predicted pial surfaces (cyan).

to the mesh simplification and decimation. Regardless of the input grid size, DeepCSR is prone to fail in the deep sulcus regions, since the implicit representation can be affected by partial volume effects.

Ablation Study. We consider three ablation experiments to validate the effectiveness of each component in the PialNN architecture. First, we only use single-scale brain MRI rather than a multi-scale image pyramid. Second, instead of cube sampling, we only employ point sampling, which samples the MRI voxels at the exact position of each vertex without neighbourhood. Third, we substitute the MLP layers with Graph Convolutional Networks (GCN) (Kipf and Welling, 2016). The results are listed in Table 3.1 and the error maps are given in Figure 3.4, which shows the Chamfer distance between the output surface and the ground truth surface generated by FreeSurfer. The multi-scale input slightly improves the geometric accuracy, while cube sampling contributes considerably to the performance of PialNN. There is no notable improvement after replacing MLP with GCN layers, whereas the GPU memory usage has increased.

Runtime. We compute the runtime of each framework, as shown in Figure 3.5, for both left and right pial surfaces reconstruction. PialNN achieves the fastest runtime with 0.52 seconds, whereas traditional pipelines (Shattuck and Leahy, 2002; Fischl, 2012; Glasser et al., 2013) usually take >10 minutes for pial surface generation based on the white matter surface. Voxel2Mesh needs 4.8 seconds as it requires mesh simplification for the input. DeepCSR runs in >100 seconds due to the time-consuming topology correction.

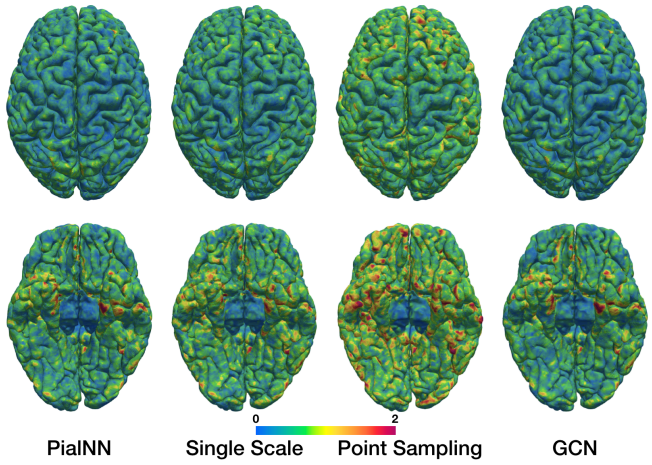


Figure 3.4: Error maps of pial surfaces in ablation study. The colour visualises the Chamfer distance ranging from 0 to 2 mm.

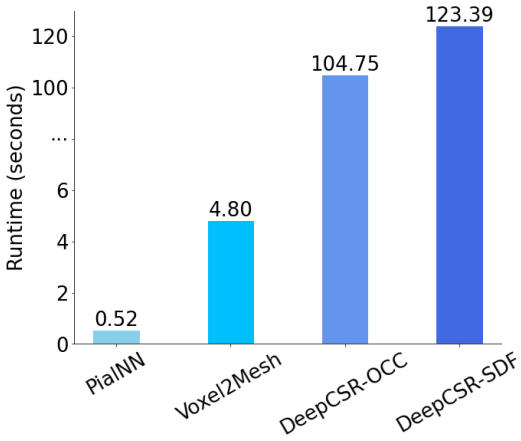


Figure 3.5: Runtime (seconds) of deep learning-based approaches for pial surface reconstruction.

3.1.5 Discussion

PialNN is a fast and memory-efficient deep learning framework for cortical pial surface reconstruction. The proposed framework learns several deformation blocks to generate a pial surface mesh from an input white matter surface. Each block incorporates the point feature extracted from the coordinates and normals, as well as the local feature extracted from the MRI intensity at the locations of the vertex and its neighbourhood. Experiments demonstrate that PialNN achieves the best performance with highest accuracy and fastest runtime (within one second) compared to state-of-the-art deep learning baselines.

However, the current PialNN framework does not provide the reconstruction of cortical white matter surfaces. In addition, PialNN is not able to avoid self-intersections in pial surfaces, especially the intersections between two adjacent banks of deep and narrow cortical sulci. In next section, we will introduce diffeomorphic surface deformations for both cortical white matter and pial surface reconstruction, with theoretical guarantees to prevent surface self-intersections.

3.2 Learning Diffeomorphic Cortical Surface Reconstruction by Neural ODEs

3.2.1 Introduction

The rapid advance of MRI has encouraged data acquisition for large cohort neuroimage studies, *e.g.*, ADNI (Jack Jr et al., 2008), ABIDE (Di Martino et al., 2014), OASIS (Marcus et al., 2007, 2010; LaMontagne et al., 2019), UK Biobank (Sudlow et al., 2015; Alfaro-Almagro et al., 2018), HCP (Van Essen et al., 2013), developing HCP (dHCP) (Edwards et al., 2022), and Baby Connectome Project (BCP) (Howell et al., 2019). To extract meaningful information from the resulting flood of data, various data processing pipelines have been developed for large-scale image analysis. One essential avenue for neuroimage studies is surface-based analysis of the cerebral cortex, which relies on accurate cortical surface reconstruction (Dale et al., 1999; Fischl et al., 1999a). As the cerebral cortex is highly folded, it is challenging to extract geometrically accurate cortical surface meshes. Naturally a cortical surface mesh must be a closed manifold and topologically equivalent to a 2-sphere without any holes or handles (Fischl et al., 1999a,b, 2001).

Traditional Neuroimage Processing Pipelines. Traditional cortical surface reconstruction approaches generally employ an empirically defined automatic processing pipeline. For example, the well-known neuroimage analysis software, FreeSurfer (Fischl, 2012), reconstructs white matter surfaces by applying mesh tessellation to the segmented WM, followed by mesh smoothing (Dale and Sereno, 1993) and topology correction (Fischl et al., 2001; Ségonne et al., 2007). The white matter surfaces are then refined and expanded to pial surfaces by deformable surface models along with self-intersection checking (Dale et al., 1999). Although many neuroimage processing pipelines have been proposed and widely used, such as FreeSurfer (Dale et al., 1999; Fischl et al., 1999a; Fischl, 2012), CIVET (MacDonald et al., 2000; Kim et al., 2005), BrainSuite (Shattuck and Leahy, 2002), CRUISE

(Xu et al., 1999; Han et al., 2004), the HCP pipeline (Glasser et al., 2013) and the dHCP pipeline (Makropoulos et al., 2018), they are united by two major limitations. Firstly, these pipelines are usually time-consuming as they integrate several computationally intensive image and geometric processing algorithms. This makes it challenging to scale up to large neuroimage population studies. Secondly, it is difficult to develop a single pipeline that can robustly process brain images from different age groups. For instance, neonatal brain MR images differ considerably in intensity values, size and shape from adults (Allen et al., 2002; Prastawa et al., 2005; Nie et al., 2012; Orasanu et al., 2014; Makropoulos et al., 2016; Hughes et al., 2017). This leads to poor reconstruction results when methods designed for adult neuroimage analysis are applied to such younger groups.

Learning-based Cortical Surface Reconstruction. As an end-to-end alternative to traditional pipelines, deep learning approaches have shown great potential for the reconstruction of 3D shapes while exhibiting convincingly fast runtime and powerful representation learning abilities. An increasing number of deep neural networks (DNNs) have been developed to tackle cortical surface reconstruction (Henschel et al., 2020; Wickramasinghe et al., 2020; Cruz et al., 2021; Ma et al., 2021; Hong et al., 2021; Gopinath et al., 2021; Lebrat et al., 2021). For instance, FastSurfer (Henschel et al., 2020) reduced the processing time of the original FreeSurfer pipeline (Fischl, 2012) by adopting a convolutional neural network (CNN) for brain tissue segmentation and a fast spherical mapping for cortical surface reconstruction. Cruz et al. (2021) presented the DeepCSR framework to predict implicit surface representations involving occupancy fields and signed distance functions (SDFs). Both FastSurfer and DeepCSR rely on topology correction (Fischl et al., 2001; Bazin and Pham, 2005, 2007; Ségonne et al., 2007) to guarantee the genus-0 topology of the surfaces. They use the Marching Cubes algorithm (Lorensen and Cline, 1998) for isosurface extractions. However, topology correction algorithms are computationally expensive, leading to >30 minutes of total runtime for both methods. Additionally, the surfaces extracted from volumetric representations are easily corrupted by partial volume effects (Ballester et al., 2002), especially for cortical sulci in pial surfaces.

Instead of learning an implicit surface representation, explicit learning-based approaches (Wickramasinghe et al., 2020; Ma et al., 2021) aim to deform an initial mesh with predefined topology to a target mesh. To extract surfaces from 3D medical images, Wickramasinghe et al. (2020) proposed Voxel2Mesh to learn iterative mesh deformations based on volumetric input. Several regularisation terms were introduced to improve the mesh quality. We presented the PialNN architecture (Ma et al., 2021) in Section 3.1 to expand an initial white matter surface to a pial surface according to the local information captured from brain MRI volumes. These explicit approaches are fast since no topology correction is required. Nevertheless, the main issue is that there are no theoretical guarantees to prevent surface self-intersections, in particular those between two adjacent sulcal banks in cortical pial surfaces.

Learning Diffeomorphic Surface Deformation. Non-manifold cortical surfaces with self-intersections can lead to inaccurate estimation of morphological features of the cerebral cortex, such as cortical thickness, curvature, surface area and volume (Fischl et al., 1999a; Fischl and Dale, 2000). Diffeomorphic transformation (Ruelle and Sullivan, 1975; Arsigny, 2004; Younes, 2010; Lee, 2012) is an effective way to preserve the manifoldness of the surfaces. In neuroimage analysis, the diffeomorphic transformation has been widely used for image registration and many frameworks have been presented such as LDDMM (Beg et al., 2005), DARTEL (Ashburner, 2007), and diffeomorphic demons (Vercauteren et al., 2009). For learning-based approaches, Balakrishnan et al. (2019) and Dalca et al. (2018, 2019) proposed VoxelMorph framework to learn stationary velocity fields for medical image registration. The deformation field is solved by scaling and squaring method (Higham, 2005; Arsigny et al., 2006).

To learn diffeomorphic surface deformation, Gupta and Chandraker (2020) proposed Neural Mesh Flow (NMF) with which they introduced Neural Ordinary Differential Equations (ODEs) (Chen et al., 2018) for 3D mesh generation. NMF learns a sequence of diffeomorphic flows between two meshes and models the trajectories of the mesh vertices

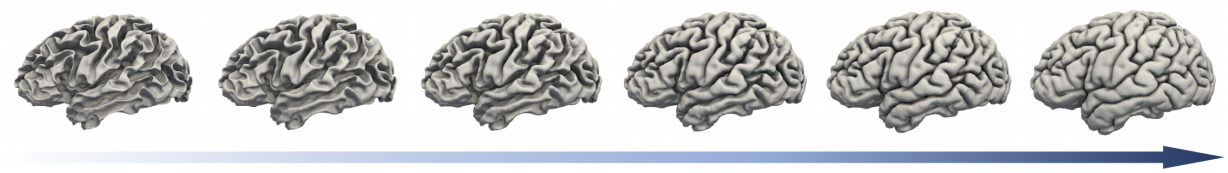


Figure 3.6: A diffeomorphic flow modelled by CortexODE. A white matter surface (leftmost) is deformed into a pial surface (rightmost).

as ODEs (Coddington et al., 1956; Walter, 2013). This implicitly regularises the manifold properties and reduces self-intersections in the generated meshes. However, NMF is limited by its insufficient shape representation capacity since it only augments a global feature vector for all vertices. Recently, Lebrat et al. (2021) presented the CorticalFlow framework for cortical surface reconstruction. CorticalFlow employs diffeomorphic mesh deformation (DMD) modules to learn a series of stationary velocity fields, which define diffeomorphic deformations from an initial template mesh to white matter and pial surfaces. The forward Euler method is adopted to approximate the flow ODE and the numerical condition is derived to constrain the step size of the integration. This produces manifold meshes that reduce self-intersections effectively.

Compared to traditional cortical surface reconstruction approaches using deformable surface models (Dale et al., 1999; MacDonald et al., 2000; Han et al., 2004; Kim et al., 2005; Schuh et al., 2017) or point matching approaches such as the non-rigid iterative closest point (ICP) algorithm (Arun et al., 1987; Besl and McKay, 1992; Zhang, 1994; Chui and Rangarajan, 2000), diffeomorphic surface deformations produce cortical surfaces as smooth manifolds with higher surface quality. Moreover, the diffeomorphic transformations theoretically preserve the surface topology and prevent self-intersections, except for those introduced by numerical errors from discretisation, thereby achieving high computational efficiency and topological accuracy without the dependence on time-consuming topology correction (Fischl et al., 2001; Shattuck and Leahy, 2001; Han et al., 2002; Bazin and Pham, 2005, 2007; Ségonne et al., 2007) and self-intersection detection (Möller, 1997; Dale et al., 1999; Park et al., 2001) required by traditional approaches.

Contributions. In Section 3.1, we have introduced the PialNN model (Ma et al., 2021) for pial surface reconstruction. However, the current PialNN framework still relies on traditional neuroimage processing pipelines to generate white matter surfaces as input. Furthermore, PialNN can produce self-intersections during pial surface deformation, causing inaccurate estimation of cortical attributes such as cortical thickness. To address these limitations in PialNN, in this section, we propose a novel geometric deep learning model, called CortexODE, for both cortical white matter and pial surface reconstruction with theoretical guarantees to avoid surface self-intersections.

Based on the Neural ODE framework (Chen et al., 2018), CortexODE extends PialNN (Ma et al., 2021) to learn diffeomorphic surface deformations for cortical surface reconstruction. It predicts continuous dynamics to deform an input surface to a target shape based on volumetric input. The trajectory of each vertex in the surface mesh is modelled as an ODE by a learnable deformation network following the architecture of PialNN. This constructs a diffeomorphic flow between surfaces as illustrated in Figure 3.6. The deformation network is proved to be Lipschitz continuous, and the upper bound of the Lipschitz constant is estimated to theoretically reduce surface self-intersections (Coddington et al., 1956; Ruelle and Sullivan, 1975; Walter, 2013; Perko, 2013; Chen et al., 2018; Gupta and Chandraker, 2020). We further provide numerical conditions for general Runge-Kutta method to ensure each integration step is a homeomorphism.

We incorporate CortexODE into a deep learning-based processing pipeline. Given an input brain MRI, our pipeline employs a 3D U-Net (Ronneberger et al., 2015) to predict a WM segmentation, which is transformed to an SDF by a distance transform method (Butt and Maragos, 1998). To obtain an initial closed manifold mesh, the SDF is processed by a topology correction algorithm (Bazin and Pham, 2005, 2007), which is re-implemented and improved to run within one second. An initial white matter surface is extracted by Marching Cubes algorithm (Lorensen and Cline, 1998). Then, two CortexODE models learn to shrink the initial surface to a white matter surface and expand the white matter

surface to a pial surface respectively. An entire CortexODE-based pipeline only needs 5 seconds to extract cortical surfaces of both brain hemispheres. The pipeline is able to make continuous prediction in long-term neuroimaging studies after appropriate training. We evaluate the performance on public large-scale neuroimage datasets across different age groups, including dHCP (Edwards et al., 2022), HCP (Wickramasinghe et al., 2020) and ADNI (Jack Jr et al., 2008) for neonates, young and old adults respectively. The CortexODE code is publicly available at `https://github.com/m-qiang/CortexODE`. The main contributions are summarised as:

- The CortexODE-based pipeline can extract high quality cortical surfaces from brain MRI volumes within 5 seconds. It achieves the highest geometric accuracy compared to other state-of-the-art deep learning-based frameworks.
- CortexODE learns a diffeomorphic flow to deform an input surface to a target shape, which provides theoretical guarantees for preventing self-intersections without additional regularisation terms.
- We re-implement and accelerate the standard topology correction algorithm (Bazin and Pham, 2005, 2007) to run $20\times$ faster.
- The CortexODE pipeline is evaluated on multiple large-scale neuroimage datasets in various age groups. The reliability and robustness of the proposed pipeline are validated by test-retest experiments and ablation studies.

3.2.2 CortexODE Framework

The CortexODE framework models the trajectories of points on an input surface as ODEs, where the derivatives of the points are parameterised by a DNN called *deformation network*. We show that the deformation network is Lipschitz continuous with respect to the input points. This provides theoretical guarantees for the prevention of surface self-

intersections according to the existence and uniqueness theorem for the solutions of ODE initial value problems (IVPs) (Coddington et al., 1956; Walter, 2013).

3.2.2.1 Neural ODE-based Framework

Neural ODEs. A Neural ODE (Chen et al., 2018) can be defined as:

$$\frac{dx(t)}{dt} = F_\theta(x(t), t), \tag{3.3}$$

where the right-hand side (RHS) function F_θ is a DNN with parameters θ. $x(t) \in \Omega$ is a continuous hidden state of the neural network for $t \in [0, T]$, and Ω is an arbitrary domain such as $\mathbb{R}^n$ and colour images. The Neural ODE takes $x(0) = x_0$ as the input while the output is the final state $x(T)$. The unique solution of the ODE (3.3) exists if the RHS function F_θ is Lipschitz continuous (Coddington et al., 1956; Walter, 2013), *i.e.*, there exists a constant $L_F > 0$ such that for any $x, y \in \Omega$,

$$\|F_\theta(x) - F_\theta(y)\|_p \leq L_F \|x - y\|_p, \tag{3.4}$$

where $\|x\|_p = (\sum_i |x_i|^p)^{1/p}$ is the p-norm for $p \geq 1$.

CortexODE. We propose CortexODE framework for cortical surface reconstruction based on Neural ODEs. Given an initial surface $S_0 \subset \mathbb{R}^3$, CortexODE models the trajectory of each point on the surface as an autonomous ODE:

$$\frac{dx(t)}{dt} = F_\theta(x(t), U), \; x(0) = x_0, \tag{3.5}$$

where $x_0 \in S_0$ is the coordinate of a point on the surface S_0. The solution $x(t) = x_0 + \int_0^t F_\theta(x(s), U)\mathrm{d}s$ represents the trajectory of each point for $t \in [0, T]$. $x(T)$ is the final coordinate after deformation. $U \in \mathbb{R}^{D_1 \times D_2 \times D_3}$ is a volumetric input, and the RHS function $F_\theta : \mathbb{R}^3 \times \mathbb{R}^{D_1 \times D_2 \times D_3} \to \mathbb{R}^3$ is a learnable *deformation network* as shown in Figure 3.7. During

the integration of the ODE (3.5), the deformation network F_θ is evaluated repeatedly to predict the deformations of all surface points at each time step.

CortexODE learns a continuously deformable surface S_t with points $x(t) \in S_t$ for $t \in [0, T]$. A diffeomorphic flow (Ruelle and Sullivan, 1975; Younes, 2010; Lee, 2012; Gupta and Chandraker, 2020) is defined by the ODE in Equation (3.5) from the initial surface S_0 to the final prediction S_T. According to the existence and uniqueness theorem for ODE solutions (Coddington et al., 1956; Walter, 2013), if F_θ is Lipschitz continuous with respect to x, the trajectories $x(t)$ of different points will never intersect with each other, which theoretically prevents the surface from intersecting with itself. In practice, however, the self-intersections can be introduced by various sources of numerical errors, *e.g.*, ODE integration, mesh discretization, and volume resolution.

3.2.2.2 Deformation Network

The architecture of the deformation network F_θ, *i.e.*, the RHS function of ODE (3.5), is shown in Figure 3.7. It follows the architecture of PialNN (Figure 3.1) and is designed to be Lipschitz continuous with respect to the coordinates of input surface points. Similar to PialNN, the inputs of the deformation network consist of point features extracted from surface points as well as local features learned from brain MRI volumes.

Point Feature. Given m points on the input surface S, we use a fully connected (FC) neural network layer f_1 to extract the *point feature* $z_{point} = f_1(x) \in \mathbb{R}^C$ from the coordinate $x \in \mathbb{R}^3$ of each point, where C is the dimension of the feature vector z_{point}. The l-th FC layer is defined as $f_l(x) = \phi(W_l x + b_l)$, where ϕ is a non-linear activation function, W_l and b_l are learnable weight and bias. Unlike PialNN, the normal vectors of the surface points are not involved in the point features. This is necessary for the Lipschitz continuity of the deformation network, since two neighbouring surface points in Euclidean space may have opposite normal directions.

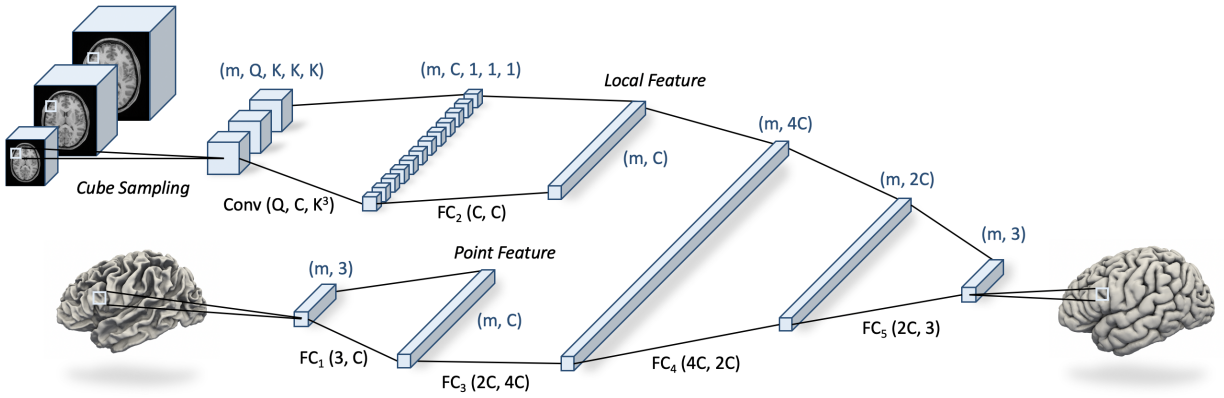


Figure 3.7: The architecture of the deformation network F_θ in the CortexODE framework. The deformation network extracts a point feature from the coordinate of the surface point, as well as a local feature from multi-scale brain MRI inputs by cube sampling. Two feature vectors are concatenated to predict the derivatives of the points. Here m is the number of the input surface points, C is the dimension of the feature vectors, and Q is the number of the input scales. The deformation network F_θ is evaluated repeatedly at each integration step to solve the Neural ODE.

Cube Sampling. Next, the local information of each surface point is captured from the brain MRI intensity. As shown in Figure 3.7, for each surface point x, we find its location in the multi-scale brain MRI inputs. For each scale, a cube of size K^3 is extracted, which contains the voxel at the location x as well as its neighbourhood voxels in the brain MRI. This procedure is called *cube sampling*.

More specifically, for a brain MRI volume U with size of $D_1 \times D_2 \times D_3$, we consider multi-scale inputs U_q with size of $\lfloor \frac{D_1}{2^{q-1}} \rfloor \times \lfloor \frac{D_2}{2^{q-1}} \rfloor \times \lfloor \frac{D_3}{2^{q-1}} \rfloor$, which downsamples the original image by $1/2^{q-1}$ for $q \in \{1, .., Q\}$, where Q is the number of the input scales. Then for each coordinate x, we can find its corresponding location in all volumes u_q. The voxel value $u_q \in \mathbb{R}$ at the location x in the q-th scale brain MRI input U_q is computed by a trilinear interpolation $u_q = \mathrm{Lerp}(x/2^{q-1}, U_q)$, where $\mathrm{Lerp}(\cdot)$ denotes the interpolation function. The coordinate x is scaled by $1/2^{q-1}$ to match the location in U_q. To capture the neighbourhood information of x, for each scale q, we consider a grid of size K^3 with centre $x/2^{q-1}$, which is defined by a set $\{x/2^{q-1}+\delta_{ijk}\}_{i,j,k\in\{1,..,K\}}$ for $\delta_{ijk} := [i - \lceil K/2 \rceil, j - \lceil K/2 \rceil, k - \lceil K/2 \rceil] \in \mathbb{Z}^3$. Therefore, for each scale q, we can sample a cube $\mathbf{u}_q$ of size K^3, containing the voxel values at the locations of all points in the grid. Letting $\mathbf{u}(x) \in \mathbb{R}^{Q\times K\times K\times K}$ be the stack of all cubes

$\mathbf{u}_q$ at point x, then each entry of $\mathbf{u}$ can be defined as:

$$\mathbf{u}_{qijk}(x) = \text{Lerp}\left(x/2^{q-1} + \delta_{ijk}, U_q\right), \tag{3.6}$$

for $1 \leq q \leq Q$ and $i, j, k \in \{1, ..., K\}$. Such a cube sampling method can extract local information effectively from the input brain MRI with arbitrary image resolution.

Local Feature. After cube sampling, we obtain a total of $m \cdot Q$ cubes. As shown in Figure 3.7, a convolutional layer with kernel size K followed by a FC layer f_2 is employed to encode the cubes $\mathbf{u}(x)$ to a *local feature* $z_{local} \in \mathbb{R}^C$ for each point $x \in S$. Here the convolutional layer is equivalent to a FC layer f_0, so that we have $z_{local} = f_2 \circ f_0(\mathbf{u}(x))$.

Deformation. We concatenate the point feature and local feature of x to a new feature vector, which is passed to a sequence of FC layers f_3, f_4, f_5 to learn the final deformation $F_\theta(x, U) \in \mathbb{R}^3$. Finally, the deformation network F_θ can be expressed explicitly as:

$$F_\theta(x, U) = f_5 \circ f_4 \circ f_3 \left(\left[f_1(x)^\top, f_2 \circ f_0(\mathbf{u}(x))^\top\right]^\top\right). \tag{3.7}$$

3.2.2.3 Existence and Uniqueness of Solutions

In this section, we show the deformation network F_θ is Lipschitz continuous, so that there exits a unique solution for the IVP (3.5). Assuming the activation functions ϕ in F_θ are ReLU or LeakyReLU, we have the following theorem.

Theorem 3.1. *The RHS function F_θ of the ODE in Equation (3.5) is Lipschitz continuous with respect to the input $x \in \mathbb{R}^3$ with Lipschitz constant*

$$L_F = \|W_5\|_p \|W_4\|_p \|W_3\|_p \left(\|W_1\|_p^p + 2K^3 (U_{\max} - U_{\min})^p \|W_2\|_p^p \|W_0\|_p^p\right)^{\frac{1}{p}}. \tag{3.8}$$

Then, there exists a unique solution for the IVP in Equation (3.5).

Proof. See Section 3.2.6. □

Here $U_{\max}$ and $U_{\min}$ denote the maximum and minimum entry of the input brain MRI volume U. Based on the Theorem 3.1, for different initial points $x_0 \in S_0$, there exist unique solutions $x(t)$ of the ODE (3.5) with respect to the initial values, and the trajectories of the solutions will never intersect with each other. This theoretically prevents self-intersections in deformable surface S_t for $t \in [0, T]$.

3.2.2.4 CortexODE Discretisation

To solve the IVP for ODE (3.5) in CortexODE, we use explicit fixed-step methods for ODE discretisation (Süli and Mayers, 2003):

$$x_{n+1} = x_n + h\Delta x_n, \tag{3.9}$$

where $n \in \{0, ..., N-1\}$, N is the number of the total steps of the iteration, and $h = T/N$ is the step size. To estimate Δx_n, Lebrat et al. (2021) adopted explicit forward Euler method. In order to approximate CortexODE, we consider an r-stage explicit Runge-Kutta (RK) method (Butcher, 1987, 2016; Süli and Mayers, 2003). The general form of the RK method for an autonomous ODE can be formulated as:

$$x_{n+1} = x_n + h \sum_{i=1}^{r} \beta_i s_i(x_n), \tag{3.10}$$

where s_i is defined as follows:

$$s_i(x) = F_\theta\Big(x + h \sum_{j=1}^{i-1} \alpha_{i,j} s_j(x)\Big). \tag{3.11}$$

The coefficients β_i and $\alpha_{i,j}$ are defined by a Butcher tableau (Butcher, 1963). F_θ is the deformation network. Next, we show that under certain conditions, each integration step in Equation (3.10) is a homeomorphism, which reduces self-intersections numerically.

Theorem 3.2. *Let $\mathcal{G}(x) = x + h\sum_{i=1}^{r}\beta_i s_i(x)$ where r is the stage of the RK method. If the time step h and the Lipschitz constant L of F_θ satisfy $\eta(h, L) < 1$, where*

$$\eta(h, L) := \sum_{c=1}^{r} h^c L^c \sum_{i_0=1}^{r} |\beta_{i_0}| \prod_{k=1}^{c-1} \sum_{i_k=1}^{i_{k-1}-1} |\alpha_{i_{k-1},i_k}|, \tag{3.12}$$

then $\mathcal{G}$ is a homeomorphism, i.e., *$\mathcal{G}$ is a continuous bijection and $\mathcal{G}^{-1}$ is continuous.*

Proof. See Section 3.2.6. □

To discretise CortexODE, we consider forward Euler, Midpoint, and 4th-order RK methods as ODE solvers (Süli and Mayers, 2003). Theorem 3.2 extends the results of homeomorphism from an Euler method (Lebrat et al., 2021) to an any-stage RK method. The Euler method can be viewed as a special case of the RK method with $\beta_1 = 1$ and $\alpha_{i,j} = 0$. Then the sufficient condition of homeomorphism is $\eta(h, L) = hL < 1$. Similarly, for 2nd-order Midpoint method with $\beta_1 = 0$, $\beta_2 = 1$ and $\alpha_{2,1} = 1/2$, it should hold $\eta(h, L) = hL + \frac{1}{2}h^2L^2 < 1$. Due to the repeated evaluation of the RHS deformation network F_θ at each integration step, we use a larger step size h (*e.g.*, $h = 0.1$) to solve the ODEs during training to reduce the training time. In the inference phase, we can use a smaller h to reduce surface self-intersections introduced by the numerical errors. (See Section 3.2.4.3 for detailed discussion.)

By solving the ODE, CortexODE deforms an initial surface S_0 to the final predicted surface S_N with points x_N. According to Theorem 3.2, the surface self-intersections can be prevented numerically as each integration step is a bijective mapping. More specifically, for each integration step, any two distinct points on the surface cannot be mapped to the same position. However, it should be noted that for a polygon mesh representation, we only deform the vertices of the mesh instead of all points on the surface. As a result, the limited mesh resolution, *i.e.*, the number of vertices, can also introduce a few mesh self-intersections (See Section 3.2.4.3).

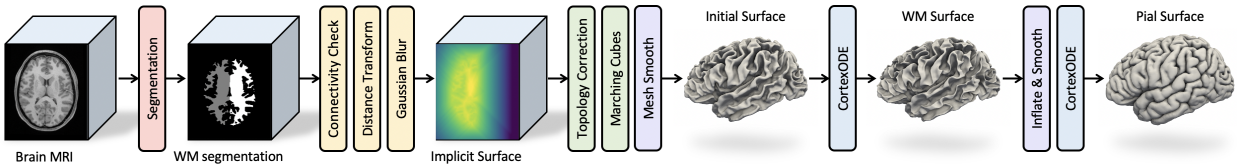


Figure 3.8: The CortexODE-based cortical surface reconstruction pipeline. Given a brain MRI volume, the pipeline learns to predict a WM segmentation and then transforms it to a signed distance function (SDF). After topology correction and isosurface extraction, two CortexODE models are trained to deform an initial surface into a white matter surface and to deform an inflated white matter surface into a pial surface respectively.

3.2.3 CortexODE-based Processing Pipeline

In this section, we introduce a CortexODE-based processing pipeline for both cortical white matter and pial surface reconstruction, which is depicted in Figure 3.8. The pipeline is composed of white matter segmentation, signed distance transformation, fast topology correction, initial surface extraction, and CortexODE-based surface deformations.

3.2.3.1 White Matter Segmentation

Given an input brain MRI volume $U \in \mathbb{R}^{D_1 \times D_2 \times D_3}$, the CortexODE-based pipeline uses a DNN to learn a WM segmentation map $B \in [0, 1, 2]^{D_1 \times D_2 \times D_3}$. The segmentation has three categories involving background, left WM and right WM. Note that here the WM segmentation refers to the interior of the white matter surface, which contains cortical WM, deep grey matter, ventricle, hippocampus and other tissues within the surface. To predict the segmentation, a 3D U-Net (Ronneberger et al., 2015) is trained by minimising the cross entropy loss $\mathcal{L}_{ce}$ between the prediction and ground truth (GT) segmentations. The GT segmentations are generated by traditional pipelines (Fischl, 2012; Glasser et al., 2013; Makropoulos et al., 2018) which leverage the anatomical information of the brain MRI. More advanced segmentation approaches, such as QuickNAT (Roy et al., 2019) and FastSurfer (Henschel et al., 2020), can be used to improve the performance of the segmentation and the following surface reconstruction (see Section 3.2.4.5).

3.2.3.2 Signed Distance Transformation

As shown in Figure 3.8, based on the predicted WM segmentation B, we generate a signed distance function (SDF) which represents an implicit surface. We first check the connectivity of the binary WM segmentation and select the largest connected component which contains most of the voxels. Next, a distance transform algorithm (Butt and Maragos, 1998) is adopted to transform the segmentation B to a volumetric level set $\mathcal{B}_d \in \mathbb{R}^{D_1 \times D_2 \times D_3}$, where the value of each voxel in $\mathcal{B}_d$ is the distance from its location to the boundary of the WM segmentation as shown in Figure 3.9. The voxels in the exterior of the segmentation have negative value while the values of the interior voxels are positive. A Gaussian filter with standard deviation σ is used to smooth the volumetric level set. Then, a continuous SDF $\mathcal{B} : \mathbb{R}^3 \to \mathbb{R}$ is obtained by applying a trilinear interpolation $\mathrm{Lerp}(\cdot)$ to the level set $\mathcal{B}_d$, that is, $\mathcal{B}(\cdot) = \mathrm{Lerp}(\cdot, \mathcal{B}_d)$. Each level $\lambda \in \mathbb{R}$ in the SDF defines an implicit surface:

$$S^{\lambda} = \{x \mid \mathcal{B}(x) = \lambda\}, \tag{3.13}$$

where $x \in \mathbb{R}^3$ are the points on the surfaces. The 0-level surface S^0 defines an implicit white matter surface.

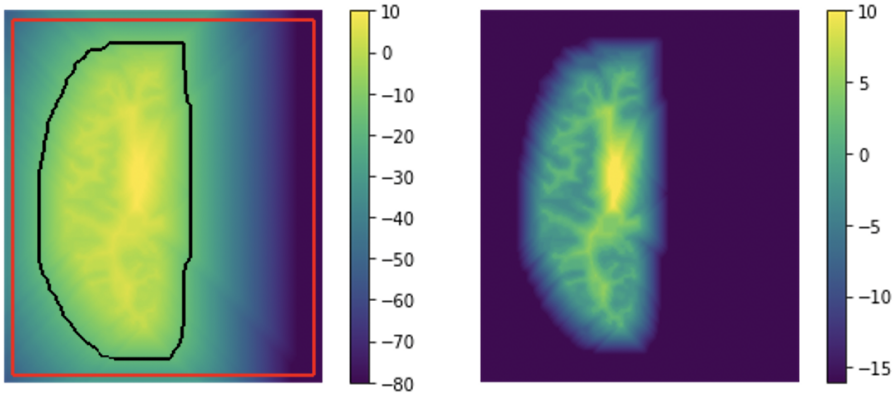


Figure 3.9: Left: a discrete SDF. The red bounding box marks the initial points for the original topology correction algorithm (Bazin and Pham, 2005, 2007). The black contour marks the initial points on the boundary $S^{\hat{\lambda}}$ for our improved version. Right: the topology correction result with boundary $S^{\hat{\lambda}=-16}$.

3.2.3.3 Topology Correction

To guarantee spherical topology of the implicit surfaces, we apply the topology correction algorithm by Bazin and Pham (2005, 2007) to the discrete SDF $\mathcal{B}_d$. The algorithm adopts a topology-preserving fast marching approach to fix the topology of the surfaces S^λ defined by $\mathcal{B}$ under any level λ. However, since the original implementation (Bazin and Pham, 2005, 2007; Cruz et al., 2021) is computationally intensive, we re-implement and improve the original algorithm, which is consequently accelerated by a factor of 20. First of all, we re-implement the topology correction algorithm (Bazin and Pham, 2005, 2007) in Python, while the original algorithm was implemented by Java. We use Numba (Lam et al., 2015) to accelerate the algorithm. Numba is a just-in-time (JIT) compiler for Python, which is suitable for dense computation.

In addition, we improve the initialisation of the original algorithm (Bazin and Pham, 2005, 2007). To fill the holes or handles in the input level set, the original algorithm starts topology correction from initial points on the boundary of the level set, which is shown as the red bounding box in Figure 3.9. The algorithm processes the voxels iteratively from lower level to higher level. Therefore, only the voxels inside the boundary need to be processed. For cortical surface extraction, since we consider one brain hemisphere at a time, instead of correcting the topology of the entire level set, we only need to process the voxels inside a $\hat{\lambda}$-level surface $S^{\hat{\lambda}}$ for $\hat{\lambda} < 0$. This not only accelerates the runtime but also ensures that the 0-level surface, *i.e.*, the white matter surface, has desired topology.

To obtain the boundary points of the surface $S^{\hat{\lambda}}$, we first create a binary mask containing all voxels inside the surface $S^{\hat{\lambda}}$. More precisely, the voxel of the binary mask is set to one if its level λ satisfies $\lambda \geq \hat{\lambda}$. Then, we apply morphological dilation to enlarge and expand the binary mask. The dilated voxels represent the boundary of $S^{\hat{\lambda}}$, which is illustrated as the black contour in Figure 3.9. These boundary voxels are used as initial points by topology correction algorithm, and the values of the voxels outside the boundary are set to $\hat{\lambda}$. Figure 3.9-Right shows a topologically corrected result with $\hat{\lambda} = -16$. The initial-

isation avoids processing of 76% of the voxels. Such an accelerated topology correction algorithm only needs one second to process a volume of size $192 \times 224 \times 192$.

3.2.3.4 Cortical Surface Reconstruction

Initial Surface Extraction. We use the Marching Cubes algorithm (Lorensen and Cline, 1998) to extract an initial white matter surface $S_0 = S^{\lambda_0}$ from the topology corrected SDF at the level $\hat{\lambda} < \lambda_0 < 0$, which is between the white matter and pial surfaces. The initial surface S_0 is represented by a 3D triangular mesh $\mathcal{M}_0 = \{\mathcal{V}_0, \mathcal{E}_0, \mathcal{F}_0\}$ with vertices $x_0^i \in \mathcal{V}_0$ for $i \in \{1, ..., m\}$, where $m = |\mathcal{V}_0|$ is the number of vertices. $\mathcal{E}_0$ and $\mathcal{F}_0$ are the sets of edges and faces. The initial mesh $\mathcal{M}_0$ is smoothed twice by Laplacian smoothing $\bar{x}_0^i = \sum_{j \in \mathcal{N}(i)} x_0^j / |\mathcal{N}(i)|$, where $\mathcal{N}(i)$ is the adjacency list of the i-th vertex.

White Matter Surface Reconstruction. As shown in Figure 3.8, two CortexODE models learn to shrink the initial surface to a white matter surface and to expand the white matter surface to a pial surface respectively. To train the CortexODE models, we minimise the distance between predicted surface mesh $\widehat{\mathcal{M}}$ and pseudo ground truth (GT) mesh $\mathcal{M}_*$ without regularisation terms. We generate the pseudo GT surfaces by traditional pipelines (Fischl, 2012; Glasser et al., 2013; Makropoulos et al., 2018). For white matter surface reconstruction, the input for CortexODE is the initial surface generated by the pipeline. The bidirectional Chamfer distance (Fan et al., 2017; Wang et al., 2018; Wickramasinghe et al., 2020) is computed as the loss function $\mathcal{L}_{cd}$ between $\widehat{\mathcal{M}}$ and $\mathcal{M}_*$:

$$\mathcal{L}_{cd}(\widehat{\mathcal{M}}, \mathcal{M}_*) = \frac{1}{|\widehat{\mathcal{X}}|} \sum_{\hat{x} \in \widehat{\mathcal{X}}} \min_{x_* \in \mathcal{X}_*} \|\hat{x} - x_*\|^2 + \frac{1}{|\mathcal{X}_*|} \sum_{x_* \in \mathcal{X}_*} \min_{\hat{x} \in \widehat{\mathcal{X}}} \|\hat{x} - x_*\|^2, \tag{3.14}$$

where $\hat{x} \in \widehat{\mathcal{X}}$ and $x_* \in \mathcal{X}_*$ are the coordinates of the point clouds $\widehat{\mathcal{X}}$ and $\mathcal{X}_*$, which are sampled uniformly with 150k points on the predicted and pseudo GT meshes.

Pial Surface Reconstruction. Since it is challenging to reconstruct an accurate pial surface, we use an inflated white matter surface as the input for CortexODE to extract pial

surfaces. The inflating and smoothing process is given in Algorithm 3.1. Following the training procedure of PialNN in Section 3.1.3.2, we use inflated pseudo GT white matter surfaces as the input for training CortexODE, which have the same vertex connectivity as pseudo GT pial surfaces. Thus, the predicted pial surfaces also have point-to-point correspondence to the pseudo GT pial surfaces in the training. Then, we can compute the mean squared error (MSE) loss $\mathcal{L}_{mse}$ between the predicted pial surfaces $\widehat{\mathcal{M}}$ and pseudo GT pial surfaces $\mathcal{M}_*$ directly without point matching:

$$\mathcal{L}_{mse}(\widehat{\mathcal{M}}, \mathcal{M}_*) = \frac{1}{m}\sum_{i=1}^{m} \|\hat{x}^i - x_*^i\|^2, \tag{3.15}$$

where $m = |\widehat{\mathcal{V}}| = |\mathcal{V}_*|$ is the number of mesh vertices.

Algorithm 3.1: Mesh Inflating and Smoothing

Input: A mesh $\mathcal{M}$ with m vertices x^i, the number of iterations N, and the scale parameter ρ.

for $n \leq N$ **do**
  for $i \leq m$ **do**
    $x^i = \sum_{j\in\mathcal{N}(i)} x^j/|\mathcal{N}(i)|$; `/* Smoothing */`
  for $i \leq m$ **do**
    Compute normals n^i;
    $x^i = x^i + \rho n^i$; `/* Inflating */`

3.2.4 Experiments

3.2.4.1 Experiment Details

Dataset. We evaluate the performance of our CortexODE pipeline on three publicly available datasets: the ADNI dataset (Jack Jr et al., 2008), the WU-Minn HCP Young Adult dataset (Van Essen et al., 2013), and the third released dHCP dataset (Makropoulos et al., 2018; Edwards et al., 2022). We split each dataset into training, validation and testing data by the ratio of 6/3/1. Specifically, for the ADNI dataset, we use 524 T1-weighted (T1w)

brain MRI from subjects aged from 55 to 90 years. The MRI volumes are aligned rigidly to the MNI152 template (Fonov et al., 2011) and clipped to the size of $176 \times 208 \times 176$ at $1mm^3$ isotropic resolution. The ground truth (GT) segmentations and surfaces are generated by FreeSurfer v7.2 (Fischl, 2012). For the HCP dataset, we use 600 T1w brain MRI from 22-36 year-old young adults. The volume size is $192 \times 224 \times 192$ with $1mm^3$ voxel size. The pseudo GT data are produced by the HCP pipeline (Glasser et al., 2013). The dHCP dataset is collected from neonates between 25 to 45 weeks. We use 874 T2w MRI images resampled to size of $160 \times 208 \times 208$ at $0.7mm^3$ isotropic resolution. The pseudo GT surfaces are extracted by the dHCP neonatal pipeline (Makropoulos et al., 2018). For all datasets, the pseudo GT white matter surface mesh and pial surface mesh have the same vertex connectivity. The coordinates of the vertices are normalised to $[-1, 1]$.

Implementation Details. For initial surface generation, we train a U-Net for 200 epochs to predict the WM segmentations. After distance transform, the SDF is smoothed by a Gaussian filter with standard deviation $\sigma = 0.5$. For topology correction, we set the level $\hat{\lambda} = -16$ for initialisation. The initial white matter surfaces are extracted at level $\lambda_0 = -0.8$. The white matter surfaces are inflated and smoothed twice with scale parameter $\rho = 0.002$ (Algorithm 3.1) as the input to reconstruct pial surfaces.

For CortexODE model, we use $Q = 3$ scales for the input and the side length of the cubes is set to $K = 5$. We set $C = 128$ as the dimension of feature vectors. Our model is lightweight and it only contains 328,835 learnable parameters. The model is discretised by an Euler solver with total numerical steps $N = 10$ and step size $h = 0.1$ for training. For inference, we set $N = 20$ for all datasets to reduce self-intersections.

We adopt the *adjoint sensitivity method* proposed by Chen et al. (2018) to train CortexODE models. This allows constant GPU memory costs during training despite the step size of ODE integration. We train four CortexODE models for 500 epochs to reconstruct white matter and pial surfaces of both left and right brain hemispheres. The Adam optimiser is used with learning rate 10^{-4} and the batch size is set to 1. The training time and

GPU memory costs are reported in Section 3.2.4.3 for different ODE solvers. The entire CortexODE-based pipeline requires 5.29GB GPU memory for inference.

Baselines. We compare our CortexODE-based pipeline with state-of-the-art (SOTA) deep learning approaches for cortical surface reconstruction, including PialNN (Ma et al., 2021), CorticalFlow (Lebrat et al., 2021), DeepCSR (Cruz et al., 2021), and Voxel2Mesh (Wickramasinghe et al., 2020). Since PialNN is designed for pial surface reconstruction, we use initial white matter surfaces generated by the CortexODE-based pipeline as the input of PialNN to reconstruct white matter surfaces. The same initial white matter surface meshes are simplified to 5k vertices as the input for Voxel2Mesh, as it is difficult to deform spheres to cortical surfaces. The input mesh is subdivided iteratively to 164k vertices by Voxel2Mesh. For CorticalFlow, we use subject-specific convex hulls as the input surfaces instead of a global convex hull which leads to unstable training in our experiments. For DeepCSR, the predicted SDF is $2\times$ the size of the input MRI volume to reduce partial volume effects. For acceleration, we apply the same topology correction algorithm as used in our pipeline for DeepCSR. For CorticalFlow and DeepCSR, the reconstructed surface meshes have up to 500k vertices. For CortexODE and PialNN, the predicted meshes have approximately 120k–160k vertices for the HCP and ADNI datasets, as well as 60k–150k vertices for the dHCP dataset. For fair comparison, all models including CortexODE are trained on an NVIDIA GeForce RTX 3080 GPU with 12GB memory.

3.2.4.2 Comparative Results

We compare the geometric accuracy, self-intersections and runtime of CortexODE with SOTA deep learning-based cortical surface reconstruction baselines. The comparative results are given in Table 3.2 and Figure 3.10–3.12.

Geometric Accuracy. We consider two distance-based metrics to measure geometric accuracy: average symmetric surface distance (ASSD) and Hausdorff distance (HD) (Cruz

Table 3.2: Comparative results of cortical surface reconstruction on geometric accuracy and self-intersections. The average symmetric surface distance (ASSD), Hausdorff distance (HD), and the ratio of the self-intersecting faces (SIF) are measured for white matter and pial surfaces on both brain hemispheres. The mean value (M), standard deviation (SD), and p-value are reported. For all metrics lower mean values indicate better results. *Significant difference ($p < 0.05$) compared to CortexODE.

		White Matter Surface									Pial Surface								
		ASSD (mm)			HD (mm)			SIF (%)			ASSD (mm)			HD (mm)			SIF (%)		
	Method	M	SD	p	M	SD	p	M	SD	p	M	SD	p	M	SD	p	M	SD	p
dHCP	CortexODE	0.107	0.024	—	0.210	0.073	—	0.020	0.014	—	0.128	0.027	—	0.275	0.074	—	**0.284**	0.147	—
	PialNN	**0.106**	0.023	0.612	**0.207**	0.072	0.500	0.041	0.041	<0.001*	**0.117**	0.026	<0.001*	**0.245**	0.069	<0.001*	9.401	1.992	<0.001*
	CorticalFlow	0.142	0.035	<0.001*	0.285	0.133	<0.001*	0.073	0.056	<0.001*	0.172	0.041	<0.001*	0.436	0.174	<0.001*	8.627	2.367	<0.001*
	DeepCSR	0.216	0.083	<0.001*	0.509	0.328	<0.001*	—	—	—	0.401	0.102	<0.001*	1.522	0.504	<0.001*	—	—	—
	Voxel2Mesh	0.262	0.040	<0.001*	0.557	0.102	<0.001*	**0.010**	0.018	<0.001*	0.270	0.048	<0.001*	0.664	0.136	<0.001*	6.315	1.617	<0.001*
HCP	CortexODE	**0.115**	0.011	—	**0.233**	0.022	—	**0.004**	0.004	—	**0.163**	0.018	—	**0.347**	0.040	—	**0.367**	0.155	—
	PialNN	0.117	0.010	0.077	0.237	0.021	0.015*	0.033	0.037	<0.001*	0.194	0.019	<0.001*	0.413	0.041	<0.001*	2.599	0.929	<0.001*
	CorticalFlow	0.163	0.015	<0.001*	0.353	0.040	<0.001*	0.389	0.250	<0.001*	0.236	0.025	<0.001*	0.557	0.061	<0.001*	1.049	0.472	<0.001*
	DeepCSR	0.201	0.034	<0.001*	0.424	0.079	<0.001*	—	—	—	0.310	0.052	<0.001*	0.827	0.223	<0.001*	—	—	—
	Voxel2Mesh	0.275	0.021	<0.001*	0.641	0.068	<0.001*	0.011	0.017	<0.001*	0.306	0.025	<0.001*	0.753	0.068	<0.001*	1.962	0.606	<0.001*
ADNI	CortexODE	0.197	0.046	—	0.423	0.103	—	**0.010**	0.011	—	**0.186**	0.049	—	**0.399**	0.094	—	0.142	0.090	—
	PialNN	**0.194**	0.046	0.468	**0.417**	0.104	0.463	0.086	0.084	<0.001*	0.206	0.042	<0.001*	0.438	0.087	<0.001*	2.975	1.441	<0.001*
	CorticalFlow	0.254	0.050	<0.001*	0.587	0.139	<0.001*	0.055	0.055	<0.001*	0.238	0.044	<0.001*	0.557	0.124	<0.001*	**0.064**	0.037	<0.001*
	DeepCSR	0.296	0.051	<0.001*	0.633	0.101	<0.001*	—	—	—	0.292	0.064	<0.001*	0.616	0.165	<0.001*	—	—	—
	Voxel2Mesh	0.372	0.048	<0.001*	0.907	0.151	<0.001*	0.033	0.036	<0.001*	0.344	0.043	<0.001*	0.802	0.115	<0.001*	0.493	0.314	<0.001*

et al., 2021). The ASSD measures the average distance between two surfaces. It samples a batch of points on one surface and computes the mean distance between the points and the other surface, and vice versa. Similarly, the HD computes the maximum distance between two surfaces. For the purpose of robustness, the 90th percentile of the HD is computed rather than the maximum value (Cruz et al., 2021; Lebrat et al., 2021). We use PyTorch3D (Ravi et al., 2020) to uniformly sample 100k points on the surface meshes to compute bidirectional ASSD and HD between the predicted and pseudo GT surfaces for all baselines. The mean and standard deviation of the geometric errors are reported in Table 3.2. The lower distance means better geometric accuracy.

Furthermore, we use an independent t-test to examine the statistical significance between CortexODE and other baseline models. The p-value of the t-test is reported in Table 3.2. As common practice, we assume that $p < 0.05$ indicates there is significant difference between the geometric errors of CortexODE and compared baselines. Table 3.2 shows that CortexODE and PialNN achieve better performance on both white matter and pial surface reconstruction than other approaches. For white matter surface reconstruction, since PialNN uses the initial surfaces generated by CortexODE pipeline, there is no significant difference between CortexODE and PialNN. For pial surfaces, CortexODE is significantly better than PialNN on the HCP and ADNI datasets. Although PialNN outperforms CortexODE on the dHCP dataset, it produces 9.4% self-intersections which can lead to highly inaccurate estimation of cortical features (see Section 3.2.4.4).

Visual Inspection. We visualise the reconstructed cortical surfaces of the subjects in the dHCP, HCP and ADNI datasets. Figure 3.10 shows the cortical surface meshes with colour maps reflecting their geometric errors, and Figure 3.11 depicts the projected surfaces on the brain MRI slices. The artefacts of the surfaces are highlighted. According to the visualisation, the CortexODE-based pipeline produces surfaces with good mesh quality and high geometric accuracy. The results match the ground truth surfaces generated by the traditional pipelines, or even have better mesh quality. The pial surfaces re-

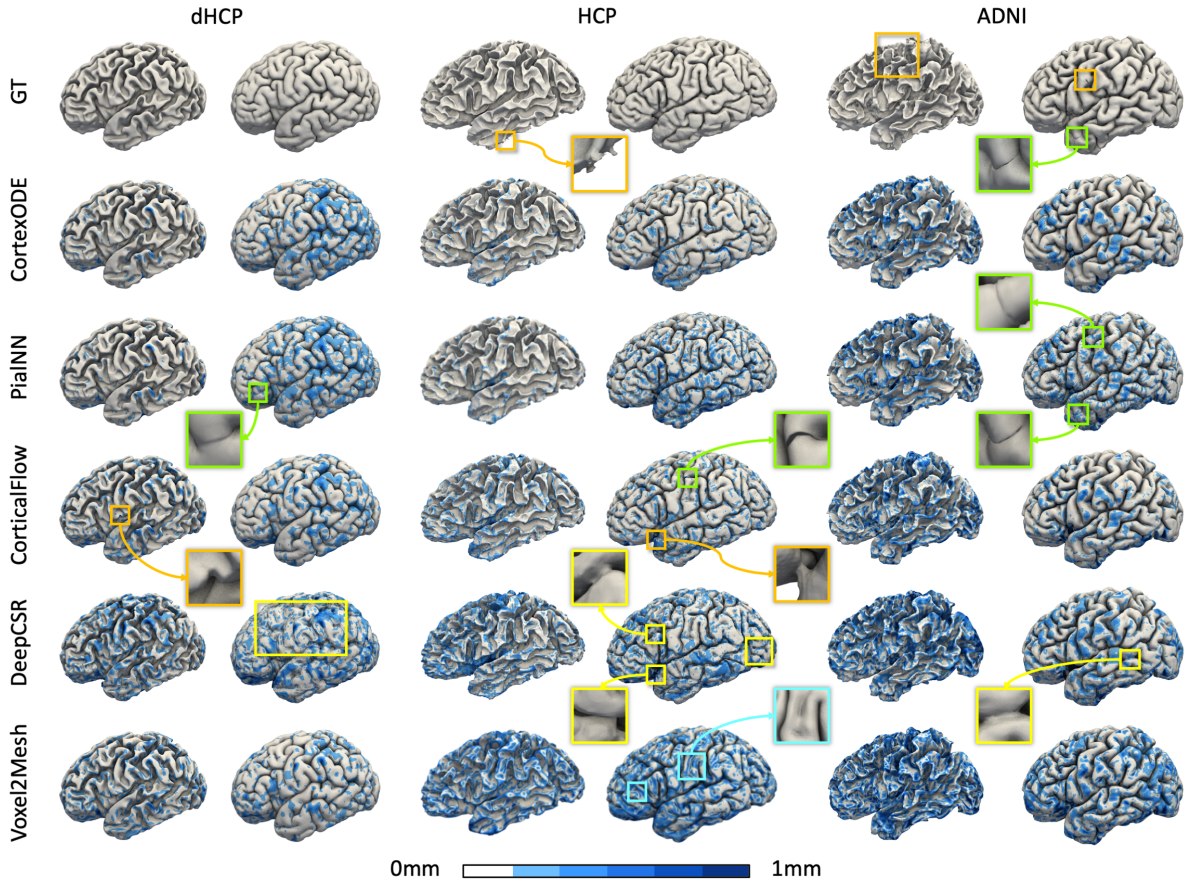


Figure 3.10: Visualisation of reconstructed cortical surfaces on dHCP, HCP and ADNI datasets. The geometric error map (blue) visualises the distance (0–1mm) between the predicted surfaces and the ground truth. The artefacts (corruption, self-intersection, partial volume effect, and over-smoothness) on surfaces are highlighted by different colours.

constructed by PialNN is not smoothed well and more self-intersections are observed as highlighted. The DeepCSR results are corrupted by partial volume effects in the narrow sulcal regions, especially on the dHCP dataset. The surfaces predicted by Voxel2Mesh tend to be over-smoothed due to the mesh regularisation terms.

Self-intersection. To verify that CortexODE reduces surface self-intersections effectively, we compute the percentage of self-intersecting faces (SIFs) as shown in Table 3.2. The results demonstrate that CortexODE produces almost no SIFs in white matter surfaces and less than 0.4% SIFs in pial surfaces. CortexODE has significantly fewer SIFs than all other approaches, except that CorticalFlow has fewer SIFs in pial surfaces on the ADNI datasets. However, as illustrated in Figure 3.11, the sulcal regions are more narrow for

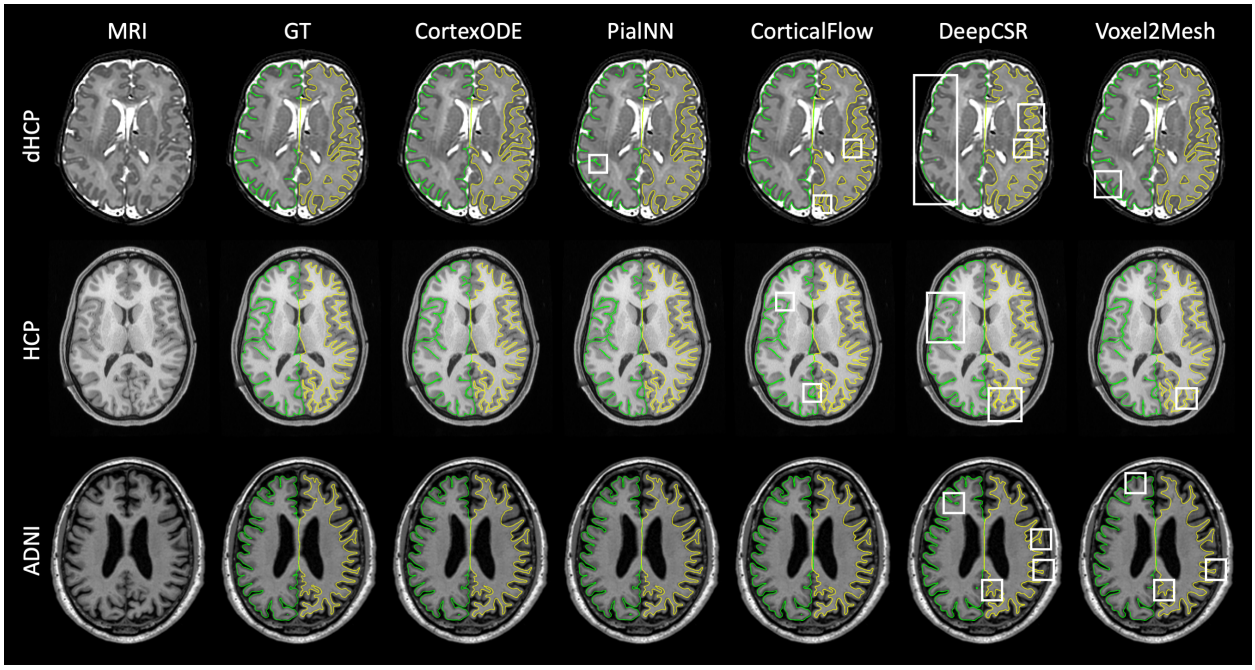


Figure 3.11: Visualisation of left pial surfaces (green) and right white matter surfaces (yellow) projected on MRI slices. The artefacts are highlighted by white boxes.

neonates than adults. Therefore, CorticalFlow fails to prevent SIFs on the dHCP datasets, as it is difficult to deform a convex hull into deep and narrow sulci without intersections. Although PialNN achieves similar geometric accuracy to CortexODE, it has much more SIFs than CortexODE in the pial surfaces as it has no theoretical guarantees to reduce SIFs, which impede its clinical applications (see Section 3.2.4.4). DeepCSR introduces no SIFs since the surfaces are extracted by Marching Cubes. For pial surface reconstruction, the input white matter surfaces are inflated along the normal direction (Algorithm 3.1). Therefore, our approach can also effectively reduce intersections between white matter and pial surfaces. In our experiments, CortexODE only produces 1.57/1.24/1.77% intersecting faces between white matter and pial surfaces on the dHCP/HCP/ADNI datasets.

Runtime Analysis. The inference time of all deep learning baselines is shown in Figure 3.12. We divide the runtime into volume-level (*e.g.*, segmentation, topology correction and isosurface extraction) and surface-level (*e.g.*, mesh smoothing, subdividing and deformation) processing time. The runtime is measured on an NVIDIA GeForce RTX 3080 GPU and Intel Core i7 CPUs. From Figure 3.12, since CortexODE, PialNN

and Voxel2Mesh use the same pipeline to generate initial surfaces, they have the same volume-level processing time. The DeepCSR framework requires 2 minutes due to the voxel-wise prediction on a high resolution volume. CorticalFlow achieves the fastest runtime because it has no volume-level pre-processing. In summary, the CortexODE-based pipeline is accurate and efficient. It runs within 5 seconds for all datasets, whereas traditional pipelines (Fischl, 2012; Glasser et al., 2013; Makropoulos et al., 2018) usually require more than 5 hours for a single subject's cortical surface reconstruction.

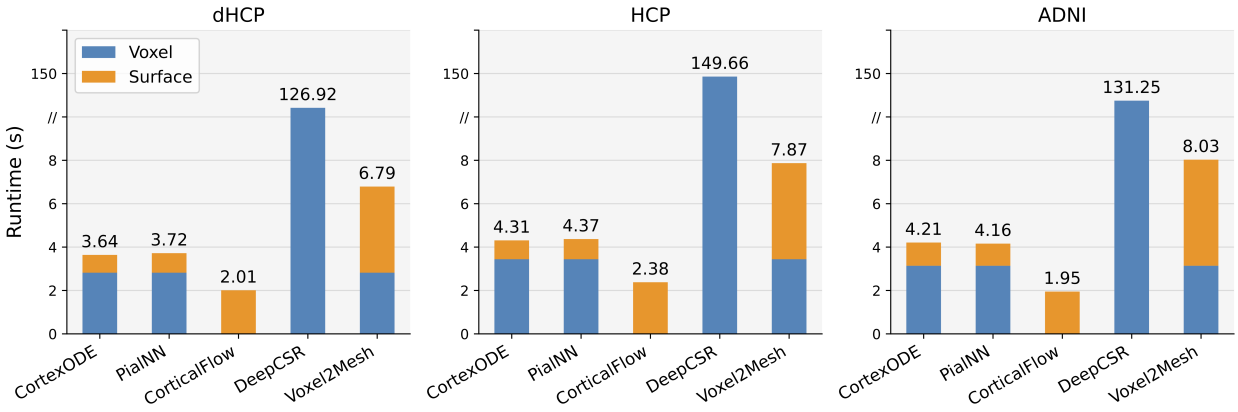


Figure 3.12: The runtime (second) of cortical surface reconstruction on dHCP, HCP and ADNI datasets. The blue bars indicate the voxel-level processing time and the orange bars indicate surface-level runtime.

3.2.4.3 ODE Integration

ODE Solvers. CortexODE can be discretised by various explicit ODE solvers. We compare the performance of CortexODE models using different ODE solvers including forward Euler, midpoint (Mid), and fourth-order Runge Kutta (RK4) (Süli and Mayers, 2003). For evaluation, we train CortexODE models for pial surface reconstruction on the HCP dataset with different ODE solvers, including Euler-10 ($N = 10$), Euler-20 ($N = 20$), Mid-10 ($N = 10$), and RK4-5 ($N = 5$). N is the number of total steps for ODE solvers in training and $h = T/N$ is the step size with $T = 1$s. We set the inflation scale $\rho = 0.001$ for the input white matter surfaces to avoid the SIFs introduced by inflation. Several metrics of the ODE solvers for training are reported in Table 3.3. The Lipschitz constant L_F

of the deformation network F_θ is computed based on Equation (3.8) with 2-norm. The higher-order methods require more number of function evaluation (NFE), *i.e.*, the times of forward propagation of F_θ, but the error decreases much faster. The training time increases linearly with the NFE, while the GPU memory cost is constant for training using adjoint method despite the step size.

Table 3.3: The metrics of ODE solvers for training. The training step size h_{train}, number of function evaluation (NFE), accumulated error of ODE solvers, training time t_{train} for each backpropagation iteration, GPU memory cost for training, and Lipschitz constant L_F are reported for different solvers.

Solver	h_{train}	NFE	Error	t_{train}	GPU	L_F
Euler-10	0.1	10	$\mathcal{O}(h)$	0.47s/it	7.15GB	94.26
Euler-20	0.05	20	$\mathcal{O}(h)$	0.93s/it	9.03GB	95.72
Mid-10	0.1	20	$\mathcal{O}(h^2)$	0.92s/it	8.97GB	110.10
RK4-5	0.2	20	$\mathcal{O}(h^4)$	0.92s/it	9.20GB	107.14

Performance. The performance of the trained models is evaluated for total steps $N = 10, 20, 50, 100, 200$ with step size $h = 1/N$. The condition $\eta(h, L_F) < 1$ in Theorem 3.2 holds for all solvers when $N = 200$. For each ODE solver, the ASSD, HD, percentage of SIFs, and inference time are reported in Figure 3.13. For the lower-order Euler method, Figure 3.13 reveals that by decreasing the step size, the ratio of SIFs reduces to 0.11% but the geometric error increases slightly. The higher-order methods such as midpoint and RK4 provide more accurate and stable approximation for ODE solutions, whereas they are more computationally expensive. The performance will not decrease when tested on small step sizes, while the SIFs do not decrease as well. In practice, we can choose suitable ODE solvers and step sizes for different scenarios determined by the demand for accuracy, efficiency and mesh quality.

Limitations. One limitation of CortexODE is that the SIFs still exist even though the condition $\eta(h, L) < 1$ is already satisfied. As discussed in Section 3.2.2.4, one reason is that the homeomorphic mapping should be applied to all points on a smooth surface to prevent self-intersections. However, the polygon mesh only has a limited number of vertices. This inevitable discretisation causes resolution limits and thus self-intersections.

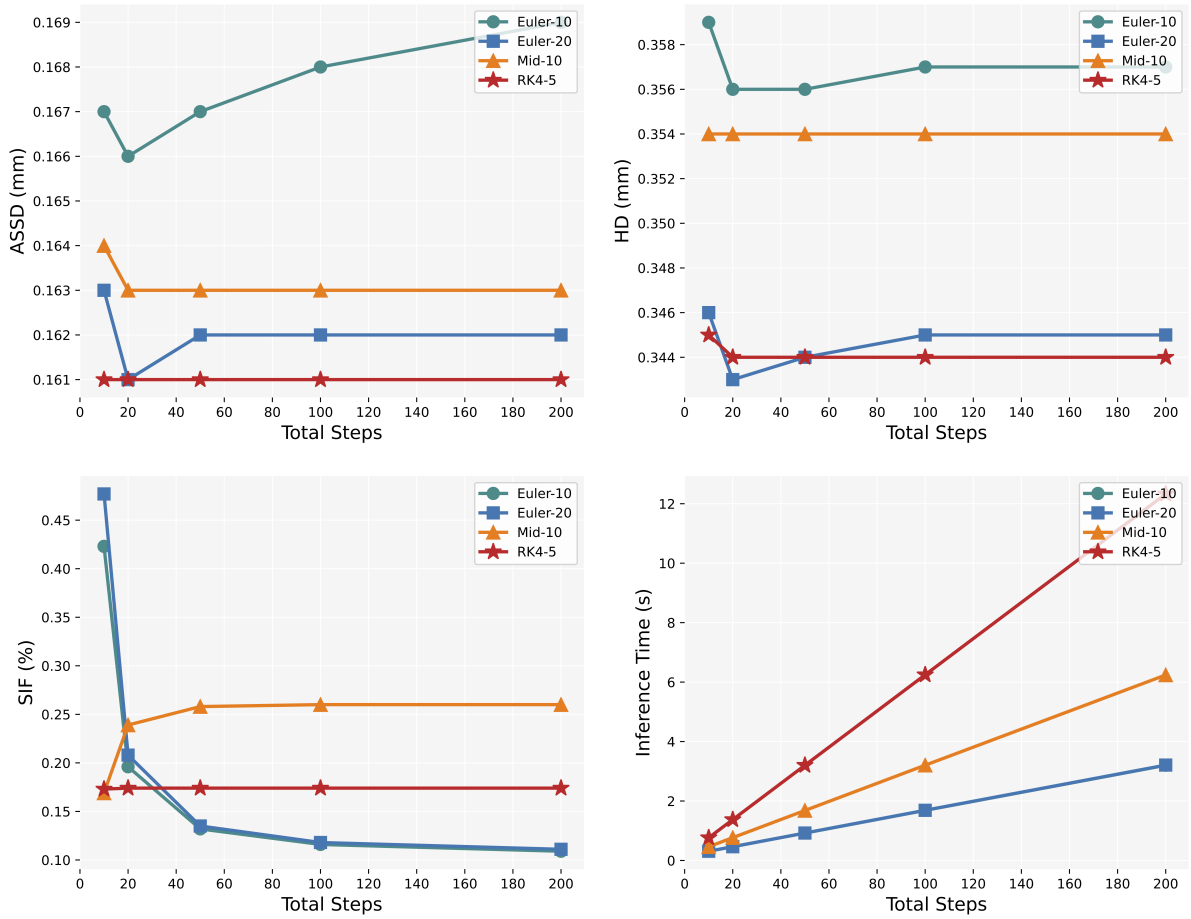


Figure 3.13: Comparisons on different ODE solvers. We set fixed step sizes to train the CortexODE models and use different step sizes for the inference. The ASSD, HD, SIFs and inference time are evaluated for total integration steps $N = 10, 20, 50, 100, 200$.

In addition, the CortexODE model is trained using a relatively large step size h, which produces SIFs during training. This also introduces SIFs in the inference despite using of smaller step sizes, especially for higher-order solvers which are more stable. It should be noted that although SIFs are not completely prevented, CortexODE still reduces SIFs effectively and achieves better mesh quality than existing learning-based approaches while providing theoretical support.

3.2.4.4 Clinical Analysis

Reliability. The reliability of cortical surface reconstruction is an important feature required by clinical applications. Based on previous work (Henschel et al., 2020; Cruz et al.,

Table 3.4: Comparative results of test-retest reliability. The ASSD and HD are measured between each pair of cortical surfaces on the Test-Retest dataset. *Significant difference ($p < 0.05$) compared to CortexODE.

	ASSD (mm)			HD (mm)		
Method	M	SD	p	M	SD	p
CortexODE	**0.208**	0.031	—	**0.440**	0.073	—
FreeSurfer	0.225	0.024	<0.001*	0.476	0.058	<0.001*
PialNN	0.213	0.031	<0.001*	0.445	0.070	0.038*
CorticalFlow	0.220	0.033	<0.001*	0.464	0.072	<0.001*
DeepCSR	0.268	0.063	<0.001*	0.573	0.148	<0.001*
Voxel2Mesh	0.323	0.066	<0.001*	0.694	0.153	<0.001*

2021), we evaluate the reliability of CortexODE on a Test-Retest dataset (Maclaren et al., 2014). The Test-Retest dataset contains 3 subjects, each of which has 40 brain MRI scans collected within a short period. Hence, the cortical surfaces of the same subject should be identical or highly consistent. We use models pre-trained on the ADNI dataset to extract the cortical surfaces from all brain MRI volumes. The ASSD and HD are measured between all pairs of the surfaces for each subject, where each pair of the surfaces are aligned by the ICP algorithm (Besl and McKay, 1992). The results including statistical significance (t-test) are reported in Table 3.4. It shows that CortexODE achieves the best consistency compared to FreeSurfer and existing learning-based approaches.

Surface Quality. The quality of the cortical surfaces can affect the estimation of cortical features which are crucial for downstream clinical analysis (Fischl and Dale, 2000; Aël Chetelat and Baron, 2003; Lerch et al., 2008; Rimol et al., 2012). We evaluate the impact of mesh self-intersections on the measurement of cortical thickness, (pial) surface area and cortical volume. We train a CortexODE model on the HCP dataset using an Euler-20 solver. The geometric accuracy and ratio of SIFs of pial surfaces, as well as the cortical features are reported in Table 3.5 as the *Baseline* to compare. We set the time step $h = 0.01$ for inference so that the results only have 0.118% SIFs. Next, we introduce two types of SIFs into the cortical surfaces without changing geometric accuracy. This provides fair comparisons to explore the effect of SIFs on the cortical features.

Table 3.5: The results of cortical feature estimation. The ASSD (mm), HD (mm), ratio of SIFs ($\%$), cortical thickness (mm), surface area (cm^2) and cortical volume (cm^3) for one brain hemisphere are reported. The results are bold if there are significant differences ($p < 0.05$*) compared to *Baseline*.

	Baseline		Flipping			Large Step		
Metrics	M	SD	M	SD	p	M	SD	p
ASSD	0.162	0.016	0.163	0.016	0.452	0.163	0.018	0.728
HD	0.345	0.035	0.346	0.035	0.699	0.346	0.039	0.678
SIF	0.118	0.064	**3.249**	**0.071**	<0.001*	**0.477**	**0.255**	<0.001*
Thickness	2.566	0.068	2.566	0.068	0.998	**2.597**	**0.068**	<0.001*
Area	1079	113	**1106**	**116**	0.002*	1087	113	0.379
Volume	257	29	257	29	0.910	**262**	**29**	0.030*

As the first type, we consider general SIFs that are caused by poor mesh quality. Given cortical pial surfaces predicted by CortexODE, we randomly flip the edges of the mesh, *i.e.*, flip the coordinates between any pair of adjacent vertices to produce extra SIFs in the surfaces. We flip 0.4% of the edges for each surface and evaluate the geometric errors (ASSD and HD) as well as the cortical features. We use a t-test to examine the significance between *Baseline* and *Flipping*. Table 3.5 shows that the flipping creates 3.25% SIFs but does not show significant difference in distance-based geometric errors. However, the surface area is significantly over-estimated due to the intersecting and overlapping between the faces of the pial surface mesh.

As the second type, SIFs often occur in the deep and narrow sulci of cortical pial surfaces due to partial volume effects (Ballester et al., 2002). To investigate this, we use a large step size $h = 0.1$ for CortexODE (*Large Step*) such that $\eta(h, L) > 1$ and SIFs in the sulci cannot be prevented. Table 3.5 indicates that, although no significant difference is found in geometric accuracy, the cortical thickness and volume of *Large Step* are increased significantly compared to the *Baseline*. This results from the increase of SIFs (0.477%) which occur along with the over-inflation of the cortical surfaces in sulcal regions. According to above experiments, despite the same geometric accuracy, the occurring of SIFs can significantly affect the estimation of cortical features. Therefore, it is essential to prevent surface self-intersections to provide reliable cortical features for clinical analysis.

Table 3.6: Ablation study for WM segmentation in CortexODE-based pipeline. The Dice coefficient (%), IoU (%) and runtime (second) for segmentation are reported. The ASSD and HD are compared between the cortical surfaces extracted by CortexODE-based pipeline using predicted segmentations.

	Segmentation			White Matter Surface		Pial Surface	
Method	Dice	IoU	Runtime	ASSD	HD	ASSD	HD
U-Net-Seg	98.07	96.21	0.596	0.115	0.233	0.163	0.347
U-Net-SDF	97.12	94.43	**0.235**	0.146	0.296	0.165	0.349
QuickNAT	**98.57**	**97.18**	8.342	**0.103**	**0.208**	**0.155**	0.332
FastSurfer	98.52	97.09	4.643	0.104	**0.208**	**0.155**	**0.331**

3.2.4.5 Ablation Study

We conduct ablation experiments to verify the effectiveness of the components in the CortexODE-based pipeline.

Segmentation. We fist investigate how different deep learning-based segmentation approaches can affect the geometric accuracy of cortical surfaces extracted by CortexODE. We compare the performance of WM segmentation between U-Net (U-Net-Seg) (Ronneberger et al., 2015), QuickNAT (Roy et al., 2019) and FastSurfer (Henschel et al., 2020) on the HCP dataset. In addition, we consider using a U-Net (U-Net-SDF) to predict the SDF directly. The WM segmentation of U-Net-SDF is defined by the voxels within 0-level. The Dice coefficient, Intersection over Union (IoU), and runtime of the approaches are evaluated and reported in Table 3.6, where the runtime consists of the inference time of neural networks and post-processing time of implicit surface creation. The geometric errors of reconstructed cortical white matter and pial surfaces using the predicted segmentations are reported in Table 3.6 as well. Table 3.6 shows that QuickNAT and FastSurfer can provide better segmentation, which results in more accurate cortical surfaces for CortexODE. However, both are computationally intensive and require 4 to 8 seconds for inference. U-Net-SDF performs worse than U-Net-Seg since it is more difficult to learn a distance-based level set than a binary segmentation. Moreover, deep learning-based approaches can produce imperfect initial WM segmentations, leading to inaccurate initial

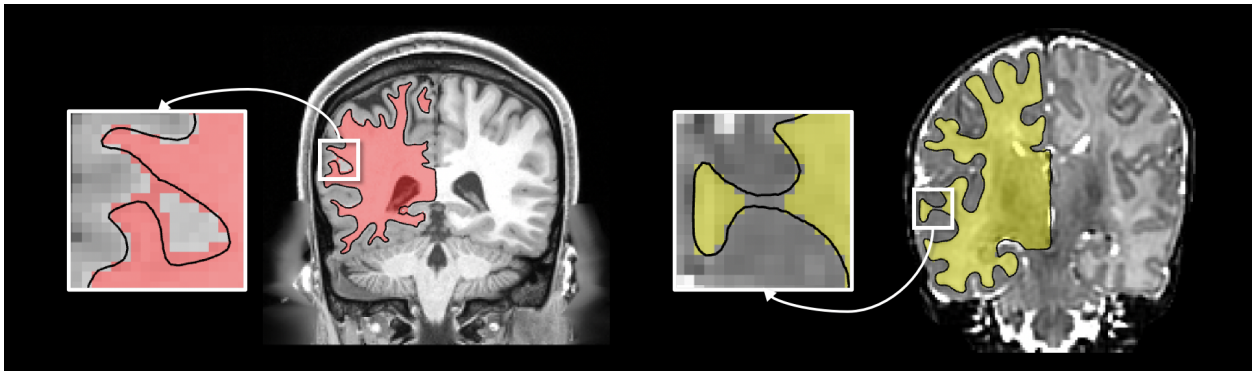


Figure 3.14: Cortical surfaces (black contour) extracted by CortexODE-based pipeline based on imperfect segmentations. Left: over-segmentation on the HCP data. Right: disconnected segmentation on the dHCP data.

surfaces. Figure 3.14 shows cortical surfaces predicted by CortexODE when the WM segmentation is over-segmented or disconnected. It demonstrates the robustness of our CortexODE-based pipeline, which reconstructs accurate cortical surfaces even if the WM segmentation is imprecise.

Topology Correction. We compare our accelerated topology correction algorithm to its original implementation (Bazin and Pham, 2005, 2007). The runtime is measured on an Intel Core i7 CPU. As shown in Table 3.7, for discrete SDFs with size of $192 \times 224 \times 192$ on the HCP dataset, the Numba-accelerated algorithm with improved initialisation (Numba+Init) only runs one second on average. This is $20\times$ faster than the original Java implementation which requires 21.48s. The Numba compiler accelerates the Python implementation from 144.38s to 3.45s. Our initialisation method reduces the number of processed voxels by 76% for topology correction, and thus it speeds up the original algorithm effectively. In all implementations, the corrected SDFs produce surfaces with the Euler characteristic number $\chi = V - E + F = 2$, which means the surfaces have genus zero topology. We also check the topology of cortical surfaces generated by the baseline models in Table 3.2. The average Euler characteristic number is 2 for all approaches, which means all predicted surfaces have the correct topology.

Initial Surface Extraction. We verify the effectiveness of the processing pipeline for initial surface extraction. More specifically, we conduct ablation studies on isosurface extraction

Table 3.7: Comparison on different implementations of the topology correction algorithm (Bazin and Pham, 2005, 2007). The runtime, percentage of processed voxels and Euler characteristic number are reported. The results are evaluated on the discrete SDFs of left white matter surfaces with size of $192 \times 224 \times 192$.

Method	Runtime	Processed Voxels	Euler Number
Numba+Init	0.99s	20.71%	2
Numba	3.45s	97.05%	2
Python+Init	27.78s	20.71%	2
Python	144.38s	97.05%	2
Java (Original)	21.48s	97.05%	2

and surface smoothing. For isosurface extraction, we employ Marching Cubes (Lorensen and Cline, 1998) to extract an inflated white matter surface as the initial surface at λ_0-level of the SDF instead of 0-level. As shown in Figure 3.15-Left, the implicit white matter surface defined by the SDF (0-level surface) is corrupted due to the imperfect segmentation, while the initial surface at λ_0-level (Figure 3.15-Middle) can reduce these corruptions effectively. Generally $-1 < \lambda_0 < -0.5$ is a reasonable range for the initial surface extraction. In our experiments, we set $\lambda_0 = -0.8$ which is validated to achieve superior results in various age groups including the dHCP, HCP and ADNI datasets (Table 3.2). For the HCP dataset, the geometric error (ASSD) is 0.607mm between initial surfaces and GT white matter surfaces. CortexODE deforms the initial surfaces into white matter surfaces (Figure 3.15-Right) and reduces the error to ASSD = 0.115mm. This performs much better than the implicit white matter surface at 0-level which only achieves ASSD = 0.211mm.

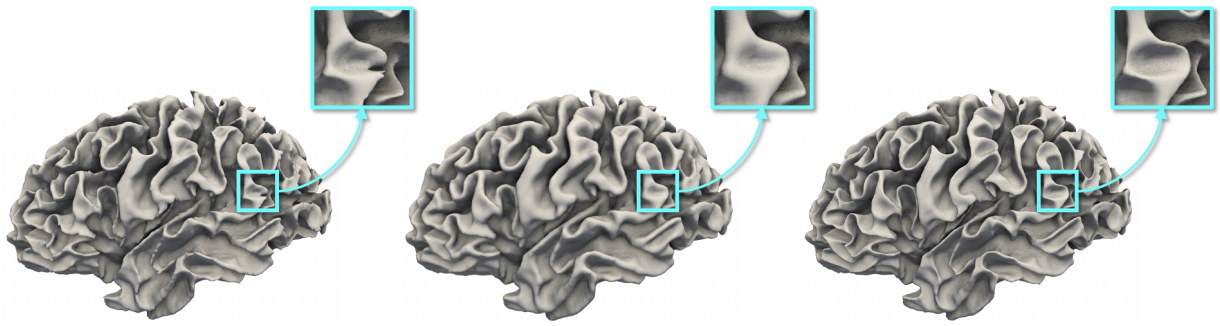


Figure 3.15: Ablation study for isosurface extraction of the initial surface. Left: implicit white matter surface defined by SDF (0-level surface). Middle: initial surface extracted at λ_0-level. Right: white matter surface predicted by CortexODE.

The CortexODE pipeline includes two smoothing steps. First, an implicit surface smoothing with a Gaussian filer is applied to the SDF to reduce the corruptions caused by inaccurate segmentation. Second, two iterations of Laplacian mesh smoothing are used to remove the artefacts introduced by Marching Cubes. Figure 3.16 visualises the initial surfaces without either of two smoothing steps. It turns out that the smoothing steps are necessary to produce high-quality surface meshes. In practice, the Gaussian blurring with standard deviation $\sigma = 0.5$ is sufficient to avoid most of the corruptions and keep the details of the surface.

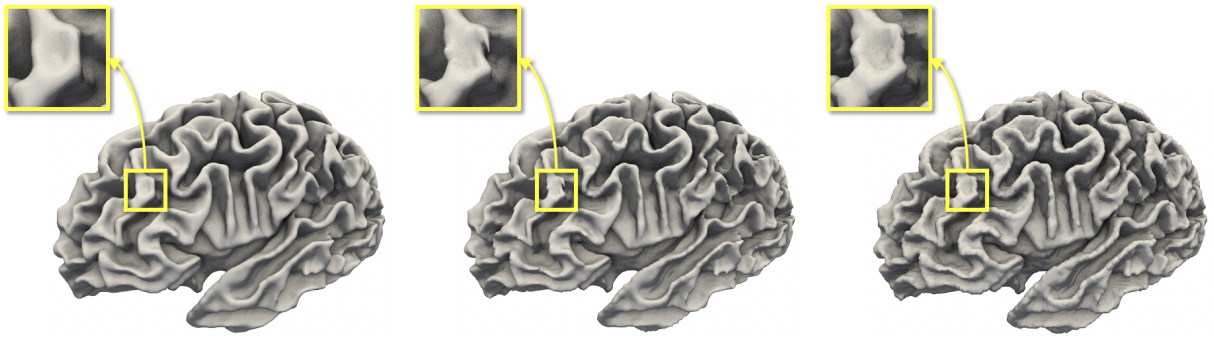


Figure 3.16: Ablation study for surface smoothing of the initial surface. Left: initial surface created by CortexODE. Middle: initial surface without implicit surface smoothing. Right: initial surface without Laplacian mesh smoothing.

CortexODE Architecture. To examine the architecture of the CortexODE framework, we consider the effect of different input scales and cube sampling sizes. We train CortexODE models with input scales $Q = 1, 2, 3$ or cube sizes $K = 1, 3, 5$ for pial surface reconstruction on dHCP, HCP and ADNI datasets. The ASSD and HD are provided in Figure 3.17, which demonstrates that multiple scales and large cube size improve the performance of CortexODE effectively. A larger kernel/cube size K will enlarge the receptive fields, which lead to better performance but also increase the GPU memory requirements. The selection of scale Q depends on the resolution of the input brain MRI image. We can use $Q = 2$ for low-resolution image and $Q = 3$ for higher resolution. Generally, it is sufficient to use $K = 5$ and $Q = 3$ to produce accurate cortical surfaces.

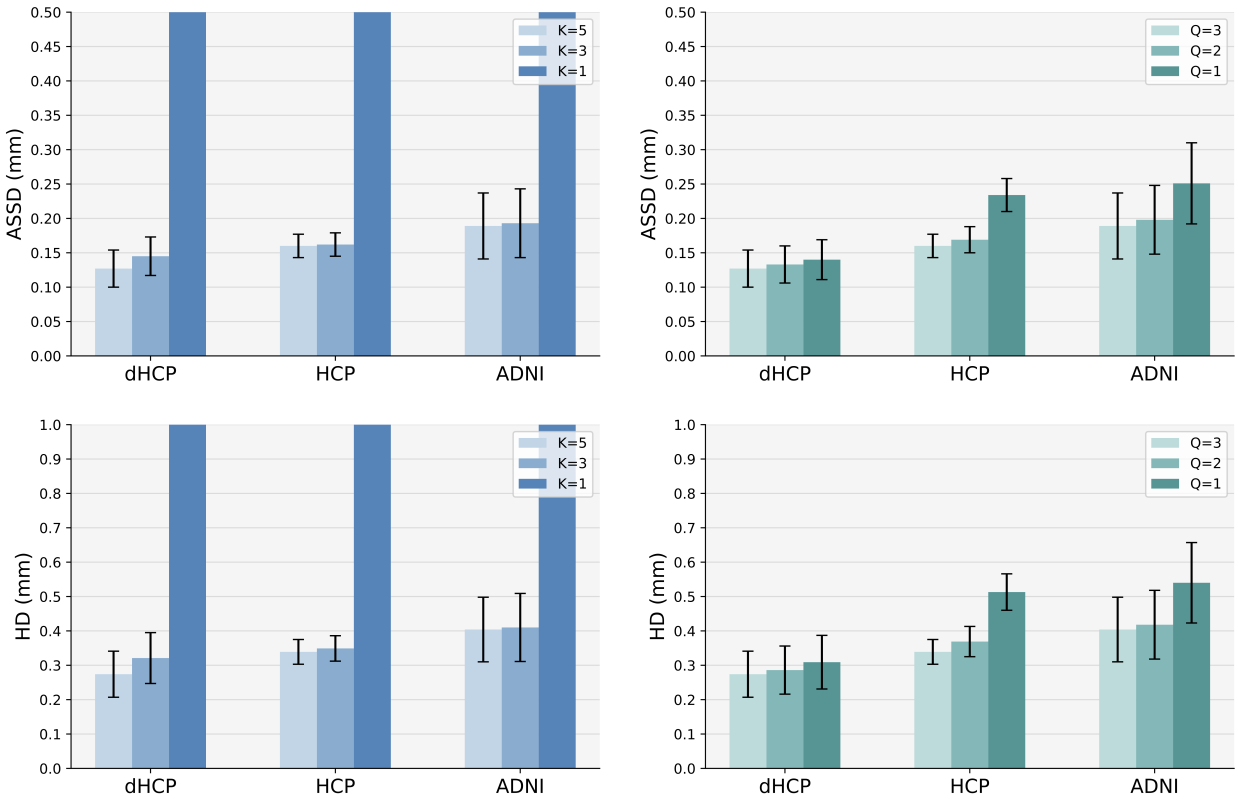


Figure 3.17: Ablation study for CortexODE architecture. The ASSD and HD of the reconstructed pial surfaces are reported for input scales $Q = 1, 2, 3$ and cube sizes $K = 1, 3, 5$.

3.2.5 Discussion

In this section, we introduced CortexODE, a novel geometric deep learning framework for diffeomorphic cortical surface reconstruction. CortexODE extends PialNN (Section 3.1) to tackle both cortical white matter and pial surface reconstruction. Based on the architecture of PialNN, CortexODE employs Neural ODEs to model diffeomorphic flows between surfaces and provides theoretical guarantees to prevent surface self-intersections. An efficient CortexODE-based pipeline is further developed by incorporating WM segmentation, distance transform and fast topology correction. The experiments demonstrate that the proposed pipeline reconstructs high quality cortical surfaces in less than 5 seconds with <0.2mm geometric error and <0.4% self-intersecting faces. CortexODE provides a fast alternative solution for cortical surface reconstruction in large-scale neuroimage studies of various age groups, including neonates, young and old adults. To

further validate the surface quality and reliability of CortexODE, the errors of geodesic distances on cortical surfaces, which can evaluate non-Euclidean surface distortions and identify inaccurate topology correction, could be measured in future work.

3.2.6 Proof of Theorems

Theorem 3.1 (Existence and Uniqueness of Solutions).

Proof. For FC layer f_l and any $x, y \in \mathbb{R}^3$, it holds

$$\|f_l(x) - f_l(y)\|_p \leq L_\phi \|W_l\|_p \|x - y\|_p, \tag{3.16}$$

where $L_\phi = 1$ for ReLU or LeakyReLU activation functions. Based on Equation (3.7) and (3.16), we have

$$\begin{aligned} \|F_\theta(x, U) - F_\theta(y, U)\|_p \leq & \|W_5\|_p \|W_4\|_p \|W_3\|_p \\ & \cdot \left(\|W_1\|_p^p \|x - y\|_p^p + \|W_2\|_p^p \|W_0\|_p^p \|\mathbf{u}(x) - \mathbf{u}(y)\|_p^p \right)^{\frac{1}{p}} \end{aligned} \tag{3.17}$$

By the definition of $\mathbf{u}(x)$ in Equation (3.6),

$$\begin{aligned} \|\mathbf{u}(x) - \mathbf{u}(y)\|_p^p &= \sum_{q=1}^{Q} \sum_{i,j,k=1}^{K} |\mathbf{u}_{qijk}(x) - \mathbf{u}_{qijk}(y)|^p \\ &\leq K^3 \sum_{q=1}^{Q} 2^{p(1-q)} ((U_q)_{\max} - (U_q)_{\min})^p \|x - y\|_p^p \\ &\leq 2K^3 (U_{\max} - U_{\min})^p \|x - y\|_p^p. \end{aligned} \tag{3.18}$$

The first inequality holds because the trilinear interpolation $\mathrm{Lerp}(\cdot)$ is piece-wise linear and continuous. Combining Equation (3.17) and (3.18), we can conclude that F_θ is Lipschitz continuous, *i.e.*,

$$\|F_\theta(x, U) - F_\theta(y, U)\|_p \leq L_F \|x - y\|_p, \tag{3.19}$$

where L_F is defined in Equation (3.8). Since F_θ is the RHS function of ODE (3.5), according to the existence and uniqueness theorem of ODE solutions (Arnold, 1992), there exists a unique solution for the IVP in Equation (3.5). □

Theorem 3.2 (Homeomorphism of ODE Integration).

Proof. Firstly, by the Lipschitz continuity of F_θ, we have

$$\|s_i(x) - s_i(y)\|_p \leq L\|x-y\|_p + hL\sum_{j=1}^{i-1}|\alpha_{i,j}|\|s_j(x) - s_j(y)\|_p. \tag{3.20}$$

Following the proofs in Lebrat et al. (2021), for any $x, y \in \mathbb{R}^3$ it holds

$$\begin{aligned}
\|\mathcal{G}(x) - \mathcal{G}(y)\|_p \geq & \|x-y\|_p - h\sum_{i_0=1}^{r}|\beta_{i_0}|\|s_{i_0}(x) - s_{i_0}(y)\|_p \\
\geq & (1 - hL\sum_{i_0=1}^{r}|\beta_{i_0}|)\|x-y\|_p - h^2L\sum_{i_0=1}^{r}|\beta_{i_0}|\sum_{i_1=1}^{i_0-1}|\alpha_{i_0,i_1}|\|s_{i_1}(x) - s_{i_1}(y)\|_p \\
\geq & (1 - hL\sum_{i_0=1}^{r}|\beta_{i_0}| - h^2L^2\sum_{i_0=1}^{r}|\beta_{i_0}|\sum_{i_1=1}^{i_0-1}|\alpha_{i_0,i_1}|)\|x-y\|_p \\
& - h^3L^2\sum_{i_0=1}^{r}|\beta_{i_0}|\sum_{i_1=1}^{i_0-1}|\alpha_{i_0,i_1}|\sum_{i_2=1}^{i_1-1}|\alpha_{i_1,i_2}|\|s_{i_2}(x) - s_{i_2}(y)\|_p \\
\geq & (1 - \eta(h,L))\,\|x-y\|_p.
\end{aligned} \tag{3.21}$$

Since $\eta(h, L) < 1$, we have $\mathcal{G}(x) \neq \mathcal{G}(y)$ for any $x \neq y$, which means $\mathcal{G}$ is an injection.

To prove $\mathcal{G}$ is a surjection, for any $z \in \mathbb{R}^3$, we define $\mathcal{H}(x) = z - h\sum_{i=1}^{r}\beta_i s_i(x)$. Similar to the procedure in (3.21), for any x and y,

$$\|\mathcal{H}(x) - \mathcal{H}(y)\|_p = \|h\sum\nolimits_{i=1}^{r}\beta_i(s_i(x) - s_i(y))\|_p \leq \eta(h,L)\|x-y\|_p. \tag{3.22}$$

According to the contraction mapping theorem (Ciarlet, 2013), there exists a unique fixed point $\bar{x}$ such that $\mathcal{H}(\bar{x}) = \bar{x}$. Hence, for any z, we can find a unique $\bar{x}$ such that $\mathcal{G}(\bar{x}) = z$,

i.e., $\mathcal{G}$ is a surjection.

Similar to (3.21) and (3.22), $\mathcal{G}$ is Lipschitz continuous with Lipschitz constant $1 + \eta(h, L)$. By replacing x with $\mathcal{G}^{-1}(x)$ in (3.21), we show that $\mathcal{G}^{-1}$ is also Lipschitz continuous:

$$\|\mathcal{G}^{-1}(x) - \mathcal{G}^{-1}(y)\|_p \leq (1 - \eta(h, L))^{-1}\|x - y\|_p. \tag{3.23}$$

Therefore, $\mathcal{G}$ is a homeomorphism. □

Chapter 4

Learning Cortical Surface Reconstruction for Developing Human Brain

The following publications are associated with this chapter:

- **Ma, Q.**, Li, L., Kyriakopoulou, V., Hajnal, J.V., Robinson, E.C., Kainz, B. and Rueckert, D. (2023). Conditional Temporal Attention Networks for Neonatal Cortical Surface Reconstruction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 312-322. Springer. https://doi.org/10.1007/978-3-031-43901-8_30

- **Ma, Q.**, Li, L., Robinson, E.C., Kainz, B. and Rueckert, D. (2024). Weakly Supervised Learning of Cortical Surface Reconstruction from Segmentations. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 766-777. Springer. https://doi.org/10.1007/978-3-031-72120-5_71

Cortical surface reconstruction is crucial in modelling rapid brain development during the perinatal period. In the previous chapter, we proposed explicit geometric deep learning approaches for cortical surface reconstruction. Although these approaches produce high-quality cortical surfaces for adult brain MRI, the anatomical accuracy of reconstructed cortical surfaces needs to be further improved for developing fetal and neonatal brains, which vary considerably across different ages (Bethlehem et al., 2022) and have limited image resolution due to small head size (Orasanu et al., 2014) and fast MRI acquisition (Hughes et al., 2017). Moreover, the supervised learning-based approaches presented in

Chapter 3 heavily rely on the pseudo ground truth (pGT) cortical surfaces for training. Such pGT cortical surfaces are generated by traditional neuroimage processing pipelines (MacDonald et al., 2000; Shattuck and Leahy, 2002; Han et al., 2004; Kim et al., 2005; Fischl, 2012; Glasser et al., 2013; Makropoulos et al., 2018), which can be difficult to generalise well to low-resolution fetal and neonatal brain MRI scans.

To overcome these limitations, in this chapter, we propose a novel deep neural network architecture to learn an age-conditioned time-varying velocity field for diffeomorphic neonatal cortical surface reconstruction. Such a neural network model leverages an attention mechanism to learn coarse-to-fine surface deformations while capturing the substantial variations across newborn infants at different ages. Based on the proposed architecture, we further introduce a weakly supervised learning framework for fetal cortical surface reconstruction, supervised by brain segmentations without the need for pGT cortical surfaces. A novel weakly supervised loss function is designed to alleviate the partial volume effects of low-resolution brain MRI and to encourage the pial surface to deform into deep and challenging cortical sulci. Our framework can extract cortical surfaces with high anatomical fidelity from fetal brain MRI, while traditional neuroimage processing pipelines (Makropoulos et al., 2018) fail to reconstruct accurate cortical surfaces.

4.1 Conditional Temporal Attention Networks for Neonatal Cortical Surface Reconstruction

4.1.1 Introduction

Cortical surface reconstruction is pivotal in modelling and quantifying the brain development in fetal and neonatal neuroimaging studies such as the Developing Human Connectome Project (dHCP) (Makropoulos et al., 2018; Edwards et al., 2022). Extracting cortical surfaces from developing fetal and neonatal brain images is more challenging than adults. Compared to adult brain MRI data (Marcus et al., 2007, 2010; Jack Jr et al., 2008; Van Essen et al., 2013), neonatal brain MRI scans usually have lower image resolution and contrast due to a smaller region of interest (ROI) (Allen et al., 2002; Orasanu et al., 2014; Makropoulos et al., 2016) and the use of fast imaging sequences with sparse acquisition to minimise head motion artefacts of unsedated infants (Hughes et al., 2017; Cordero-Grande et al., 2018). Besides, the rapid growth and continuously increasing complexity of the cortex during the perinatal period lead to considerable variation in shape and scale (Bethlehem et al., 2022) between neonatal cortical surfaces at different post-menstrual ages (PMA) as shown in Figure 4.1. Moreover, since the neonatal head is still small, the cortical sulci of term-born neonates are much narrower than adults as depicted in Figure 4.2. Hence, the neonatal pial surfaces are prone to be affected by partial volume effects (Ballester et al., 2002) and more likely to produce surface self-intersections.

With recent advances of geometric deep leaning (Kipf and Welling, 2016; Groueix et al., 2018; Wang et al., 2018; Wen et al., 2019; Wickramasinghe et al., 2020), an increasing number of fast deep learning-based approaches have been proposed to learn either implicit surface representation (Henschel et al., 2020; Cruz et al., 2021; Gopinath et al., 2021) or explicit mesh deformation (Ma et al., 2021, 2022; Lebrat et al., 2021; Hoopes et al., 2022; Bongratz et al., 2022; Santa Cruz et al., 2022) for cortical surface reconstruction. These

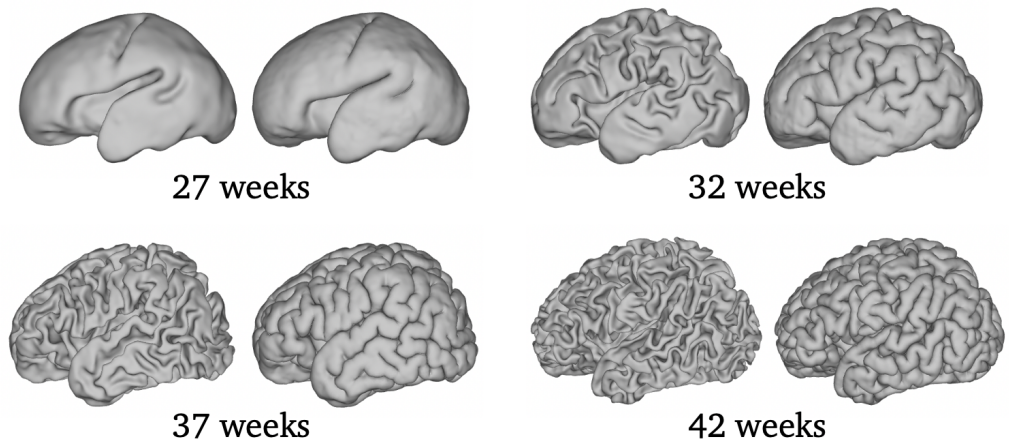


Figure 4.1: Neonatal cortical surfaces at different post-menstrual ages.

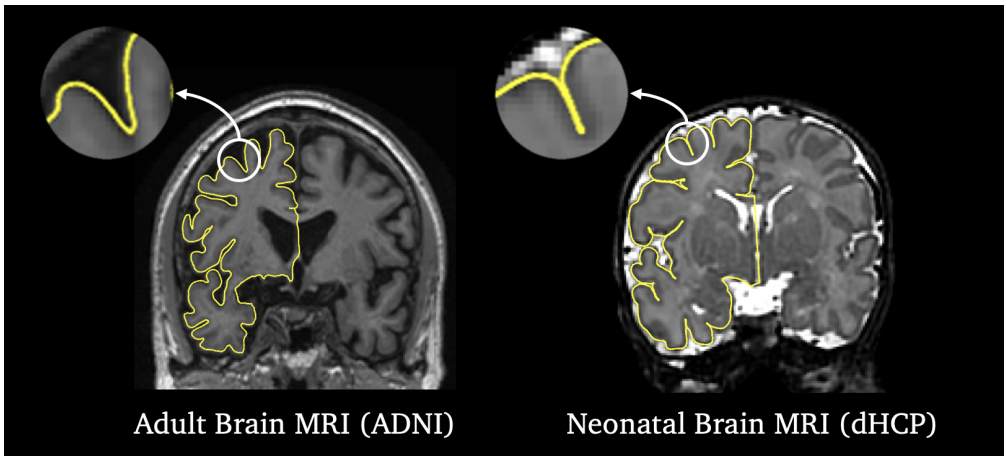


Figure 4.2: Comparison between adult and neonatal brain MRI. Left: adult brain MRI from the ADNI dataset (Jack Jr et al., 2008). Right: neonatal brain MRI from the dHCP dataset (Edwards et al., 2022).

approaches enhance the accuracy and reduce the processing time to a few seconds for a single subject. However, as far as we know, most of existing learning-based approaches so far rely on adult brain MRI dataset for training (Marcus et al., 2007, 2010; Jack Jr et al., 2008; Glasser et al., 2013), while the neonatal cortical surface reconstruction is more challenging due to the limited MRI resolution and large variation across different ages.

Recent studies focus on generating manifold cortical surface meshes and preventing mesh self-intersections by learning diffeomorphic surface deformations (Gupta and Chandraker, 2020; Lebrat et al., 2021; Ma et al., 2022; Santa Cruz et al., 2022; Sun et al., 2022). The basic idea is to model the trajectories of vertices as ordinary differential equations (ODE).

In Section 3.2, we have introduced CortexODE for diffeomorphic cortical surface reconstruction via Neural ODEs (Chen et al., 2018; Gupta and Chandraker, 2020). However, the discretisation of Neural ODEs requires repeated evaluation of a deep neural network at each integration step, leading to a trade-off between computational efficiency and numerical accuracy. Alternatively, it is more efficient to learn flow ODEs induced by stationary velocity fields (SVF), which have been widely adopted in medical image registration (Beg et al., 2005; Ashburner, 2007; Vercauteren et al., 2009; Balakrishnan et al., 2019), to deform an initial mesh template to target surfaces (Lebrat et al., 2021; Santa Cruz et al., 2022). Since a single SVF has limited representation capacity, existing approaches have proposed to train multiple neural networks to predict a sequence of SVFs for coarse-to-fine deformation (Lebrat et al., 2021; Santa Cruz et al., 2022). This improves geometric accuracy but increases the computational burden for both training and inference.

Contributions. We present Conditional Temporal Attention Network (CoTAN) in this section. Instead of learning Neural ODEs following the CortexODE framework (Section 3.2), CoTAN adopts attention mechanism (Hu et al., 2018; Roy et al., 2018) to learn a conditional time-varying velocity field (CTVF) for diffeomorphic neonatal cortical surface reconstruction. Given an input neonatal brain MR image, CoTAN first predicts multiple SVFs at different resolutions. Rather than integrating all SVFs as previous work (Gupta and Chandraker, 2020; Lebrat et al., 2021; Santa Cruz et al., 2022; Sun et al., 2022), CoTAN learns conditional temporal attention maps to attend to specific SVFs for different time steps of integration and PMA of neonates. The CTVF is represented by the weighted sum of learned SVFs, and thus a single CoTAN model is sufficient to model the large deformation and variation of neonatal cortical surfaces. The evaluation on the dHCP neonatal dataset (Edwards et al., 2022) verifies that CoTAN performs better in geometric accuracy, mesh quality and computational efficiency than state-of-the-art methods. The visualisation of attention maps indicates that CoTAN learns coarse-to-fine deformations automatically without intermediate constraints. Our source code is released publicly at `https://github.com/m-qiang/CoTAN`.

4.1.2 Method

4.1.2.1 Diffeomorphic Surface Deformation

We define the diffeomorphic surface deformation $\phi_t : \mathbb{R}^3 \times \mathbb{R} \rightarrow \mathbb{R}^3$ as a flow ordinary differential equation (ODE) following previous work (Beg et al., 2005; Gupta and Chandraker, 2020; Lebrat et al., 2021; Ma et al., 2022; Santa Cruz et al., 2022):

$$\frac{\partial}{\partial t}\phi_t = v_t(\phi_t),\ \phi_0 = Id,\ t \in [0, T], \tag{4.1}$$

where v_t is a time-varying velocity field (TVF) and Id is the identity mapping. Given an initial surface $S_0 \subset \mathbb{R}^3$ with points $x_0 \in S_0$, we define $x(t) := \phi_t(x_0)$ as the trajectories of the points on the deformable surface $S_t = \phi_t(S_0)$ for $t \in [0, T]$. Then the flow equation (4.1) can be rewritten as $\frac{d}{dt}x(t) = v_t(x(t))$ with initial value $x(0) = x_0$. By the existence and uniqueness theorem for ODE solutions (Coddington et al., 1956; Walter, 2013), if $v_t(x)$ is Lipschitz continuous with respect to x, the trajectories $x(t)$ will not intersect with each other, so that the surface self-intersections can be prevented effectively. By integrating the ODE (4.1), we obtain a diffeomorphism ϕ_T that deforms S_0 to a manifold S_T, of which the points are $x(T) = \phi_T(x_0) = x_0 + \int_0^T v_t(x(t))dt$.

4.1.2.2 Conditional Temporal Attention Network

An overview of the architecture of the conditional temporal attention network (CoTAN) is depicted in Figure 4.3. CoTAN first predicts multiple SVFs given a 3D brain MRI volume. A 3D U-Net (Ronneberger et al., 2015) is used to extract feature maps with R resolution levels, each of which scales the input size by the factor of 2^{r-R} for $r = 1, ..., R$. Then we upsample the multi-scale feature maps and learn M volumetric SVFs for each resolution, in order to enhance the representation ability. Let $\mathbf{V}$ denote all $R \times M$ discrete SVFs. The continuous multi-resolution SVFs $\mathbf{v} : \mathbb{R}^3 \rightarrow \mathbb{R}^{R \times M \times 3}$ can be obtained by $\mathbf{v}(x) =$

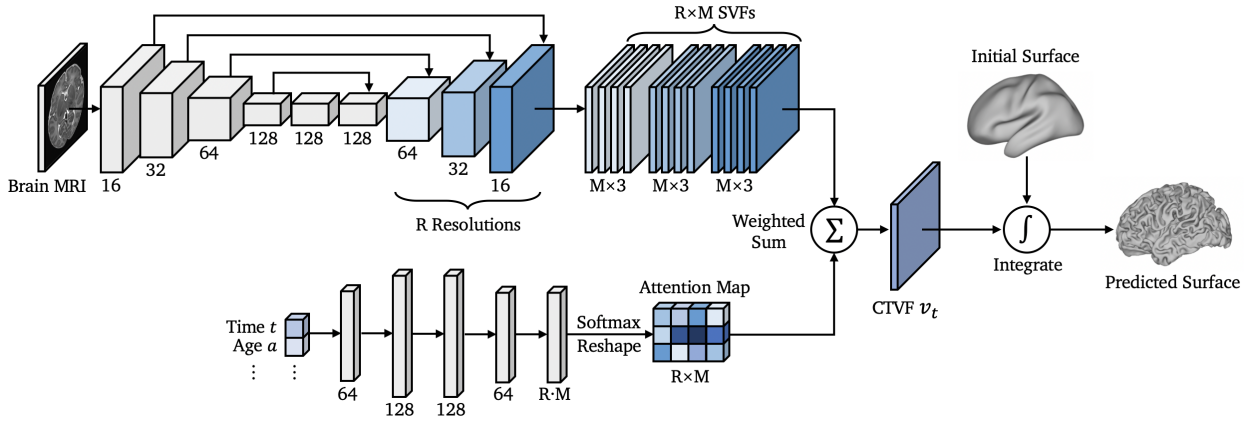


Figure 4.3: The architecture of the proposed CoTAN framework. Given an input 3D brain MRI, CoTAN uses a U-Net to predict M SVFs for each resolution level R. An attention map is learned to focus on specific SVFs varying with the input integration time t and conditioned on the age a of the neonatal subjects. Further conditioning could be achieved with minimal effort, *e.g.*, biological sex, diagnoses, etc. For each time step t, the CTVF v_t is represented by the weighted sum of all $R \times M$ SVFs. By integrating the CTVF, CoTAN deforms an input initial surface to the predicted cortical surface.

$\mathrm{Lerp}(x, \mathbf{V})$, where $\mathrm{Lerp}(\cdot)$ is the trilinear interpolation function. Each element $\mathbf{v}^{r,m} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is an SVF for $r = 1, ..., R$ and $m = 1, ..., M$. Note that $\mathbf{v}(x)$ is Lipschitz continuous since $\mathrm{Lerp}(\cdot)$ is continuous and piece-wise linear.

CoTAN adopts a channel-wise attention mechanism (Hu et al., 2018; Roy et al., 2018) to focus on specific SVFs since it is time-consuming to integrate all $R \times M$ SVFs (Gupta and Chandraker, 2020; Lebrat et al., 2021; Santa Cruz et al., 2022). The attention is conditioned on both integration time $t \in [0, T]$ and information about the subject. To model the high variation between infant brains, we consider the post-menstrual ages (PMA) $a \in \mathbb{R}$ of the neonates at scan time as the conditioning variable in this work. Note that we do not use a self-attention module (Vaswani et al., 2017; Dosovitskiy et al., 2021) to learn key and query pairs. Instead, we learn a probability attention map to measure the importance of each SVF. More precisely, as shown in Figure 4.3, we use a fully connected network (FCN) to encode the input time t and PMA a into a $(R \cdot M) \times 1$ feature vector. After reshaping and softmax activation, the FCN learns conditional temporal attention maps $\mathbf{p}(t, a) \in \mathbb{R}^{R \times M}$ which satisfy $\sum_{r=1}^{R} \sum_{m=1}^{M} \mathbf{p}^{r,m}(t, a) = 1$ for any t and a. Then a *conditional time-varying*

velocity field (CTVF) is predicted by computing the weighted sum of all SVFs:

$$v_t(x;a) = \sum_{r=1}^{R} \sum_{m=1}^{M} \mathbf{p}^{r,m}(t,a) \cdot \mathbf{v}^{r,m}(x). \tag{4.2}$$

The CTVF is adaptive to the integration time and the age of subjects, which can handle the large deformation and variation for neonatal cortical surfaces. Such an attention mechanism encourages CoTAN to learn coarse-to-fine surface deformation by attending to SVFs at different resolutions.

To deform the initial surface S_0 to the target surface, we integrate the flow ODE (4.1) with the CTVF through the forward Euler method. For $k = 0, ..., K-1$, the surface points are updated by $x_{k+1} = x_k + hv_k(x_k;a)$, where K is the total integration steps and $h = T/K$ is the step size with $T = 1$. For each step k, we only need to recompute the attention maps $\mathbf{p}(hk,a)$ and update the CTVF $v_k(x_k;a)$ accordingly by Equation (4.2). Compared to CortexODE (Section 3.2) that requires repeated evaluation of the deformation network, CoTAN only evaluates the 3D U-Net once during the integration, which saves considerable runtime. Instead of integrating multiple SVFs directly (Lebrat et al., 2021; Santa Cruz et al., 2022), CoTAN only integrates a single CTVF with higher computational efficiency.

4.1.2.3 Neonatal Cortical Surface Reconstruction

We train two CoTAN models on the dHCP neonatal dataset (Edwards et al., 2022) to 1) deform an initial surface into a white matter surface and to 2) expand the white matter surface into a pial surface as illustrated in Figure 4.4. We use the same initial surface (leftmost in Figure 4.4) for all subjects, which is created by iteratively applying Laplacian smoothing to a Conte-69 surface atlas (Van Essen et al., 2012). CoTAN does not need time-consuming topology correction if the initial surface has spherical topology. This is different from CotexODE, which creates subject-specific initial surfaces based on topology-corrected signed distance maps. We generate pseudo ground truth (pGT) sur-

faces by the dHCP structural neonatal pipeline (Makropoulos et al., 2018), which has been fully validated through quality control performed by clinical experts.

For white matter surface reconstruction, we consider loss functions that have been widely used in previous work (Fan et al., 2017; Wang et al., 2018; Wickramasinghe et al., 2020; Bongratz et al., 2022): the Chamfer distance loss $\mathcal{L}_{cd}$, the mesh Laplacian loss $\mathcal{L}_{lap}$, and the normal consistency loss $\mathcal{L}_{nc}$. The bidirectional Chamfer distance computes the distance between two point clouds $\mathcal{X}$ and $\mathcal{Y}$:

$$\mathcal{L}_{cd}(\mathcal{X}, \mathcal{Y}) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \min_{y \in \mathcal{Y}} \|x - y\|^2 + \frac{1}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} \|x - y\|^2. \tag{4.3}$$

Given a 3D triangular mesh $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ with vertices $\mathcal{V}$, edges $\mathcal{E}$ and faces $\mathcal{F}$, the mesh Laplacian loss regularises the smoothness of the mesh:

$$\mathcal{L}_{lap}(\mathcal{M}) = \frac{1}{|\mathcal{V}|} \sum_{i=1}^{|\mathcal{V}|} \left\| x_i - \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} x_j \right\|^2, \tag{4.4}$$

where x_i is the i-th vertex, $|\mathcal{V}|$ is the number of vertices, and $\mathcal{N}(i)$ includes its adjacent vertices. The normal consistency loss is defined as:

$$\mathcal{L}_{nc}(\mathcal{M}) = \frac{1}{|\mathcal{E}|} \sum_{f_i \cap f_j \in \mathcal{E}} \left(1 - \langle n_i, n_j \rangle\right), \tag{4.5}$$

which constrains the dot product between the unit normals n_i, n_j of any pairs of faces $f_i, f_j \in \mathcal{F}$ that share the same edge $f_i \cap f_j = e \in \mathcal{E}$. The final loss is weighted by $\mathcal{L} = \mathcal{L}_{cd} + \lambda_{lap}\mathcal{L}_{lap} + \lambda_{nc}\mathcal{L}_{nc}$. We train CoTAN in two steps for white matter surface extraction. First, we pre-train the model using relatively large weights λ_{lap} and λ_{nc} for regularisation. The Chamfer distance is computed between the vertices of the predicted and pGT surfaces. These ensure that the initial surface can be deformed robustly during training. Then, we fine-tune CoTAN using small weights to increase geometric accuracy. The distances are computed between 150k uniformly sampled points on the surfaces.

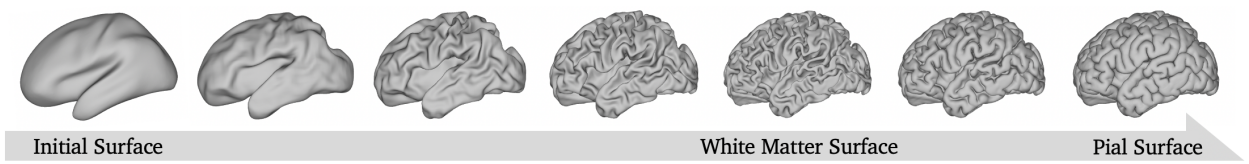


Figure 4.4: Diffeomorphic deformation from an initial template mesh to cortical surfaces.

For pial surface reconstruction, we follow Chapter 3 (Ma et al., 2021, 2022) and use the pGT white matter surfaces as the input for training. Then the mean squared error (MSE) loss $\mathcal{L}_{mse} = \frac{1}{N}\sum_{i=1}^{N} \|\hat{x}_i - x_i^*\|^2$ can be computed between the vertices of predicted and pGT pial meshes, where N is the number of vertices. No point matching is required since the pGT white matter and pial surfaces have the same mesh connectivity. Therefore, the MSE loss provides stronger supervision than the Chamfer distance, while the latter is prone to mismatching the points in narrow sulci, resulting in mesh self-intersections.

4.1.3 Experiments

4.1.3.1 Implementation Details

We evaluate CoTAN on the third release of dHCP neonatal dataset (Edwards et al., 2022), which includes 877 T2-weighted (T2w) brain MRI scanned from newborn infants at PMA between 27 to 45 weeks. The MRI images are affinely aligned to the MNI152 space and clipped to the size of $112 \times 224 \times 160$ for each brain hemisphere. The dataset is split into 60/10/30% for training/validation/testing. For the CoTAN model, we set the resolution $R = 3$ and the number of SVFs $M = 4$ for each resolution. For integration, we set the total number of steps to $K = 50$ with step size $h = 0.02$. We re-mesh the initial mesh to 140k vertices, of which the coordinates are normalised to $[-1, 1]$. We use the Adam optimiser for training. For the white matter surface, we first pre-train CoTAN for 100 epochs using a learning rate of $\gamma = 10^{-4}$ and weights $\lambda_{lap} = 0.5$, $\lambda_{nc} = 5 \times 10^{-4}$ for the loss function. Then we fine-tune for 100 epochs using smaller weights $\lambda_{lap} = 0.1$ and $\lambda_{nc} = 10^{-4}$ with $\gamma = 2 \times 10^{-5}$. For the pial surface, we set the maximum channel size of CoTAN as 32

Table 4.1: Comparative results of neonatal cortical surface reconstruction on the dHCP dataset. The geometric accuracy (ASSD and HD90) and mesh quality (the ratio of SIFs) are reported for white matter and pial surfaces. Smaller values mean better results. *CoTAN (ours) achieves significant improvement ($p < 0.05$) compared to baselines.

Method	White Matter Surface			Pial Surface		
	ASSD (mm)	HD90 (mm)	SIF (%)	ASSD (mm)	HD90 (mm)	SIF (%)
CoTAN	**0.107±0.026**	**0.217±0.076**	**0.001±0.004**	**0.121±0.029**	**0.259±0.075**	**0.071±0.034**
CortexODE	0.109±0.052	0.231±0.326	**0.001±0.002**	0.134±0.052*	0.306±0.358*	0.221±0.114*
CFPP	0.118±0.028*	0.241±0.085*	0.075±0.057*	0.124±0.031*	0.273±0.086*	2.457±1.003*
CorticalFlow	0.122±0.029*	0.247±0.080*	0.048±0.032*	0.157±0.031*	0.331±0.089*	9.798±2.902*
Vox2Cortex	0.115±0.035*	0.233±0.110*	0.253±0.169*	0.130±0.039*	0.291±0.141*	14.366±2.262*
DeepCSR	0.129±0.047*	0.276±0.211*	–	0.299±0.070*	1.214±0.337*	–

to avoid overfitting and train for 200 epochs with $\gamma = 10^{-4}$. We only consider left brain hemisphere in the experiments. All experiments are conducted on a NVIDIA GeForce RTX 3080 GPU with 12GB memory.

4.1.3.2 Comparative Results

We compare the performance of CoTAN with existing learning-based cortical surface extraction approaches including CortexODE (Ma et al., 2022), CorticalFlow++ (CFPP) (Santa Cruz et al., 2022), CorticalFlow (Lebrat et al., 2021), Vox2Cortex (Bongratz et al., 2022) and DeepCSR (Cruz et al., 2021). We employ the fast topology correction (Ma et al., 2022) for DeepCSR. CoTAN can guarantee spherical topology, and the Euler characteristic number is $\chi = 2$ for all predicted surfaces.

Geometric Accuracy. We measure the geometric accuracy by commonly used metrics (Cruz et al., 2021; Bongratz et al., 2022; Ma et al., 2022): average symmetric surface distance (ASSD) and 90th percentile of Hausdorff distance (HD90). The distances are computed between uniformly sampled 100k points on the predicted and pGT surfaces. For fair comparison, the predicted cortical meshes have around 140k vertices for all baseline approaches. We conduct a paired t-test to examine the statistical significance. As reported in Table 4.1, CoTAN achieves significantly superior geometric accuracy ($p < 0.05$) compared to all state-of-the-art baselines, except for white matter surfaces of CortexODE.

Mesh Quality. We evaluate the mesh quality by the ratio of self-intersecting faces (SIF) as compared in Table 4.1. Note that due to the narrower sulcal gaps of neonatal cortex compared to adult (Jack Jr et al., 2008; Marcus et al., 2007) (see Figure 4.2), all baseline methods produce more SIFs than reported in their original papers. Apart from DeepCSR that produces no SIFs since it uses the Marching Cubes algorithm (Lorensen and Cline, 1998), CoTAN achieves the best mesh quality with only 0.001% SIFs in white matter surfaces and 0.071% SIFs in pial surfaces. These remaining SIFs are likely introduced by the discretisation of triangular mesh representation and ODE integration. Compared to CortexODE, CoTAN integrates a CTVF efficiently using a smaller step size with $K = 50$ integration steps, reducing numerical errors significantly with fewer SIFs.

CoTAN produces fewer SIFs for three reasons. Firstly, CoTAN employs diffeomorphic deformation, while the non-diffeomorphic Vox2Cortex creates 14% SIFs. We further set the integration steps $K = 5$ for CoTAN such that the deformation is no longer diffeomorphic. The SIFs of pial surfaces are increased to 2.99$\pm$1.19%. Secondly, CoTAN reconstructs the pial surface by expanding the input white matter surface. It is difficult to avoid collisions during the deformation from a smooth template into deep sulci, *e.g.*, in Vox2Cortex and CorticalFlow. Lastly, CoTAN uses MSE loss rather than Chamfer distance for pial surface extraction, which alleviates the mismatch between points in the narrow sulci.

Computational Efficiency. We report the runtime and GPU memory cost for both training and testing, as well as the number of learnable parameters for CoTAN and all baseline approaches in Table 4.2. The runtime includes both model inference and post-processing. CoTAN only requires 0.21 seconds to extract cortical surfaces for each hemisphere. This is 2$\times$ faster than the best baseline and 9$\times$ faster than CortexODE, which consists of computationally intensive topology correction as well as iterative neural network evaluation for Neural ODE integration. CoTAN is also orders of magnitude faster compared to the dHCP neonatal pipeline (Makropoulos et al., 2018), which requires more than 6 hours to process a single subject. CoTAN can be trained end-to-end and reduces training time by

Table 4.2: Comparative results of runtime, GPU memory cost, and the number of model parameters for both training and testing.

Method	Runtime		GPU (GB)		Model
	Train	Test	Train	Test	#Param
CoTAN (Ours)	57.9h	**0.21**s	8.71	4.05	2.47M
CortexODE	51.6h	1.88s	9.24	4.61	1.99M
CFPP	131.5h	0.57s	9.86	3.56	1.03M
CorticalFlow	105.7h	0.51s	9.12	3.56	1.03M
Vox2Cortex	63.7h	1.48s	6.96	6.26	6.54M
DeepCSR	**15.1**h	10.69s	**5.26**	**2.33**	4.65M

Table 4.3: The results of ablation experiments for CoTAN on white matter surface reconstruction.

Method	ASSD (mm)	HD90 (mm)
CoTAN (Ours)	**0.107±0.026**	**0.217±0.076**
Pre-train	0.116±0.029	0.238±0.095
$R = 1$	0.112±0.027	0.232±0.081
$M = 1$	0.111±0.030	0.231±0.097
SVF ($a = t = 0$)	0.138±0.037	0.291±0.109
CVF ($t = 0$)	0.135±0.036	0.285±0.111
TVF ($a = 0$)	0.108±0.028	0.222±0.090

46% compared to CorticalFlow and CFPP, which have to train three U-Nets consecutively to parameterise three SVFs for a single surface. Although CoTAN uses relatively more parameters to learn surface deformations, it is still memory efficient which only costs 8.7GB and 4GB GPU memory for training and testing.

Qualitative Comparison. One limitation of our experiments is that we only train and evaluate CoTAN based on the pGT cortical surfaces generated by the dHCP structural pipeline (Makropoulos et al., 2018). Previous approaches (Cruz et al., 2021; Ma et al., 2022; Bongratz et al., 2022) have been validated by test-retest experiments (Maclaren et al., 2014). However, this is infeasible for neonatal subjects, of which the brains develop rapidly even within a short period. To verify the superior anatomical accuracy of CoTAN, we provide qualitative comparison between the dHCP pipeline and CoTAN in Figure 4.5. It demonstrates that CoTAN can effectively mitigate corruptions introduced by the dHCP pipeline in cortical surfaces of neonatal subjects at different ages.

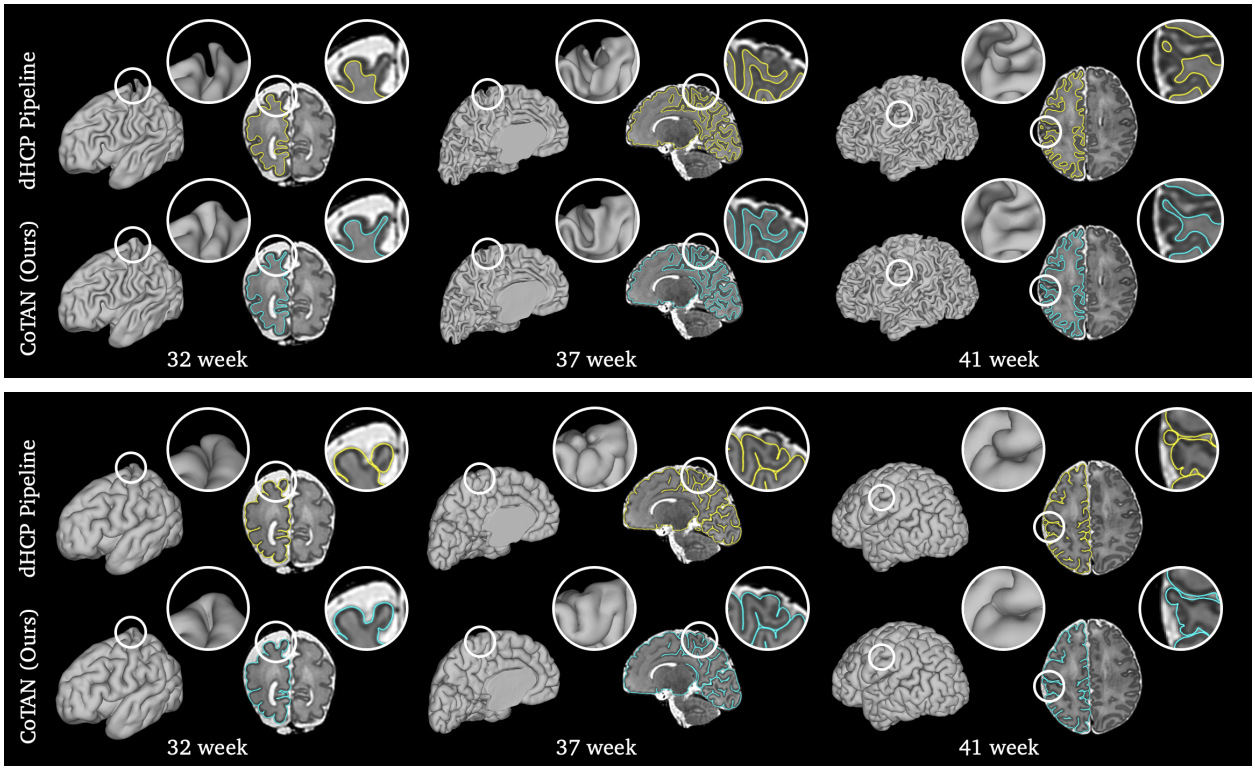


Figure 4.5: Visualisation of neonatal cortical surfaces generated by the dHCP structural pipeline (Makropoulos et al., 2018) and CoTAN for different ages. CoTAN shows better anatomical accuracy as highlighted in both surface meshes and brain MRI images. Top: white matter surfaces. Bottom: pial surfaces.

4.1.3.3 Ablation Study

We conduct ablation studies on CoTAN and evaluate the geometric accuracy on the white matter surface reconstruction. First, without fine-tuning, the geometric errors increase by 8% as reported in Table 4.3. Then, we consider CoTAN with single resolution ($R = 1$) or only predict a single SVF ($M = 1$) for each resolution. The geometric distances increase in both cases. Next, we examine the effectiveness of the CTVF $v_t(x; a)$ by fixing the input time $t = 0$ or age $a = 0$. We train CoTAN models to predict the TVF ($a = 0$), CVF ($t = 0$) and SVF ($a = t = 0$), which are degraded from the CTVF. Table 4.3 shows that the SVF increases geometric errors by 30% due to its limited representation capacity. The CVF increases accuracy slightly by learning conditional deformations adaptive to the age of neonates. The TVF exploits temporal information to model a wider range of deformations and effectively improves the performance. Lastly, we train a U-Net to

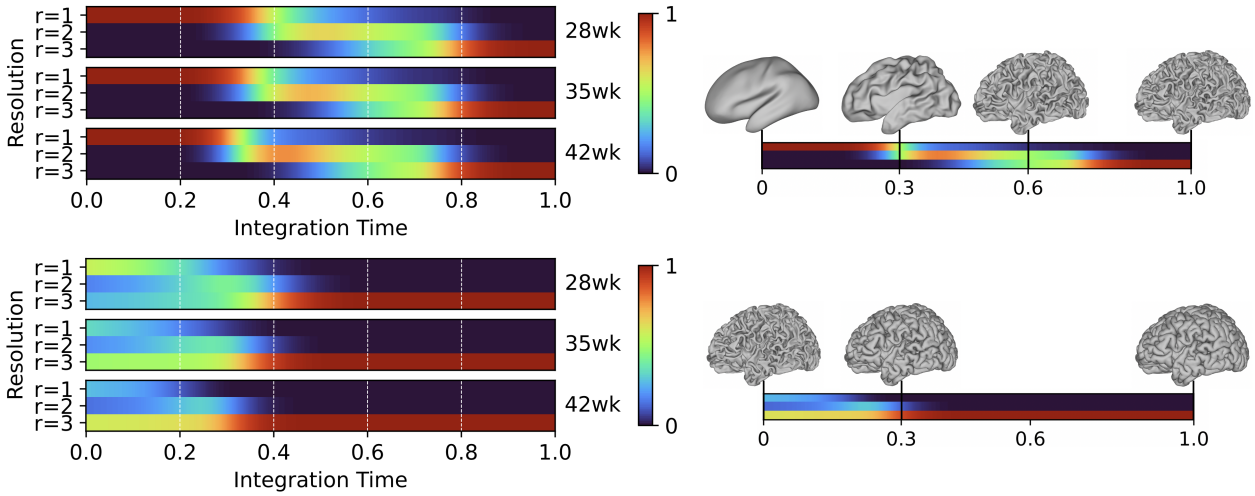


Figure 4.6: Visualisation of attention maps predicted by CoTAN for white matter surface (top) and pial surface (bottom) reconstruction. Left: conditional temporal attention maps $\mathbf{p}^r(t, a)$ at different time and ages. Right: coarse-to-fine diffeomorphic cortical surface deformations learned by CoTAN.

predict $R \times M$ SVFs and integrate them directly without attention. Since the gradients are backpropagated through all SVFs, it requires >140 hours training time which is 2.4× slower than our attention-based CoTAN. The model is also sensitive to small updates, which can affect all SVFs. This results in exploding gradients which we have observed in the training, whereas CoTAN can be trained robustly by integrating a single CTVF.

4.1.3.4 Attention Maps

We explore the attention maps $\mathbf{p}^{r,m}(t, a)$ learned by CoTAN. We define $\mathbf{p}^r = \sum_{m=1}^{M} \mathbf{p}^{r,m}$ to reflect the importance of the SVFs at each resolution level $r = 1, ..., R$, where $R = 3$ and larger r means higher resolution. Figure 4.6 visualises the attention maps $\mathbf{p}^r(t, a)$ for cortical surface reconstruction with integration time $t \in [0, 1]$ and age $a \in \{28, 35, 42\}$. It shows that CoTAN pays more attention to low resolution for younger subjects (28 week) whose brains have not fully developed yet. More deformations at higher resolutions ($r \geq 2$) are required for older neonates (≥ 35 week) with highly folded cortex

Additionally, Figure 4.6 manifests that the attention maps focus on low resolution ($r = 1$) at the beginning of the integration, and then attends to high resolution ($r = 3$) when $t \to 1$.

In particular, for white matter surface reconstruction, Figure 4.6 (top-right) shows that an initial white matter surface deforms into a coarse shape for $t \leq 0.3$ and learns fine details for $t \geq 0.6$. Figure 4.7 further displays the magnitude $\|v_t\|$ of the CTVF for $t \in [0, 1]$ and illustrates that the CVTF captures large and coarse deformations at the beginning, and then model small and fine deformations as time evolves. These demonstrate that CoTAN learns coarse-to-fine deformations automatically without any supervision on the intermediate deformations.

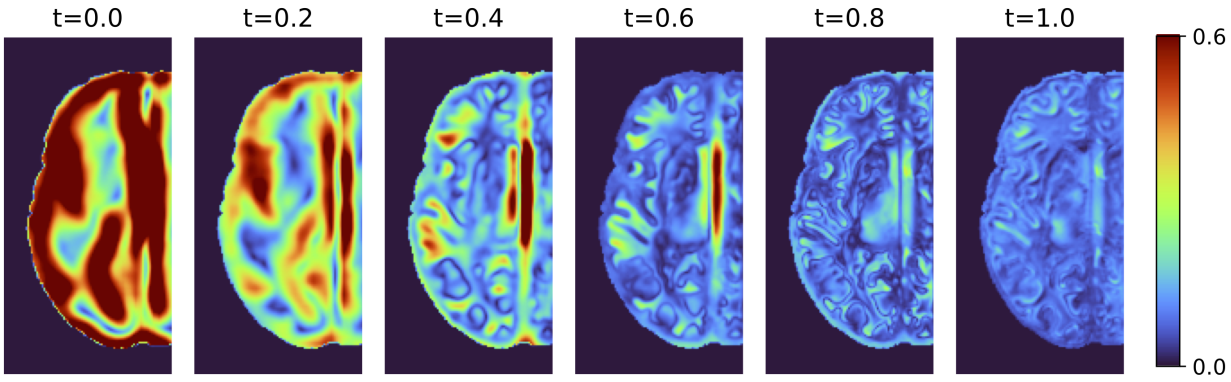


Figure 4.7: Visualisation of the CTVF v_t for white matter surface reconstruction with integration time $t \in [0, 1]$. The colour map reflects the magnitude $\|v_t\|$ of the CTVF.

4.1.4 Discussion

In this section, we propose CoTAN for diffeomorphic neonatal cortical surface reconstruction. CoTAN employs an attention mechanism to combine multiple SVFs to a CTVF, which is integrated to predict coarse-to-fine surface deformations, conditioned on the PMA ages of neonatal subjects. CoTAN outperforms existing deep learning baselines in geometric accuracy, computational efficiency, and mesh quality on the dHCP neonatal datasets. The architecture of CoTAN can be further extended and applied to extract cortical surfaces conditioned on both demographic and pathological information of subjects. Although CoTAN is customised for neonatal cortical surface reconstruction, it can be generalised seamlessly to extract cortical surfaces from adult brain MRI.

The accuracy and efficiency of CoTAN make it superior than PialNN and CortexODE frameworks presented in Chapter 3, except for processing brain MRI at very high image resolution. In this case, CortexODE is preferable as it only captures local features from brain MRI intensity. Similar to PialNN and CortexODE, the training of CoTAN models still heavily relies on the pGT cortical surfaces generated by traditional cortical surface reconstruction pipelines (Fischl, 2012; Glasser et al., 2013; Makropoulos et al., 2018). Thus, CoTAN may fail to be trained on low-resolution brain MRI data for which traditional pipelines are unable to provide acceptable pGT cortical surfaces. In next section, we will introduce a weakly supervised learning framework for cortical surface reconstruction to reduce the dependence on pGT cortical surfaces.

4.2 Weakly Supervised Fetal Cortical Surface Reconstruction Guided by Segmentations

4.2.1 Introduction

Cortical surfaces play an essential role in visualising the anatomical structure as well as quantitative characterising the morphology of the cortex. Traditional neuroimage processing pipelines (MacDonald et al., 2000; Han et al., 2004; Fischl, 2012; Dai et al., 2013; Glasser et al., 2013; Makropoulos et al., 2018) achieved great success for cortical surface reconstruction from adult or neonatal brain MRI. However, these pipelines normally incorporate multiple processing steps and require more than 6 hours for a single subject. The limited accuracy of prior processing steps will also cause subsequent corruptions in cortical surfaces. Moreover, traditional pipelines contain a large amount of carefully tuned parameters, which make them difficult to generalise across data domains in different age groups or acquisition protocols.

To boost the accuracy and efficiency of cortical surface reconstruction, recent studies (Cruz et al., 2021; Gopinath et al., 2021; Henschel et al., 2020; Wang et al., 2023; Gopinath et al., 2023b) leverage deep learning-based approaches to predict implicit surface representations, *i.e.*, binary segmentations and signed distance functions (SDF). Explicit 3D meshes are extracted from the implicit surfaces by Marching Cubes algorithm (Lorensen and Cline, 1998). Topology correction algorithms (Fischl et al., 2001; Bazin and Pham, 2005, 2007; Ségonne et al., 2007) are used to detect and repair topological errors such that the extracted surfaces are homeomorphic to a 2-sphere. Although existing approaches are capable of producing accurate segmentations (Roy et al., 2019; Henschel et al., 2020; Billot et al., 2023a; Uus et al., 2023a; Wang et al., 2023) or signed distance functions (SDF) (Cruz et al., 2021; Gopinath et al., 2021, 2023b), these volumetric representations intrinsically face the challenges of partial volume effects (Ballester et al., 2002), leading to difficulties

for the Marching Cubes algorithm in accurately capturing the cortical folding of extracted surfaces. This issue is especially pronounced within the cortical sulci, in particular for low-resolution brain MRI such as fetal and neonatal data.

Instead of learning implicit surfaces, latest works (Lebrat et al., 2021; Ma et al., 2021, 2022, 2023; Bongratz et al., 2022, 2024; Hoopes et al., 2022; Santa Cruz et al., 2022; Chen et al., 2023; Zheng et al., 2024) focus on learning explicit surface deformations end-to-end from brain MRI, which only require a few seconds to deform an initial mesh to target cortical surfaces. However, one major limitation of existing deep learning-based cortical surface reconstruction approaches, including the PialNN, CortexODE, and CoTAN frameworks presented in previous sections, is that they heavily rely on the supervision of pseudo ground truth (pGT) cortical surfaces generated by traditional pipelines (Fischl, 2012; Glasser et al., 2013; Makropoulos et al., 2018). Their long processing time makes it expensive to collect a large dataset for training. Also, errors in pGT surfaces inevitably form an upper bound for the achievable accuracy. Moreover, traditional pipelines may fail to extract pGT surfaces from fetal or clinical MRI sequences, which are inherently limited in scan time, thereby resulting in lower image resolution.

Contributions. We present CoSeg, a weakly supervised cortical surface reconstruction framework without the need for pGT surfaces during the training phase. CoSeg learns cortical surfaces explicitly with the weak supervision of cortical ribbon segmentations, which are more accessible than pGT cortical surfaces particularly for low-resolution images (Billot et al., 2023a; Uus et al., 2023a). We adapt the architecture of CoTAN introduced in Section 4.1 to a temporal attention network (TA-Net) to learn diffeomorphic surface deformations from brain MRI at different age groups. To address partial volume effects on the pial surfaces, we propose a novel weakly supervised loss consisting of a boundary loss, which inflates the white matter surface towards the boundary of the cortical grey matter (cGM) segmentation, as well as an inflation loss, which constraints the inflation to follow the normal direction. We evaluate CoSeg on the HCP young adult dataset (Van Es-

sen et al., 2013) and the dHCP fetal dataset (Price et al., 2019). The codes for CoSeg are released publicly at https://github.com/m-qiang/CoSeg.

4.2.2 Method

4.2.2.1 Temporal Attention Networks

Given an initial surface $S_0 \subset \mathbb{R}^3$ with points $x_0 \in S_0$, a diffeomorphic surface deformation can be defined via an ordinary differential equation (ODE) (Beg et al., 2005):

$$\frac{d}{dt}x(t) = v(x(t), t),\ x(0) = x_0,\ t \in [0, T], \tag{4.6}$$

where $x(t)$ is the deformed surface point, and $v(x, t) \in \mathbb{R}^3$ is a learnable time-varying velocity field (TVF). By integrating the flow ODE (4.6), the initial surface S_0 is deformed to a target surface S_T with the points $x_T = x(T)$. Since our CoSeg framework is model agnostic, we can employ arbitrary deep neural network models, such as CortexODE and CoTAN, to parameterise the TVF $v(x, t)$. In this section, we simplify the CoTAN architecture to a temporal attention network (TA-Net) to learn a TVF. While CoTAN is conditioned on the ages of the subjects, the TA-Net is only conditioned on the integration time t such that it can be extended to adult data. As depicted in Figure 4.8, given an input brain MRI, TA-Net uses a 3D U-Net (Ronneberger et al., 2015) to learn $M = 2$ stationary velocity fields (SVF) for each of $R = 3$ resolution levels. Thus, there are total $R \times M$ SVFs denoted by $\mathbf{v}(x) \in \mathbb{R}^{RM\times 3}$. Given an input integration time $t \in [0, T]$, a channel-wise time-varying attention map $\mathbf{p}(t) \in \mathbb{R}^{RM}$, which satisfies $\sum_{i=1}^{RM} \mathbf{p}_i(t) = 1$, is predicted by a fully connected network to measure the importance of all $R\times M$ SVFs. By computing the weighted sum over all SVFs, the TVF is represented as $v(x, t) = \mathbf{v}(x)^\top \mathbf{p}(t)$.

We discretise a surface S by a 3D mesh $\mathcal{M} = (\mathcal{V}, \mathcal{F}, \mathcal{E})$, where $\mathcal{V}, \mathcal{F}, \mathcal{E}$ are the sets of vertices, faces, and edges. Then, we integrate the ODE (4.6) with the forward Euler

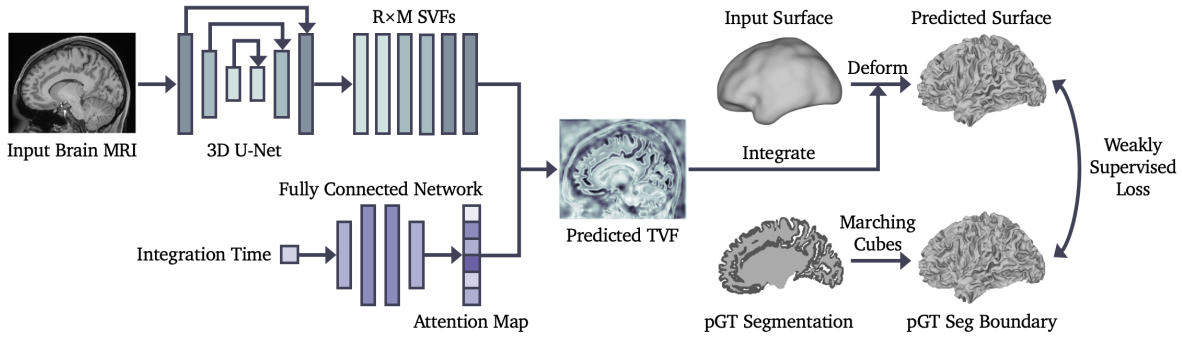


Figure 4.8: The architecture of the TA-Net. TA-Net predicts a TVF by learning a weighted sum of multiple SVFs. The TVF is integrated to produce diffeomorphic surface deformations which fit an initial input surface to the predicted cortical surface.

method. For each integration step $k = 0, ..., K-1$, the vertices are updated by $x_{k+1} = x_k + hv(x_k, hk)$, where $h = T/K$ is the step size and $x_0 \in \mathcal{V}_0$ is the vertex of an input mesh $\mathcal{M}_0$. By integration, the TA-Net deforms the input mesh $\mathcal{M}_0$ to a predicted cortical surface mesh $\widehat{\mathcal{M}}$ with vertices $\hat{x} = x_K$. The TA-Net can be trained by minimising a weakly supervised loss between the predicted mesh $\widehat{\mathcal{M}}$ and the boundary of pGT segmentation.

4.2.2.2 Initial Surface Generation

We affinely align all brain MRI and pGT cortical ribbon segmentations to a standard space such as MNI152 (Fonov et al., 2011). A fixed initial surface for all subjects is extracted from a template ribbon segmentation, which is obtained from either an atlas or the group average of all pGT segmentations in the training set. Inspired by Ma et al. (2022) and Santa Cruz et al. (2022), we use a distance transform algorithm (Butt and Maragos, 1998) to convert the template white matter (WM) segmentation to a SDF, where the interior/exterior of the segmentation has negative/positive value. The SDF is smoothed by a Gaussian filter with standard deviation $\sigma = 6.0$. An initial mesh $\mathcal{M}_0$ is extracted by Marching Cubes (Lorensen and Cline, 1998) from the SDF at level $l = 1.5$. Such an initial mesh has genus-0 topology since the implicit surface has been sufficiently smoothed and inflated. We iteratively apply Laplacian smoothing and isotropic explicit remeshing (Cignoni et al., 2008) to the initial mesh $\mathcal{M}_0$ until it has the desired number of vertices.

4.2.2.3 White Matter Surface Reconstruction

CoSeg learns cortical surfaces weakly supervised by cortical ribbon segmentations, which contain the label maps of WM and cGM. The WM label includes subcortical structures such as deep grey matter and ventricle. The pGT segmentations can be obtained from either advanced learning-based approaches (Dai et al., 2013; Roy et al., 2019; Henschel et al., 2020; Billot et al., 2023a; Uus et al., 2023a) or manual annotations.

To learn the white matter surface, we minimise the commonly used bidirectional Chamfer (bi-Chamfer) distance $\mathcal{L}_{cd}$ (Fan et al., 2017; Wang et al., 2018; Wickramasinghe et al., 2020) between the predicted white matter surface mesh $\widehat{\mathcal{M}}$ and the boundary of the pGT WM segmentation. The boundary is represented as a 3D surface mesh $\mathcal{M}_*$ extracted from the WM segmentation by Marching Cubes (Lorensen and Cline, 1998). Taubin smoothing (Taubin, 1995a,b) is applied to alleviate the grid artefacts of the extracted mesh. Then, the bi-Chamfer loss (Fan et al., 2017) is defined as:

$$\mathcal{L}_{cd}(\widehat{\mathcal{M}}, \mathcal{M}_*) = \frac{1}{|\widehat{\mathcal{V}}|} \sum_{\hat{x} \in \widehat{\mathcal{V}}} \min_{x_* \in \mathcal{V}_*} \|\hat{x} - x_*\|^2 + \frac{1}{|\mathcal{V}_*|} \sum_{x_* \in \mathcal{V}_*} \min_{\hat{x} \in \widehat{\mathcal{V}}} \|x_* - \hat{x}\|^2, \tag{4.7}$$

where $\hat{x}$ and x_* are the vertices of the predicted mesh $\widehat{\mathcal{M}}$ and the pGT WM segmentation boundary $\mathcal{M}_*$ respectively. We add the edge length loss $\mathcal{L}_{edge}$ and normal consistency loss $\mathcal{L}_{nc}$ (Wickramasinghe et al., 2020; Bongratz et al., 2022; Ma et al., 2023) to enforce surface smoothness. The final loss is defined as $\mathcal{L}_{white} = \mathcal{L}_{cd} + w_{edge}\mathcal{L}_{edge} + w_{nc}\mathcal{L}_{nc}$ with weights w_{edge} and w_{nc}.

4.2.2.4 Pial Surface Reconstruction

Since the cGM segmentations are strongly affected by partial volume effects (Ballester et al., 2002; Van Leemput et al., 2003; Tohka et al., 2004), it only provides weak supervision especially for low-resolution data. Figure 4.9-a shows that the cGM segmentation

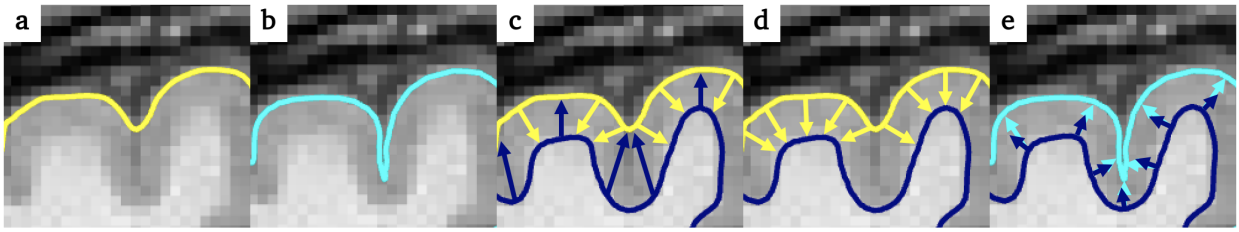


Figure 4.9: Visualisation for the computation of different loss functions on a T1w brain MRI from the HCP dataset (Van Essen et al., 2013). (a) The cGM segmentation boundary (yellow); (b) The expected pial surface (cyan); (c) Bi-Chamfer distance between the cGM segmentation boundary and input white matter surface (blue). (d) The boundary loss, *i.e.*, single-directional Chamfer distance. (e) The inflation loss between the vertex displacement (cyan) and the normal vector (blue) of the input white matter surface.

fails to identify the voxels in the deep cortical sulci. Different from white matter surface reconstruction, the pial surfaces learned via the bi-Chamfer loss may fail to capture cortical folding and thus cause inaccurate estimation of morphological features.

Inspired by deformable surface models (Schuh et al., 2017), we propose a novel weakly supervised loss to tackle partial volume effects on the pial surfaces. The predicted white matter surface is used as the input mesh $\mathcal{M}_0$. First, we introduce a *boundary loss*:

$$\mathcal{L}_{boundary}(\widehat{\mathcal{M}}, \mathcal{M}_*) = \frac{2}{|\mathcal{V}_*|} \sum_{x_* \in \mathcal{V}_*} \min_{\hat{x} \in \widehat{\mathcal{V}}} \|x_* - \hat{x}\|^2. \tag{4.8}$$

The boundary loss is equivalent to a single-directional Chamfer loss that computes the shortest distance from the boundary $\mathcal{M}_*$ of the pGT cGM segmentation to the predicted pial surface $\widehat{\mathcal{M}}$. As illustrated in Figure 4.9-c,d, the proposed boundary loss will extend the input white matter surface towards the cGM boundary without affecting the sulcal regions, whereas the bi-Chamfer distance will deform the surface outwards from the deep sulci. Next, an *inflation loss* is defined as:

$$\mathcal{L}_{inflation}(\widehat{\mathcal{M}}, \mathcal{M}_0) = 1 - \frac{1}{|\widehat{\mathcal{V}}|} \sum_{i=1}^{|\widehat{\mathcal{V}}|} \frac{\hat{x}^i - x_0^i}{\|\hat{x}^i - x_0^i\| + \epsilon} \cdot n(x_0^i), \tag{4.9}$$

where $\epsilon = 10^{-12}$, x_0^i is the i-th vertex of the input white matter surface $\mathcal{M}_0$, and $n(x_0^i)$ is the normal vector with $\|n(x_0^i)\| = 1$. The inflation loss measures cosine similarity between the

vertex displacement and the normal vectors (Figure 4.9-e). This ensures that the inflation of input white matter surface adheres to its normal direction. The final weakly supervised loss is defined as $\mathcal{L}_{weak} = \mathcal{L}_{boundary} + w_{inflation}\mathcal{L}_{inflation}$.

The weakly supervised loss encourages the input white matter surface to inflate along the normal until it fits the cGM boundary or two gyri touch each other. Such inflation can alleviate the partial volume effects and allows the pial surfaces to deform into the deep cortical sulci. During the inflation, mesh self-intersections can be prevented by diffeomorphic surface deformations without additional collision detection. Considering the smoothness terms, the final loss for pial surface reconstruction is defined as $\mathcal{L}_{pial} = \mathcal{L}_{weak} + w_{edge}\mathcal{L}_{edge} + w_{nc}\mathcal{L}_{nc}$.

At the beginning of the training, we have $\widehat{\mathcal{M}} = \mathcal{M}_0$ since the input white matter surface has not been deformed. The gradient of the inflation loss $\mathcal{L}_{inflation}$ with respect to the i-th vertex can be computed as $\nabla_i \mathcal{L}_{inflation}|_{\widehat{\mathcal{M}}=\mathcal{M}_0} = -n(x_0^i)/(\epsilon|\widehat{\mathcal{V}}|)$. However, this leads to exploding gradients as ϵ is very small. To address this issue, we pre-train the TA-Net for a few epochs by replacing $\mathcal{L}_{weak}$ with a MSE loss $\mathcal{L}_{mse}(\widehat{\mathcal{M}}, \mathcal{M}_N) = \sum_{i=1}^{|\widehat{\mathcal{V}}|} \|\hat{x}^i - x_N^i\|^2/|\widehat{\mathcal{V}}|$, where $\mathcal{M}_N$ is an inflated white matter surface with vertices $x_N^i = x_{N-1}^i + \gamma n(x_{N-1}^i)$ for $N = 1, ..., 10$ and $\gamma = 0.1$. The predicted surface learns to extend outwards during pre-training while avoiding exploding gradients. After pre-training, we resume to use $\mathcal{L}_{weak}$ as the reconstruction loss.

4.2.3 Experiments

4.2.3.1 Experiments on Adult Brain MRI

We evaluate CoSeg on the public HCP young adult dataset (Van Essen et al., 2013) with 1,113 T1w brain MRI, cortical ribbon segmentations and cortical surfaces extracted by the HCP pipeline (Glasser et al., 2013) with quality control. The segmentations are used as

Table 4.4: Comparison with existing learning-based approaches in terms of the types of surface representation, required supervision and primary loss functions.

Method	Type	pGT	Loss function
3D U-Net (Ronneberger et al., 2015)	Implicit	Segmentation	Cross Entropy
DeepCSR (Cruz et al., 2021)	Implicit	SDF	L1 Loss
Vox2Cortex (Bongratz et al., 2022)	Explicit	Surface	Bi-Chamfer
CortexODE (Ma et al., 2022)	Explicit	Surface	Bi-Chamfer
CFPP (Santa Cruz et al., 2022)	Explicit	Surface	Bi-Chamfer
CoTAN (Ma et al., 2023)	Explicit	Surface	Bi-Chamfer
CoSeg (Ours)	Explicit	Segmentation	Weakly Supervised

the pGT for training, and the pGT cortical surfaces are only used for evaluation. The dataset is split by the ratio of 60/10/30% for training/validation/testing. All T1w images and segmentations have been affinely aligned to the MNI152 space (Fonov et al., 2011). We compute the average segmentations of all training subjects as a template, from which we extract an initial surface with 160k vertices. Both MRI and segmentations are resampled to $1mm/2mm$ isotropic resolutions and clipped to the sizes of 112×224×176 and 56×112×88 for each brain hemisphere. Without loss of generality, we only consider the left hemisphere. We train TA-Nets for 200 epochs by Adam optimiser with learning rate 0.0001. For pial surfaces, we first pre-train the TA-Net using the MSE loss $\mathcal{L}_{mse}$ for 20 epochs, and then optimise the weakly supervised loss $\mathcal{L}_{weak}$ for 180 epochs. We use $w_{edge} = 0.5$ and $w_{nc} = 5.0$ for smoothness terms. The integration time and step size are set to $T = 1$ and $h = 0.02$ with $K = 50$ steps. We compare CoSeg with both implicit and explicit learning-based cortical surface reconstruction approaches as listed in Table 4.4. All experiments are performed on a NVIDIA RTX 3080 GPU with 12GB memory.

Comparison with Implicit Approaches. We compare CoSeg with implicit learning-based approaches for $1mm$ and $2mm$ resolutions. For CoSeg, we train two TA-Nets for white matter and pial surfaces supervised by the pGT segmentation boundary. The weights $w_{inflation}$ are set to 2.0/5.0 for $1mm/2mm$ resolutions. Since the main backbone of the TA-Net is a 3D U-Net (Ronneberger et al., 2015), we train U-Nets to learn segmentations for fair comparison. We also compare to DeepCSR (Cruz et al., 2021), of which the pGT SDFs

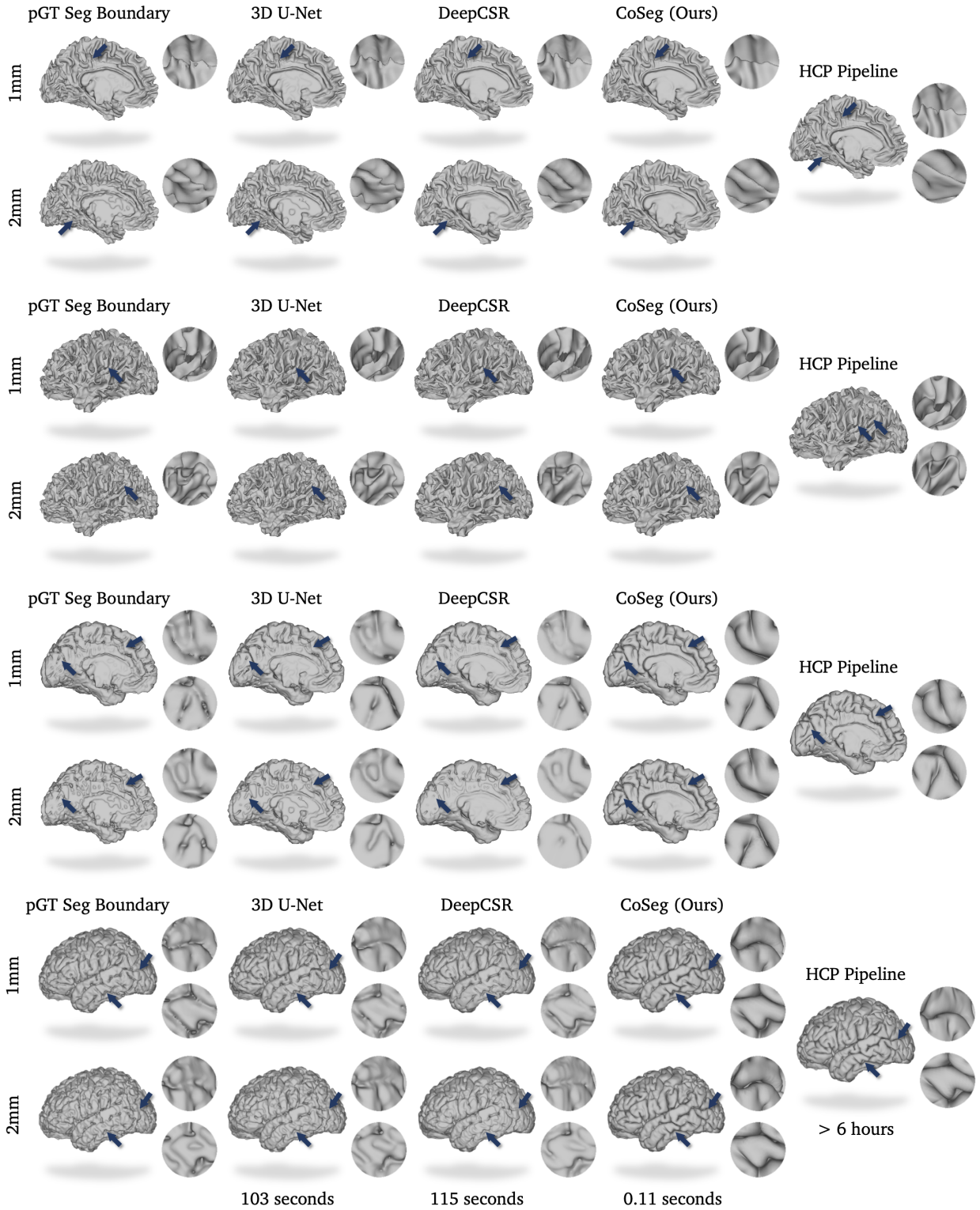


Figure 4.10: Cortical surfaces predicted by implicit approaches on the HCP dataset. The runtime required for each brain hemisphere (including topology correction) is reported. From top to bottom: white matter surface (medial view), white matter surface (lateral view), pial surface (medial view), pial surface (lateral view).

Table 4.5: Comparison with implicit approaches on the HCP dataset at $1mm/2mm$ resolutions. The ASSD (mm) and HD90 (mm) are compared.

		White Matter		Pial	
	Method	ASSD↓	HD90↓	ASSD↓	HD90↓
$1mm$	3D U-Net	0.180	**0.373**	0.432	1.257
	DeepCSR	0.213	0.468	0.436	1.456
	CoSeg (Ours)	**0.177**	0.389	**0.257**	**0.553**
$2mm$	3D U-Net	0.385	0.831	0.833	3.125
	DeepCSR	0.404	1.019	0.904	4.007
	CoSeg (Ours)	**0.330**	**0.756**	**0.361**	**0.767**

are created by the distance transform (Butt and Maragos, 1998) of pGT segmentations. The surface meshes ($|V|\approx$150k) are extracted by Marching Cubes followed by topology correction (Bazin and Pham, 2005, 2007) and Taubin smoothing (Taubin, 1995a,b). The performance is evaluated by the average symmetric surface distance (ASSD) and 90th-percentile Hausdorff distance (HD90) (Cruz et al., 2021; Bongratz et al., 2022; Ma et al., 2022) between the predicted and pGT cortical surfaces.

The comparative results are presented in Table 4.5 and Figure 4.10. Table 4.5 reports that CoSeg achieves superior geometric accuracy except for white matter surfaces at $1mm$. Figure 4.10 shows that CoSeg is able to deform the pial surfaces into deep sulci, while the implicit approaches fail to capture the cortical sulci due to intrinsic partial volume effects of volumetric data, leading to large geometric errors for pial surfaces. CoSeg only requires 0.11s of runtime to predict cortical surfaces for each brain hemisphere. This is orders of magnitude faster than the HCP pipeline, U-Net and DeepCSR as reported in Figure 4.10. CoSeg only produces a negligible number of self-intersecting faces, *i.e.*, 0.06/3.65 out of 330k faces on average for white matter/pial surface respectively.

Comparison with Explicit Approaches. We compare CoSeg with explicit learning-based approaches including Vox2Cortex (Bongratz et al., 2022), CortexODE (Ma et al., 2022), CorticalFlow++ (CFPP) (Santa Cruz et al., 2022), and CoTAN (Ma et al., 2023). For CoTAN we use the same architecture as the TA-Net for adult data. To show the advantage of proposed weakly supervised loss, we train all approaches to learn pial surfaces from the

Table 4.6: Comparison with explicit learning-based approaches and ablation studies of CoSeg on the HCP dataset at $1mm/2mm$ resolutions. The ASSD (mm), HD90 (mm), errors of cortical thickness e_{thick} (mm) and sulcal depth e_{sulc} (mm) are compared. The best results are in bold and the second best results are underlined.

Method	Resolution ($1mm$)				Resolution ($2mm$)			
	ASSD↓	HD90↓	e_{thick} ↓	e_{sulc} ↓	ASSD↓	HD90↓	e_{thick} ↓	e_{sulc} ↓
Vox2Cortex	0.292	0.641	0.511	1.238	0.542	2.026	0.731	1.432
CortexODE	0.352	0.796	0.448	0.672	0.610	1.787	0.742	1.303
CFPP	0.274	0.671	0.496	1.417	0.578	2.385	0.872	1.647
CoTAN	0.293	0.769	0.534	1.438	0.538	1.966	0.781	1.781
CoSeg ($w_{inflate} = 0.0$)	<u>0.253</u>	<u>0.564</u>	0.379	1.389	0.383	0.980	0.551	1.580
CoSeg ($w_{inflate} = 2.0$)	**0.252**	**0.539**	**0.298**	<u>0.530</u>	**0.347**	**0.769**	0.424	<u>0.686</u>
CoSeg ($w_{inflate} = 5.0$)	0.273	0.627	<u>0.320</u>	**0.435**	<u>0.351</u>	<u>0.777</u>	**0.386**	**0.397**
CoSeg ($w_{inflate} = 10.0$)	0.298	0.736	0.362	1.120	0.370	0.881	<u>0.421</u>	0.797

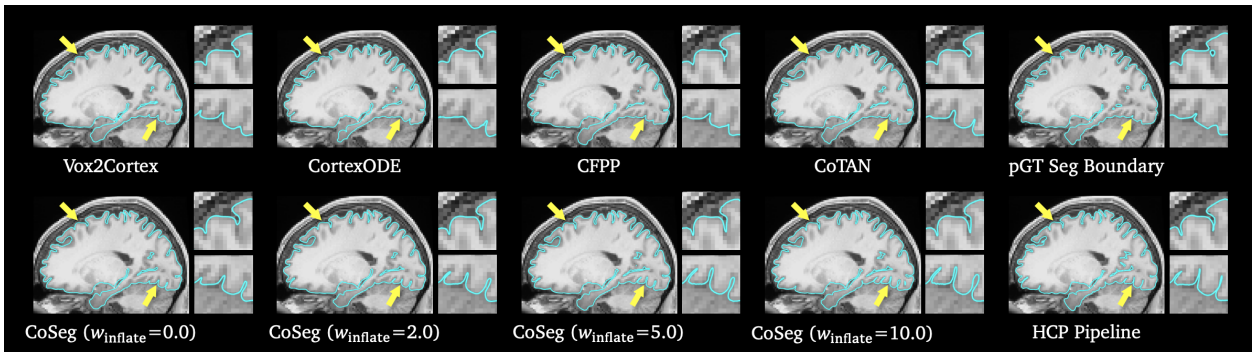


Figure 4.11: Pial surfaces predicted by explicit approaches on the HCP dataset ($2mm$).

pGT segmentation boundary at $1mm/2mm$ resolutions. The pGT white matter surfaces are used as the input for training, such that the predicted and pGT pial surfaces have the same mesh connectivity. This allows vertex-to-vertex comparison of cortical morphological features. We compute the average L1 errors of the cortical thickness and the sulcal depth of midthickness surfaces over all vertices between the predicted and pGT cortical surfaces. The geometric accuracy is measured by ASSD and HD90.

As compared in Table 4.6 and Figure 4.11, CoSeg achieves superior geometric and morphological accuracy than all baselines including CortexODE and CoTAN. Existing explicit approaches (Bongratz et al., 2022; Santa Cruz et al., 2022; Ma et al., 2022, 2023), which use the bi-Chamfer loss (see Table 4.4), overfit to the pGT segmentation boundary and thus fail to fit the deep cortical sulci especially for low resolution samples. This results

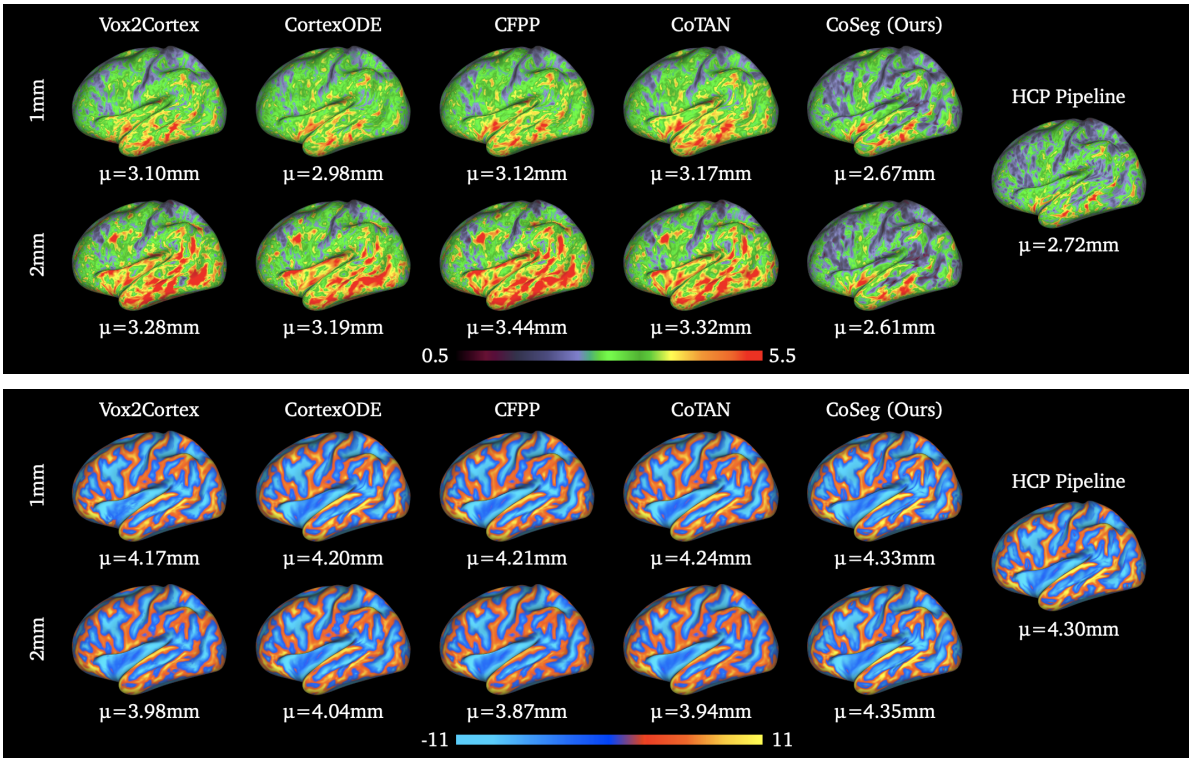


Figure 4.12: Cortical thickness (top) and sulcal depth (bottom) of cortical surfaces predicted by explicit learning-based approaches.

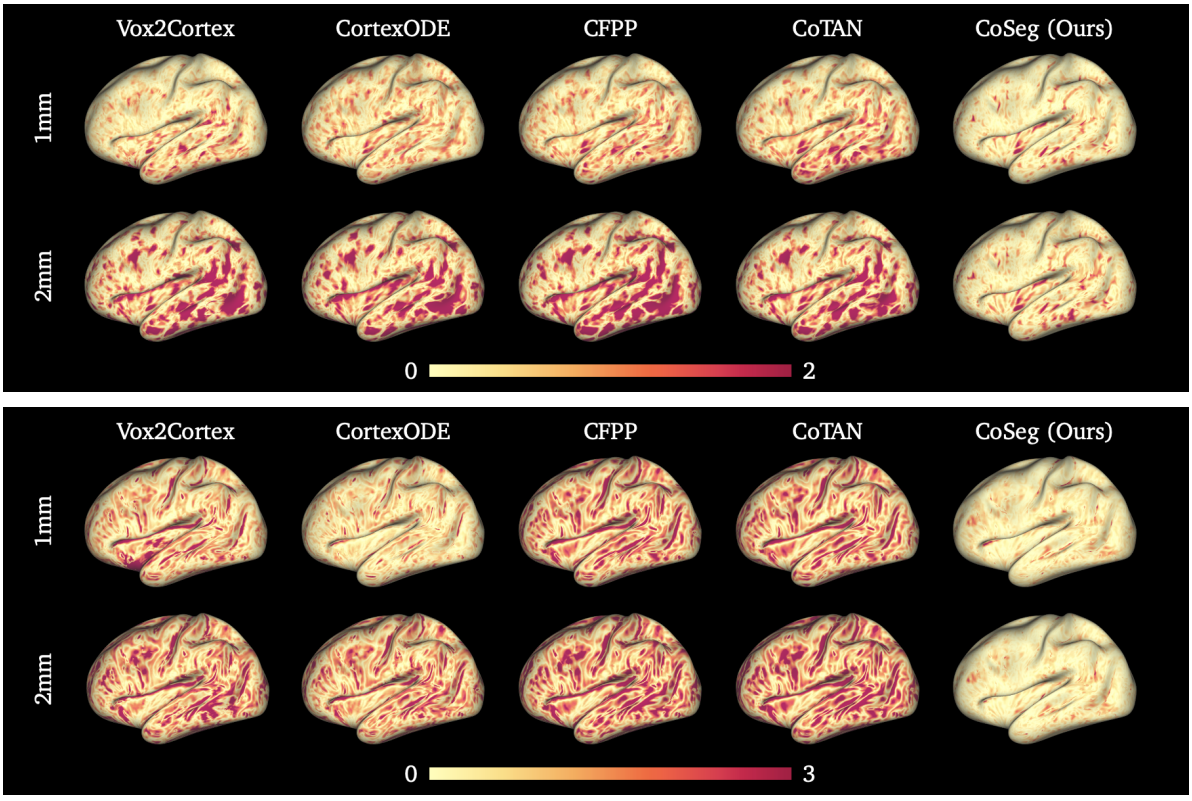


Figure 4.13: The error maps (mm) of cortical thickness (top) and sulcal depth (bottom) for explicit learning-based approaches.

in inaccurate morphological features as reported in Table 4.6, more specifically, greater cortical thickness and shallower sulcal depth as displayed in Figure 4.12 and 4.13. We also conduct ablation studies to explore the influence of different weights $w_{inflation}$ for the inflation loss $\mathcal{L}_{inflation}$ on CoSeg. The ablation studies show that the inflation loss effectively reduces morphological errors. However, excessive weight for the inflation loss also affects the performance, since the vertex displacement of the pial surface may not strictly follow the normal direction of the white matter surface.

4.2.3.2 Experiments on Fetal Brain MRI

We evaluate CoSeg on the dHCP fetal dataset (Price et al., 2019) with 241 T2w fetal brain MRI scanned at the ages of 22–38 weeks. The dataset is split by the ratio of 60/10/30%. The voxel size of the T2w images is $0.5mm^3$. However, this is a still relatively low resolution as the fetal brains are very small. The pGT tissue segmentations are created by the BOUNTI pipeline (Uus et al., 2023a), a fully validated learning-based fetal brain segmentation approach. The pGT cortical ribbon segmentations are created by merging the tissue labels. Both T2w images and segmentations are affinely aligned to a 36-week fetal brain MRI atlas (Uus et al., 2023b) and clipped to the size of 112×224×176 for each hemisphere. An initial surface is generated based on the segmentation atlas (Uus et al., 2023b). For CoSeg we set $w_{inflation} = 5.0$.

We compare CoSeg with the dHCP neonatal pipeline (Makropoulos et al., 2018), CoTAN (Ma et al., 2023) and 3D U-Net supervised by pGT segmentations. Since the Dice score cannot reflect the geometric accuracy, we provide qualitative comparisons in Figure 4.14. It shows that CoSeg predicts high-quality surfaces, while the dHCP neonatal pipeline fails to generalise well on fetal subjects. The pial surfaces predicted by U-Net and CoTAN are severely affected by the partial volume effects of the cGM. In addition, compared to the dHCP neonatal pipeline which requires 6.5 hours of runtime, CoSeg only needs less than 0.2 seconds for each brain hemisphere.

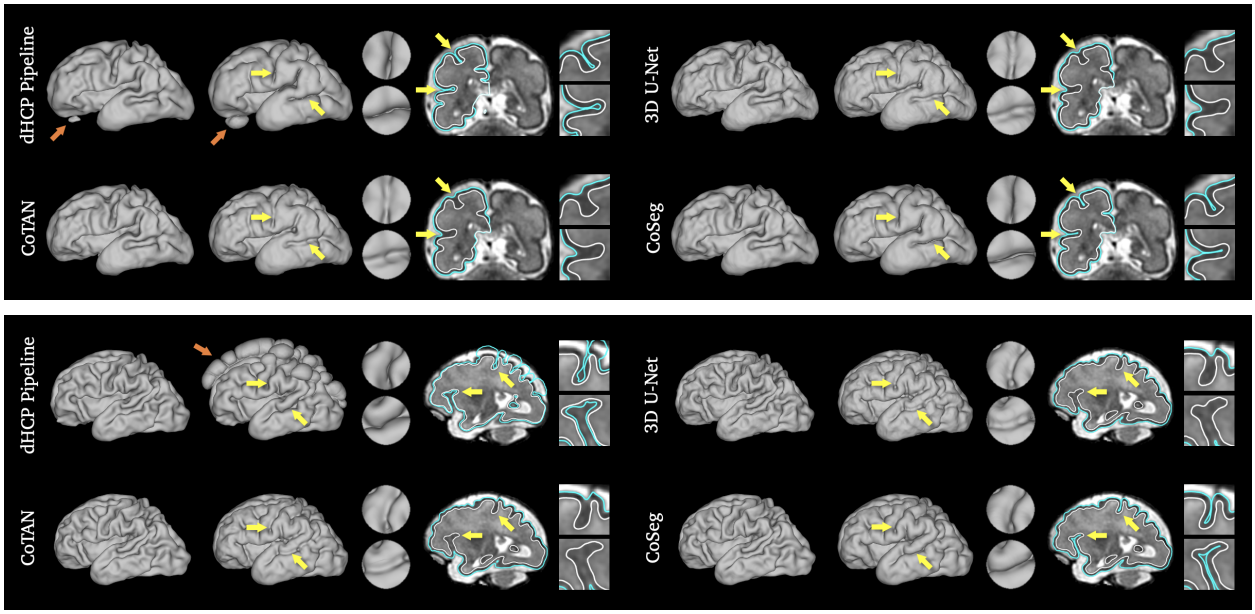


Figure 4.14: Qualitative comparison of cortical surfaces on the dHCP fetal dataset. Top: a 29.4-week subject. Bottom: a 32.4-week subject.

4.2.4 Discussion

In this section, we propose CoSeg for diffeomorphic cortical surface reconstruction, weakly supervised by cortical ribbon segmentations without the reliance on pGT cortical surfaces. A novel weakly supervised loss is designed to address partial volume effects and deform the pial surfaces into deep cortical sulci. CoSeg demonstrates remarkable results on both HCP young adult and dHCP fetal brain MRI datasets, whereas the CortexODE and CoTAN approaches introduced in prior sections do not achieve ideal performance without the supervision from pGT cortical surfaces. CoSeg saves considerable processing time required by traditional cortical surface reconstruction pipelines (Fischl, 2012; Glasser et al., 2013; Makropoulos et al., 2018) for dataset preparation. In particular, for brain MRI with limited image resolution, *e.g.*, fetal brain MRI scanned *in utero*, CoSeg is still capable of learning anatomically accurate cortical surfaces guided by brain segmentation maps, while traditional pipelines fail to reconstruct acceptable cortical surfaces.

One limitation of CoSeg is that it guides vertex displacement strictly along the normal of the input white matter surface, with potential to affect anatomical fidelity of pial surfaces.

The efficacy of CoSeg can also be influenced by the quality of pGT segmentations. Therefore, the CortexODE and CoTAN models are more effective when high-quality pGT cortical surfaces are available. Looking ahead, semi-supervised learning approaches could be explored, utilising both labelled segmentations and unlabelled MRI intensity to refine the anatomical accuracy.

Chapter 5

A Deep Learning-based Pipeline for Neonatal Cortical Surface Reconstruction

The content of this chapter is based on:

- **Ma, Q.**, Liang, K., Li, L., Masui, S., Guo, Y., Nosarti, C., Robinson, E.C., Kainz, B. and Rueckert, D. (2024). The Developing Human Connectome Project: A Fast Deep Learning-based Pipeline for Neonatal Cortical Surface Reconstruction. *Medical Image Analysis*, 100:103394. https://doi.org/10.1016/j.media.2024.103394

In previous chapters, we have introduced several geometric deep learning (DL) frameworks for adult, neonatal, and fetal cortical surface reconstruction. These DL-based approaches achieve fast inference and extract cortical surfaces end-to-end from brain MRI scans within a few seconds. However, such end-to-end frameworks do not cover downstream cortical surface analysis, which is indispensable to the studies of brain development (Wright et al., 2014; Garcia et al., 2018b) and neurological disorders (Fischl and Dale, 2000; Aël Chetelat and Baron, 2003). In this chapter, therefore, we present a complete DL-based neuroimage processing pipeline, including cortical surface reconstruction and its downstream applications, with emphasis on the Developing Human Connectome Project (dHCP) (Makropoulos et al., 2018; Edwards et al., 2022).

The dHCP aims to explore developmental patterns of human brains during the perinatal period (Edwards et al., 2022). An automated processing pipeline (Makropoulos et al., 2018) has been developed to extract cortical surfaces from structural brain MRI for the

dHCP neonatal dataset. However, the current implementation of the pipeline requires more than 6.5 hours to process a single scan, making it expensive for large-scale neuroimaging studies. In this chapter, we introduce a fast and robust DL-based pipeline for dHCP neonatal brain MRI processing, incorporating DL-based brain extraction, cortical surface reconstruction and spherical projection, as well as GPU-accelerated cortical surface inflation and cortical feature estimation. Based on the neural network architectures presented in previous chapters, we propose a multiscale deformation network to learn diffeomorphic cortical surface reconstruction end-to-end from brain MRI. A fast unsupervised spherical mapping approach is integrated to minimise metric distortions between cortical surfaces and projected spheres. Our DL-based dHCP pipeline achieves superior surface quality in manual quality control, while being nearly $1000\times$ faster than the original dHCP pipeline.

5.1 Introduction

Background. During the perinatal phase, the human brain undergoes significant developmental changes, characterised by the formation of anatomical and functional connectivity. In this context, the dHCP is created to compile an extensive dataset of brain MRI for both fetuses and neonates (Makropoulos et al., 2018; Edwards et al., 2022). This ambitious project is instrumental in facilitating an in-depth understanding of early brain development and in constructing a four-dimensional connectome for young brains. The dHCP dataset is a valuable asset for investigating the normal and abnormal patterns of brain structural development and their respective connectomes. Notably, the dHCP has publicly released its neonatal dataset, comprising MRI scans of 783 newborn infants, both pre-term and full-term, resulting in a total of 887 images (Edwards et al., 2022).

Cortical surfaces, *i.e.*, the inner and outer surfaces of the cerebral cortex, include a white matter surface lying between the white matter (WM) and cortical grey matter (cGM), as well as a pial surface lying between the cGM and cerebrospinal fluid (CSF). The cortical folding and brain development of newborn infants can be modelled and quantified by the morphological features of cortical surfaces such as cortical thickness, mean curvature and sulcal depth (Nie et al., 2012; Wright et al., 2014; Garcia et al., 2018a,b). The notable achievements of FreeSurfer (Dale et al., 1999; Fischl et al., 1999a; Fischl, 2012) and the Human Connectome Project (HCP) (Van Essen et al., 2013; Glasser et al., 2013), recognised for their automated neuroimaging processing pipelines for adults, underscore the importance of implementing cortical surface-based structural brain MRI processing and analysis for the dHCP fetal and neonatal datasets.

However, the pronounced differences between adult and neonatal brains present significant challenges in adapting existing adult MRI processing pipelines for use in newborn infants (MacDonald et al., 2000; Shattuck and Leahy, 2002; Han et al., 2004; Kim et al., 2005; Fischl, 2012; Glasser et al., 2013). One primary challenge lies in the limited res-

olution of neonatal brain MRI, a consequence of the necessarily brief data acquisition times (Hughes et al., 2017). This limitation is further compounded by the smaller head size in neonates, which results in a reduced region of interest (ROI) (Orasanu et al., 2014; Makropoulos et al., 2016). Besides, the rapid developmental changes in the neonatal brain contribute to a significant variability in the geometric shape and size of the brain across different ages (Bethlehem et al., 2022). Another notable aspect is the reversed contrast in neonatal brain MRI, attributed to the ongoing myelination process in the white matter, which is not yet complete (Prastawa et al., 2005). As a result of these factors, T2-weighted (T2w) images are predominantly used in the structural MRI processing of neonatal brains.

Related Work. To address the challenges of neonatal cortical surface extraction, Makropoulos et al. (2018) proposed an automated structural brain MRI processing pipeline for dHCP neonatal dataset. The pipeline consists of brain extraction, bias field correction, brain segmentation, cortical surface reconstruction, inflation and parcellation, cortical feature estimation, and spherical projection. The dHCP structural pipeline (Makropoulos et al., 2018) employs the FMRIB software library (FSL) brain extraction tool (BET) (Smith, 2002; Jenkinson et al., 2012) for skull stripping and the N4 algorithm (Tustison et al., 2010) for bias correction. The Draw-EM approach (Makropoulos et al., 2014) is adopted for brain regional and tissue segmentation. A deformable model-based approach (Schuh et al., 2017) is introduced for cortical surface reconstruction. Such an approach iteratively deforms an initial convex hull surface to fit the zero level set of an implicit surface defined by the WM segmentation. Based on the T2w MRI intensity, the surface is then deformed towards the WM/cGM and cGM/CSF interface to generate white matter and pial surfaces. The spherical topology is guaranteed and no topology correction (Fischl et al., 2001; Ségonne et al., 2007; Bazin and Pham, 2005, 2007) is required. For spherical projection, the dHCP pipeline adopts the Spherical Multi-Dimensional Scaling (MDS) approach (Elad et al., 2005) to minimise the geodesic distance distortions between the sphere and the white matter surface. The remaining parts of the pipeline follow the design of FreeSurfer (Fischl, 2012) and the HCP pipeline (Glasser et al., 2013).

The dHCP structural pipeline has shown its advantages on cortical surface-based neonatal MRI analysis (Makropoulos et al., 2018) and provides crucial support for downstream applications (Schuh et al., 2018; Bozek et al., 2018; Fetit et al., 2020). However, the current dHCP pipeline, as well as other conventional infant neuroimage processing pipelines such as infant FreeSurfer (Zöllei et al., 2020), iBEAT (Dai et al., 2013), and NEOCIVET (Kim et al., 2015), still have limitations. First of all, these pipelines require several hours to process a single brain MRI scan, which are computationally expensive for large-scale neuroimage studies. Additionally, the quality of cortical surfaces extracted by the pipelines heavily relies on the brain tissue segmentations. Therefore, inaccurate segmentations can cause subsequent corruptions in the cortical surfaces, especially in the parietal and occipital lobes where the T2w MRI has low image contrast (Makropoulos et al., 2014).

To overcome the limitations in the traditional neuroimage pipelines, one alternative solution is deep learning (DL), which trains a deep neural network on a large dataset and makes end-to-end predictions during inference. DL-based approaches are supported by mature frameworks such as TensorFlow (Abadi et al., 2016) and PyTorch (Paszke et al., 2019), and accelerated by modern graphics processing units (GPUs) with fast parallel computation (Owens et al., 2008; Keckler et al., 2011). Recently, several DL-based brain MRI processing pipelines have been proposed such as FastSurfer (Henschel et al., 2020), DeeepCSR (Cruz et al., 2021), SegRecon (Gopinath et al., 2021, 2023a), and iBEAT V2.0 (Wang et al., 2023). These pipelines introduce learning-based modules to predict implicit surfaces from brain MRI for cortical surface reconstruction. The implicit surfaces are represented as a segmentation or a signed distance function (SDF), which is a level set indicating the signed distance to the surface (Park et al., 2019). An explicit 3D mesh can be extracted from the implicit surface using the Marching Cubes algorithm (Lorensen and Cline, 1998). Subsequently, topology correction algorithms (Fischl et al., 2001; Shattuck and Leahy, 2001; Han et al., 2002; Bazin and Pham, 2005, 2007; Ségonne et al., 2007) are usually applied to fix topological errors in the extracted surface such that it is topologically equivalent to a sphere. However, such topology correction algorithms are usually

time-consuming as they either repeatedly project the surface onto a sphere to detect topological defects (Fischl et al., 2001; Ségonne et al., 2007), or iterate over all voxels to detect critical points (Bazin and Pham, 2005, 2007). Hence, current DL-based pipelines (Henschel et al., 2020; Wang et al., 2023; Billot et al., 2023b) still require more than half an hour to process a single subject, although they can extract accurate cortical surfaces.

In contrast to learning implicit surfaces, explicit learning-based approaches (Wickramasinghe et al., 2020; Ma et al., 2021, 2022, 2023; Lebrat et al., 2021; Hoopes et al., 2022; Santa Cruz et al., 2022; Bongratz et al., 2022, 2024; Zheng et al., 2024) predict a sequence of deformations from brain MRI to deform an initial template surface explicitly into cortical surfaces. Since explicit deformations do not change the surface topology, time-consuming topology correction is not necessary if the initial surface has genus-0 topology. Therefore, explicit approaches, being both fast and capable of end-to-end learning, require only a few seconds to extract cortical surfaces, effectively circumventing the corruptions caused by imprecise segmentations. In order to learn high-quality manifold meshes (Gupta and Chandraker, 2020), diffeomorphic deformations, which are widely used in medical image registration (Beg et al., 2005; Ashburner, 2007; Vercauteren et al., 2009; Balakrishnan et al., 2019), have been incorporated into recent explicit learning-based cortical surface reconstruction approaches such as CortexODE (Ma et al., 2022), CorticalFlow (Lebrat et al., 2021), CorticalFlow++ (Santa Cruz et al., 2022), CoTAN (Ma et al., 2023), CoSeg (Ma et al., 2024), V2C-Flow (Bongratz et al., 2024), and Zheng et al. (2024). Such approaches aim to learn diffeomorphic surface deformations to preserve the surface topology and prevent surface self-intersections, which will affect the estimation of cortical geometric features such as cortical volume, surface area and cortical thickness (Ma et al., 2022).

After surface reconstruction, spherical projection or spherical mapping, is usually performed to project the cortical surface onto a spherical mesh while minimising their metric distortions (Fischl et al., 1999a; Elad et al., 2005), *e.g.*, the distortions of edge length, face area, or geodesic distance. The spherical projection allows the cortical surface to

be represented in a spherical coordinate system for downstream tasks such as cortical surface registration (Fischl et al., 1999b; Robinson et al., 2014, 2018) and atlas construction (Fischl et al., 2004; Desikan et al., 2006; Destrieux et al., 2010; Van Essen et al., 2012; Glasser et al., 2016; Bozek et al., 2018). Conventional spherical projection approaches (Fischl et al., 1999a; Elad et al., 2005) are computationally intensive due to the iterative inflation of cortical surfaces and optimisation of metric distortions. Recently, Zhao et al. (2022) proposed an end-to-end unsupervised learning framework for spherical mapping. Multiscale Spherical U-Net models (Zhao et al., 2019a,b) are employed to learn coarse-to-fine diffeomorphic deformation fields to deform an initial projected sphere. The Spherical U-Nets are trained by minimising the unsupervised metric distortions between the deformed sphere and resampled cortical surfaces for each resolution.

Contributions. Based on the original dHCP structural pipeline (Makropoulos et al., 2018), we propose a new DL-based pipeline for dHCP neonatal cortical surface reconstruction. Such a pipeline comprises brain extraction, bias field correction, cortical surface reconstruction, cortical surface inflation, cortical feature estimation, and spherical projection. A brain mask is learned by a 3D U-Net (Ronneberger et al., 2015) for skull stripping. A multiscale diffeomorphic deformation network is proposed to extract cortical surfaces from the T2w MRI end-to-end without the requirement of tissue segmentations. The proposed multiscale deformation network incorporates differentiable Gaussian smoothing layers as well as scaling and squaring method for ODE integration with high numerical accuracy (Higham, 2005; Arsigny et al., 2006), thereby producing cortical surfaces with higher mesh quality and negligible number of surface self-intersections compared to the DL-based cortical surface reconstruction frameworks presented in prior chapters, *e.g.*, CortexODE (Chapter 3) and CoTAN (Chapter 4). Moreover, we refine the original unsupervised spherical mapping framework (Zhao et al., 2022) such that the runtime is accelerated from 2.1 to 0.3 seconds, while the edge and area distortions are reduced significantly. Following the procedure of the original dHCP pipeline, the remaining steps of the pipeline are re-implemented with GPU acceleration.

Our DL-based dHCP neonatal pipeline is primarily implemented in Python. Most of the processing steps in the pipeline are implemented with PyTorch (Paszke et al., 2019), a well-known deep learning library enabling GPU acceleration. Our DL-based pipeline is fast and memory-efficient, which only needs 24 seconds to complete all procedures on a modern GPU with 12GB memory (or less than 3 minutes on CPUs only). This is nearly 1000× faster than the original dHCP pipeline, which requires more than 6.5 hours. We perform manual quality control of cortical surfaces on the test set, which demonstrates that our dHCP DL-based pipeline achieves superior surface quality than the original dHCP pipeline while being orders of magnitude faster. The code for our DL-based dHCP pipeline is publicly released at https://github.com/m-qiang/dhcp-dl-neonatal.

5.2 Deep Learning-based dHCP Neonatal Pipeline

5.2.1 Data Acquisition

In this paper, we use the neonatal dataset from the third public dHCP data release (Edwards et al., 2022). The dHCP neonatal dataset includes 783 neonatal subjects with 887 scans. A small portion of the subjects were scanned more than once for the purpose of longitudinal study. The neonates were scanned at post-menstrual age (PMA) between 27 and 45 weeks and the distribution of their ages at birth and scan is shown in Figure 5.1. For MRI acquisition, the infants were imaged in natural sleep on a 3T Philips scanner with a dedicated neonatal brain imaging system (Hughes et al., 2017). The T2w images are acquired in two stacks of 2D slices in axial and sagittal planes at $0.8 \times 0.8mm^2$ resolution and $1.6mm$ overlapping slices. The acquisition uses the parameters repetition/echo time TR/TE=12s/156ms and SENSE factor of 2.11 (axial) and 2.60 (sagittal). A motion correction technique (Cordero-Grande et al., 2018) is adopted to compensate for head motion during imaging. The acquired images are reconstructed to high-quality 3D volumes at $0.5mm^3$ isotropic resolution (Kuklisova-Murgasova et al., 2012).

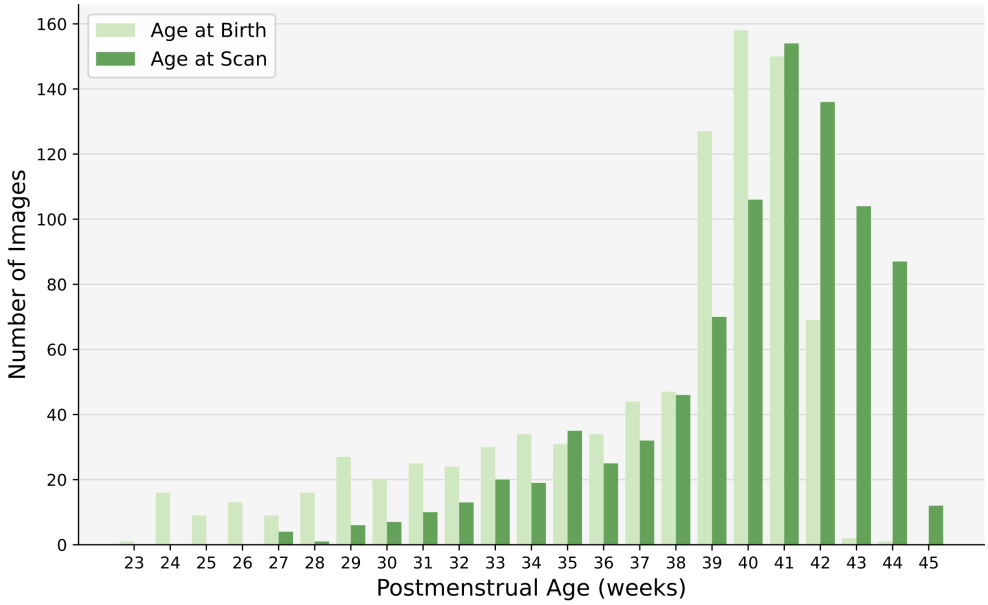


Figure 5.1: The distribution of PMA of neonates at birth and scan.

5.2.2 Pipeline Overview

The workflow of our DL-based dHCP neonatal pipeline is illustrated in Figure 5.2. Our DL-based pipeline accelerates the original dHCP structural pipeline (Makropoulos et al., 2018) by integrating several DL-based modules and re-implemented algorithms with GPU acceleration. The pipeline consists of five main steps: preprocessing, cortical surface reconstruction, cortical surface inflation, cortical feature estimation, and spherical projection. Following the original dHCP structural pipeline, we use T2w structural brain MRI as the primary inputs. Given a T2w image, the pipeline uses a 3D U-Net to learn a brain mask to extract the brain ROI. Then, N4 bias field correction (Tustison et al., 2010) is applied to normalise the intensity of the T2w image. The processed images are affinely aligned to a 40-week dHCP atlas (Schuh et al., 2018). A multiscale deformation network, which learns a sequence of diffeomorphic surface deformations, is introduced to reconstruct cortical white matter and pial surfaces end-to-end from the aligned T2w image. The predicted white matter surface is iteratively inflated and smoothed to generate inflated and very inflated surfaces. Next, the morphological features of the cortex, including cortical thickness, curvature and sulcal depth, are computed from the extracted cortical sur-

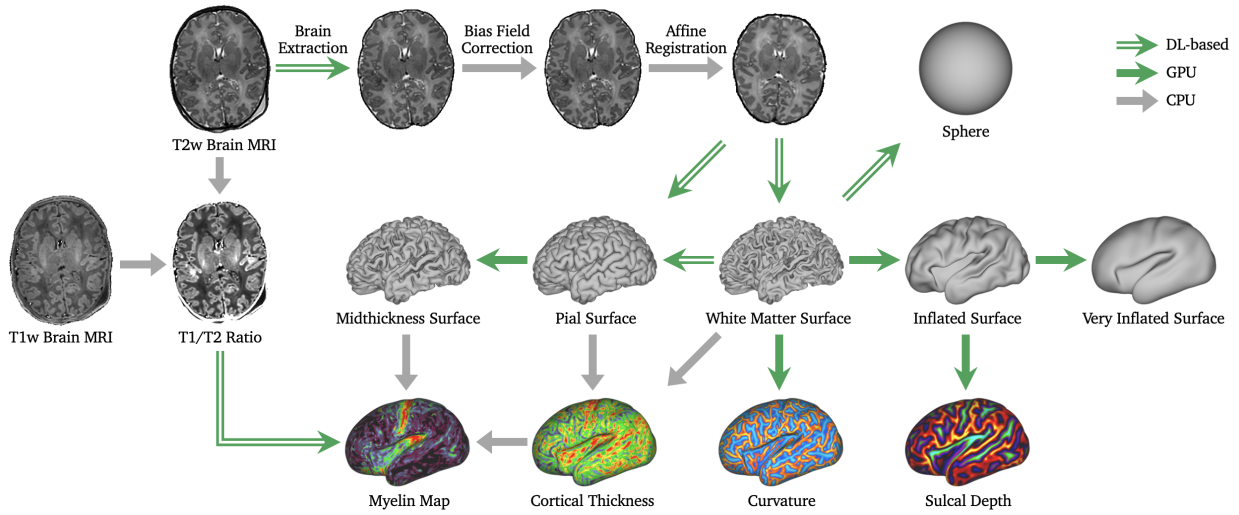


Figure 5.2: The workflow of our DL-based dHCP neonatal structural brain MRI processing pipeline. The pipeline consists of DL-based, GPU-accelerated and CPU-based processing steps. Given a T2w structural brain MRI, the pipeline learns to extract the brain ROI and applies N4 bias field correction. The processed image is then affinely aligned to a 40-week dHCP atlas. The DL-based pipeline predicts cortical white matter and pial surfaces end-to-end from the aligned image. Next, the predicted white matter surface is iteratively inflated and smoothed to produce inflated surfaces. Based on the extracted cortical surfaces, the cortical morphological features including cortical thickness, curvature and sulcal depth are computed. The myelin map is estimated if the T1w image is provided as well. Finally, an unsupervised spherical mapping approach is adopted to project the white matter surface onto a sphere.

faces. If the T1w MRI is provided, the pipeline estimates the myelin map based on the T1/T2 ratio. Finally, the pipeline employs a Spherical U-Net (Zhao et al., 2019a, 2022) to project the white matter surface onto a sphere while minimising the metric distortions.

5.3 Method

5.3.1 Dataset Preparation

To train and evaluate each component of the proposed DL-based pipeline, we use 879 images in the dHCP neonatal dataset (Edwards et al., 2022), in which 8 samples are excluded during data cleaning. Each data sample includes T2w brain MRI as well as pseudo ground truth (GT) segmentations and cortical surfaces generated by the original dHCP

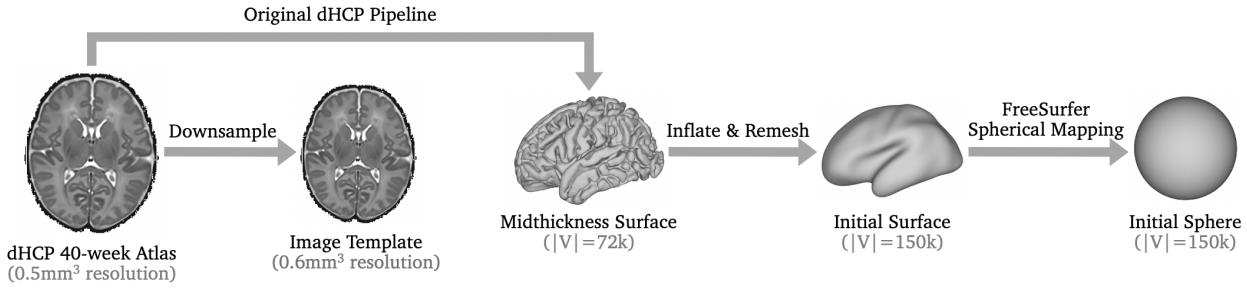


Figure 5.3: The generation of initial image, surface and sphere templates. The 40-week dHCP atlas is downsampled to $0.6mm^3$ resolution as the image template. The midthickness surface for each brain hemisphere, which is extracted from the 40-week dHCP atlas by running the original dHCP pipeline, is inflated and remeshed to create an initial template surface. Then, the initial surface is projected onto an initial sphere through FreeSurfer. $|V|$ is the number of mesh vertices.

structural pipeline (Makropoulos et al., 2018). The original T2w image has the size of $217 \times 290 \times 290$. The dataset is randomly split into training/validation/testing sets with the ratio of $60/10/30\%$. We generate the initial template image, surface and sphere for training and running the pipeline. The generated templates are shared for all subjects. As depicted in Figure 5.3, we create an image template from a 40-week dHCP atlas (Schuh et al., 2018). As we aim to develop a fast and memory-efficient pipeline, we downsample the original atlas from $0.5mm^3$ to $0.6mm^3$ isotropic resolution and clip it to the size of $176 \times 224 \times 160$ to reduce the GPU memory cost. The downsampled atlas is used as the target image template for affine registration.

Next, we run the original dHCP pipeline to extract the midthickness surfaces from the dHCP atlas. For each brain hemisphere, the midthickness surface is iteratively smoothed and remeshed to create an inflated initial surface, which is in the same space as the image template. Such an initial surface mesh has 150k vertices and is used as the input template surface for cortical surface reconstruction. This enables the proposed DL-based pipeline to predict cortical surfaces with the same mesh connectivity and fixed number of vertices, while the original dHCP pipeline extracts cortical surfaces with adapted number of vertices, leading to inaccurate estimation of cortical morphological features such as sulcal depth for younger subjects. Finally, we use FreeSurfer *mris_inflate* and *mris_sphere* (Fischl

et al., 1999a; Fischl, 2012) to project the initial surface onto an initial sphere, which has the same mesh connectivity as the initial surface with minimised metric distortions. The initial sphere is used as the input for spherical projection.

5.3.2 Preprocessing

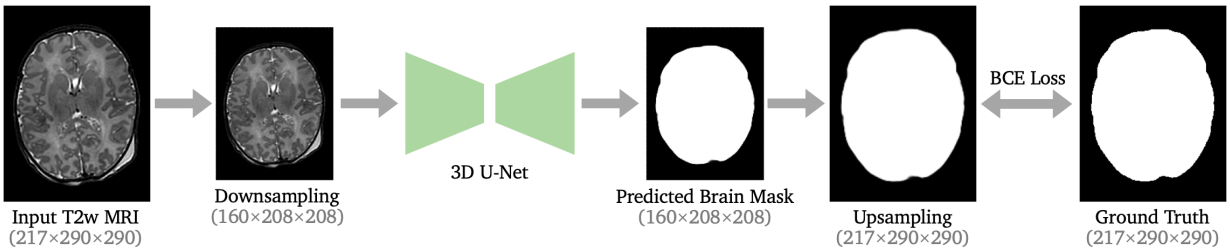


Figure 5.4: The deep learning-based approach for brain extraction. An original T2w image is downsampled as the input for a 3D U-Net to predict a brain mask. The BCE loss is computed between the upsampled predicted brain mask and the ground truth.

Brain Extraction. The preprocessing steps consist of brain extraction, bias field correction and affine registration. For brain extraction, as shown in Figure 5.4, we train a 3D U-Net (Ronneberger et al., 2015) to predict a brain mask from the input original T2w image. To reduce the GPU memory usage, we downsample the input T2w images to the size of $160 \times 208 \times 208$. The predicted brain masks are upsampled to the original size to increase accuracy. We compute the Binary Cross Entropy (BCE) loss between the predicted and GT brain masks. The GT brain masks are generated by FSL BET (Smith, 2002), which is integrated into the original dHCP structural pipeline. The 3D U-Net model is trained by Adam optimiser (Kingma and Ba, 2015) with learning rate 10^{-4} for 100 epochs.

Bias Field Correction and Affine Registration. We use N4 bias field correction (Tustison et al., 2010) to normalise the intensity of the brain-extracted T2w images. Then, we affinely align the bias-corrected image to the dHCP 40-week image template as described in Section 5.3.1. To increase accuracy, we first apply rigid registration to obtain an initial transformation, and then apply the affine registration. The affine registration transforms all MRI into a template space with the size of $176 \times 224 \times 160$, which reduces the GPU

memory cost for the following process. The proposed pipeline integrates Advanced Normalisation Tools in Python (ANTsPy) (Avants et al., 2009), which provides CPU-based implementation for both N4 algorithm and affine registration.

5.3.3 Cortical Surface Reconstruction

We propose an end-to-end learning-based approach for cortical surface reconstruction. To tackle the large deformation required by neonatal cortical surfaces, inspired by the neural network architectures proposed in Chapter 3 and 4, we introduce multiscale deformation networks, which learn a sequence of diffeomorphic transformations to deform an initial surface to cortical surfaces. The architecture is shown in Figure 5.5.

5.3.3.1 Diffeomorphic Surface Deformation

Diffeomorphic transformation has been extensively applied to medical image registration (Beg et al., 2005; Ashburner, 2007; Vercauteren et al., 2009; Balakrishnan et al., 2019) and been introduced for learning-based cortical surface reconstruction (Lebrat et al., 2021; Santa Cruz et al., 2022; Ma et al., 2022, 2023, 2024; Bongratz et al., 2024). For any domain $\Omega \subset \mathbb{R}^d$, the diffeomorphic deformations can be modelled by the following flow ordinary differential equation (ODE) (Coddington et al., 1956; Walter, 2013; Younes, 2010):

$$\frac{\partial}{\partial t}\phi_t = v(\phi_t),\ \phi_0 = id, \tag{5.1}$$

where $v : \Omega \to \mathbb{R}^d$ is a stationary velocity field (SVF), $\phi_t : \Omega \to \Omega$ is a deformation field for $t \in [0, T]$, and $\phi_0 = id$ is an identity map. If the SVF v is sufficiently smooth, then $\{\phi_t\}_{t\in[0,T]}$ represents an one-parameter subgroup of diffeomorphisms generated by v and satisfies the group law $\phi_{s+t} = \phi_s \circ \phi_t$ (Ashburner, 2007; Lee, 2012; Balakrishnan et al., 2019). Therefore, we can solve the autonomous ODE (5.1), of which the solution is a

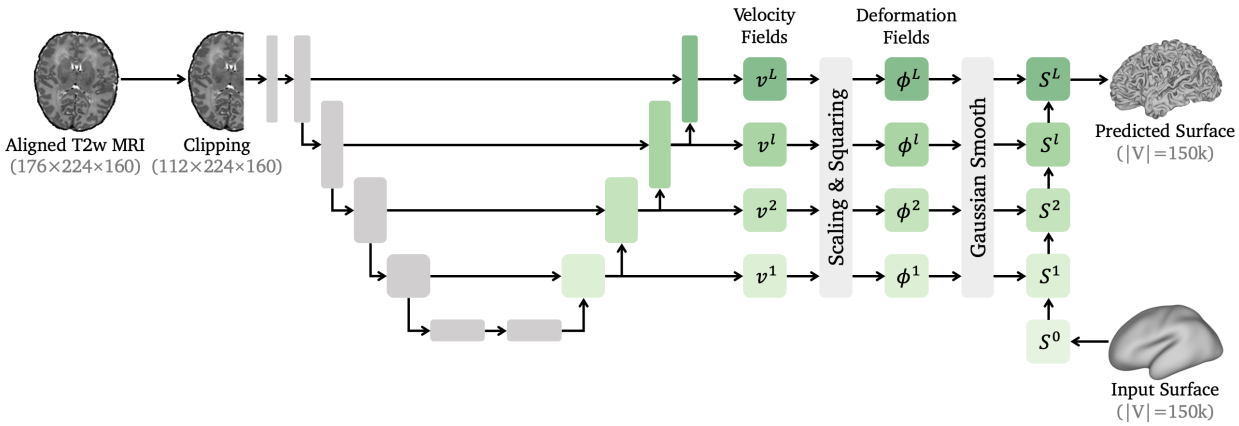


Figure 5.5: The architecture of the multiscale deformation network. An affinely aligned T2w MRI is clipped to the left or right hemisphere as the input. A 3D U-Net is employed to predict multiscale stationary velocity fields (SVF). The SVFs are integrated by scaling and squaring method to multiscale diffeomorphic surface deformation fields, which are smoothed by Gaussian blurring. The multiscale deformation fields deform an input surface iteratively to the predicted cortical surface, where $|V|$ is the number of mesh vertices.

diffeomorphism ϕ_T, using the scaling and squaring method (Higham, 2005; Arsigny et al., 2006; Ashburner, 2007; Balakrishnan et al., 2019) to iteratively compute $\phi_t = \phi_{t/2} \circ \phi_{t/2}$. For surface deformation, the deformation field ϕ_T evolves an initial surface $S_0 \subset \mathbb{R}^3$ to a target surface $S_T = \phi_T(S_0)$ while preserving the topology. Since ϕ_T is a differentiable bijection, any two distinct points on the initial surface S_0 will not be mapped to the same location, which provides theoretical guarantee for preventing surface self-intersections.

5.3.3.2 Multiscale Deformation Network

Since a single SVF has limited representation ability, in this work, we propose multiscale deformation networks to learn a series of diffeomorphic deformation fields $\phi^1, ..., \phi^L$ for cortical surface reconstruction. As shown in Figure 5.5, given an affinely aligned T2w image, we clip it to the size of $112 \times 224 \times 160$ for the left or right brain hemispheres. The multiscale deformation network predicts cortical surfaces end-to-end from the clipped brain MRI. Note that the brain segmentations are not required as the input, which would cause subsequent corruptions on the surfaces if segmentations are inaccurate. Similar to the temporal attention networks (TA-Net) described in Chapter 4, we use a 3D U-Net to

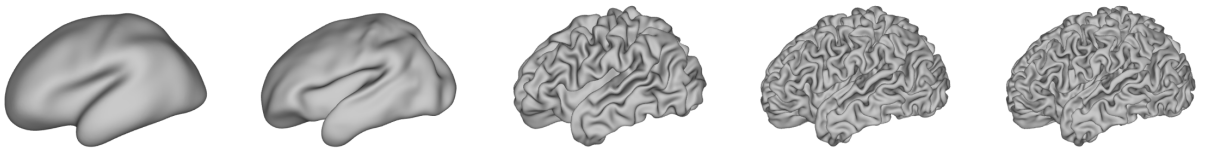


Figure 5.6: The deformed surfaces S^l generated by the multiscale deformation network for $l = 0, 1, ..., L$ with $L = 4$, where S^0 is the input template surface and S^L is the final predicted surface.

extract multiscale feature maps from the input clipped T2w image. Then, a sequence of multiscale volumetric SVFs $v^1, ..., v^L$ are predicted from the extracted feature maps. The SVFs v^l downsample the original brain MRI volume by a factor of 2^{L-l-1} for $l = 1, ..., L-1$. The last SVF v^L has the same size as the input volume to capture fine details. Each SVF v^l is associated with an ODE defined by Equation (5.1), which can be integrated to obtain the corresponding deformation field ϕ^l. The multiscale SVFs are predicted simultaneously so that the U-Net is only evaluated once during integration.

Our previous work CortexODE (Section 3.2), CoTAN (Section 4.1), and CoSeg (Section 4.2) adopted the forward Euler method to solve flow ODEs for cortical surface reconstruction. This requires a large number of integration steps to reduce numerical errors. In contrast, the multiscale deformation network predicts SVFs at different resolutions, enabling efficient ODE discretisation by the scaling and squaring method (Higham, 2005; Arsigny et al., 2006). For each scale $l = 1, ..., L$, we set the initial transformation as $\phi^l_{T/2^K} = id + \frac{T}{2^K} v^l$, and then the deformation field can be updated recurrently by $\phi^l_{T/2^{K-1}} = \phi^l_{T/2^K} \circ \phi^l_{T/2^K}$, where K is the steps of scaling and squaring. This is numerically accurate and equivalent to 2^K steps of forward Euler iteration. Nevertheless, the scaling and squaring method is not compatible with CortexODE, CoTAN and CoSeg, which are either modelled by Neural ODEs (Chen et al., 2018) or non-autonomous ODEs induced by time-varying velocity fields (TVF).

By integration, the network predicts multi-resolution diffeomorphic deformation fields ϕ^l, each of which has the same size as the corresponding SVF v^l. To further improve the

mesh quality of predicted cortical surfaces, we employ differentiable Gaussian smoothing filters with standard deviation $\sigma = 1.0$ to enhance the smoothness of each deformation field. Finally, we apply the deformation fields iteratively to an input surface $S^0 \subset \mathbb{R}^3$. The displacements of mesh vertices are sampled by trilinear interpolation. The surfaces are updated by $S^l = \phi^l(S^{l-1})$ for $l = 1, ..., L$. We set the number of scales to $L = 4$, which allows the multiscale deformation network to model coarse-to-fine surface deformations. The deformation fields ϕ^1 and ϕ^2 have been downsampled by a factor of 4 and 2 respectively. As shown in Figure 5.6, ϕ^1 and ϕ^2 contain global information and deform the input surface S^0 into coarse and smooth intermediate surfaces. ϕ^3 and ϕ^4, which have the same size as the input brain MRI, capture detailed cortical folds and provide further refinement for the final predicted white matter or pial surface S^L.

For white matter surface extraction, we use an initial template surface described in Figure 5.3 as the input surface for all subjects. The predicted white matter surfaces are further refined by Taubin smoothing (Taubin, 1995a,b) to improve the mesh quality, *i.e.*, the smoothness and regularity of the triangular mesh. For pial surface reconstruction, we use the predicted white matter surface as the input surface. Since the initial surface has genus-0 topology and the diffeomorphic deformations are topology-preserving, all reconstructed cortical surfaces have the same topology as a 2-sphere.

5.3.3.3 Model Training

We train four multiscale deformation networks for white matter and pial surface reconstruction for both brain hemispheres. We consider the following loss function:

$$\mathcal{L} = \mathcal{L}_{cd} + \lambda_{edge}\mathcal{L}_{edge} + \lambda_{nc}\mathcal{L}_{nc}, \tag{5.2}$$

where $\mathcal{L}_{cd}$ is the bidirectional Chamfer distance (Fan et al., 2017; Wang et al., 2018; Wickramasinghe et al., 2020), $\mathcal{L}_{edge}$ is the edge length loss (Wang et al., 2018; Wickramasinghe

et al., 2020), $\mathcal{L}_{nc}$ is the normal consistency loss (Wang et al., 2018; Wickramasinghe et al., 2020; Ma et al., 2023), λ_{edge} and λ_{nc} are weights for regularisation terms. The Chamfer distance $\mathcal{L}_{cd}$ is defined as the bidirectional distance between two point sets $\mathcal{X}, \mathcal{Y} \subset \mathbb{R}^3$:

$$\mathcal{L}_{cd}(\mathcal{X}, \mathcal{Y}) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \min_{y \in \mathcal{Y}} \|x - y\|^2 + \frac{1}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} \|x - y\|^2. \tag{5.3}$$

Specifically, $\mathcal{X}$ and $\mathcal{Y}$ are the sets of mesh vertices of the predicted and pseudo GT cortical surfaces respectively. The pseudo GT cortical surfaces are generated using the original dHCP pipeline (Makropoulos et al., 2018). All pseudo GT cortical surfaces are remeshed to have 150k vertices to reduce the errors introduced by point matching during the computation of the Chamfer distance. We adopt PyTorch3D (Ravi et al., 2020) which provides fast computation and gradient backpropagation for the Chamfer distance.

The edge length and normal consistency constraints are introduced to encourage the smoothness and improve the mesh quality of the predicted cortical surfaces. For a surface mesh $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$, where $\mathcal{V}, \mathcal{E}, \mathcal{F}$ are the sets for vertices, edges and faces respectively, the edge length loss is defined as:

$$\mathcal{L}_{edge}(\mathcal{M}) = \frac{1}{|\mathcal{E}|} \sum_{e_{ij} \in \mathcal{E}} \|x_i - x_j\|^2, \tag{5.4}$$

where x_i, x_j are two vertices connected by the edge e_{ij}. The normal consistency loss is defined as:

$$\mathcal{L}_{nc}(\mathcal{M}) = 1 - \frac{1}{|\mathcal{E}|} \sum_{f_i \cap f_j \in \mathcal{E}} n_i \cdot n_j. \tag{5.5}$$

Here n_i is the unit normal vector of the face $f_i \in \mathcal{F}$, and $f_i \cap f_j = e$ is the edge where the adjacent faces f_i and f_j intersect. The normal consistency loss penalises the dot product of the normal vectors between any two adjacent faces. The value of the loss is determined by the PMA of the neonates and the curvature of cortical surfaces. For older neonates with highly folded and curved cortical surfaces, there is a large difference between the normal orientations of two adjacent faces. Consequently, the normal consistency loss reaches a

high value to enforce surface smoothness. For younger subjects with immature cortex, the loss is lower as the cortical surfaces are already sufficiently smooth.

For the architecture of the multiscale deformation network, we set $L = 4$ as the number of scales as discussed in Section 5.3.3.2. A smaller L would offer insufficient global information, while a larger L is more suitable for brain MRI at higher image resolution. The numbers of channels for hidden layers are set to $\{16, 32, 64, 128, 128\}$ for white matter surface reconstruction and $\{16, 32, 32, 32, 32\}$ for pial surface to avoid overfitting. We use the same hyperparameters for both white matter and pial surface reconstruction. The regularisation weights are selected by grid search, and set to $\lambda_{edge} = 0.3$ and $\lambda_{nc} = 3.0$ to achieve the best trade-off between geometric accuracy and mesh quality on the validation set. The number of scaling and squaring steps is set to $K = 7$. Empirically, we use the Adam optimiser with learning rate 10^{-4} to train the multiscale deformation networks for 200 epochs. After training, we select the best models with minimum validation error, *i.e.*, minimum Chamfer distances between predicted and pseudo GT cortical surfaces.

5.3.4 Cortical Surface Inflation

Following the original dHCP pipeline (Makropoulos et al., 2018), our dHCP DL-based pipeline incorporates two types of cortical surface inflation. Firstly, the original dHCP pipeline employs Connectome Workbench (Glasser et al., 2013) to iteratively inflate and smooth the midthickness surface to generate inflated and very inflated surfaces. These surfaces are primarily used for visualisation. Our pipeline re-implements these surface inflation approaches using PyTorch to facilitate GPU acceleration. Secondly, the original dHCP pipeline adopts FreeSurfer *mris_inflate* command (Fischl et al., 1999a; Fischl, 2012), which is reproduced in the Medical Image Registration ToolKit (MIRTK) (Schuh, 2016; Schuh et al., 2017), to inflate the white matter surface while minimising the metric distortion between white matter and inflated surfaces. Such an inflation process is also accelerated by PyTorch with parallel computation on a GPU. During the inflation, the cor-

tical sulcal depth, or average convexity for each mesh vertex is computed by integrating the displacement of the vertex along its normal direction (Fischl et al., 1999a).

5.3.5 Cortical Feature Estimation

Based on the extracted cortical surfaces, our DL-based dHCP pipeline provides the estimation of cortical morphological features such as cortical thickness, curvature and sulcal depth, as well as functional features such as a myelin map. The estimation of cortical features is consistent with previous pipelines (Fischl, 2012; Glasser et al., 2013; Makropoulos et al., 2018). The cortical thickness is computed by averaging the bidirectional distances between white matter and pial surfaces. This is implemented with the SciPy package (Virtanen et al., 2020). Furthermore, the mean curvature of the white matter surface is computed via PyTorch-based re-implementation (Maillot et al., 1993; Glasser et al., 2013). The sulcal depth, or the mean convexity/concavity, is measured during cortical surface inflation as described in Section 5.3.4.

Following the convention of the HCP (Glasser et al., 2013) and original dHCP pipelines (Makropoulos et al., 2018), our DL-based pipeline estimates the myelin map based on the T1/T2 image ratio (Glasser and Van Essen, 2011). It should be noticed that the variation of T1/T2 contrast can also be attributed to other substantial factors apart from myelination, *e.g.*, the iron deposition. For the myelin map estimation, if the T1 image is provided and aligned to the T2 image, we divide the T1 image by the original T2 image before intensity normalisation to compute the T1/T2 ratio. Then, the myelin map is measured by projecting the T1/T2 ratio onto the midthickness surface using volume-to-surface mapping (Glasser and Van Essen, 2011; Glasser et al., 2013; Makropoulos et al., 2018), which is implemented with the Connectome Workbench. For each vertex of the midthickness surface mesh, we find its corresponding location in the T1/T2 ratio volume, and compute the Gaussian average of its neighbouring voxels as the myelin map value, where the size of the neighbour depends on the cortical thickness. Only voxels within the cortical ribbon

ROI are considered. However, it is computationally intensive to examine if a voxel or a point is between the inner and outer cortical surfaces, in particular for high resolution meshes with 150k vertices. To accelerate the volume-to-surface mapping, we train a 3D U-Net to learn the binary cortical ribbon mask, *i.e.*, the cortical GM segmentation. The learning procedure is similar to brain extraction as shown in Figure 5.4. The GT segmentation is generated by the Draw-EM approach integrated in the original dHCP pipeline.

5.3.6 Spherical Projection

The purpose of spherical projection or spherical mapping is to project the white matter surface onto a sphere while minimising metric distortions. Motivated by Zhao et al. (2022), we propose an unsupervised learning framework for spherical projection and integrate it into our DL-based dHCP pipeline. As illustrated in Figure 5.7, the inputs of our spherical projection framework include a white matter surface and a fixed initial sphere for all subjects. Note that since all white matter surfaces are deformed from the same initial template surface, the white matter surface has the same mesh connectivity as the initial sphere, which is generated from the initial surface by FreeSurfer (see Figure 5.3). Therefore, instead of inflating the white matter surface and projecting the inflated surface to create a sphere for each subject (Fischl et al., 1999a; Elad et al., 2005; Zhao et al., 2022), our approach learns to deform the initial sphere directly while minimising metric distortions between the deformed sphere and the white matter surface. In this work, we primarily consider edge and area distortions.

We resample the input white matter surface mesh such that it has the same connectivity as a standard icosphere with 163,842 vertices. This is achieved by barycentric interpolation (Robinson et al., 2014, 2018) between the initial sphere and the icosphere, of which the barycentric coordinates are pre-computed to accelerate the resampling. Then, we compute the metrics of the resampled white matter surface including edge length and face area. Each vertex contains the metrics of its adjacent edges and faces. We concatenate the

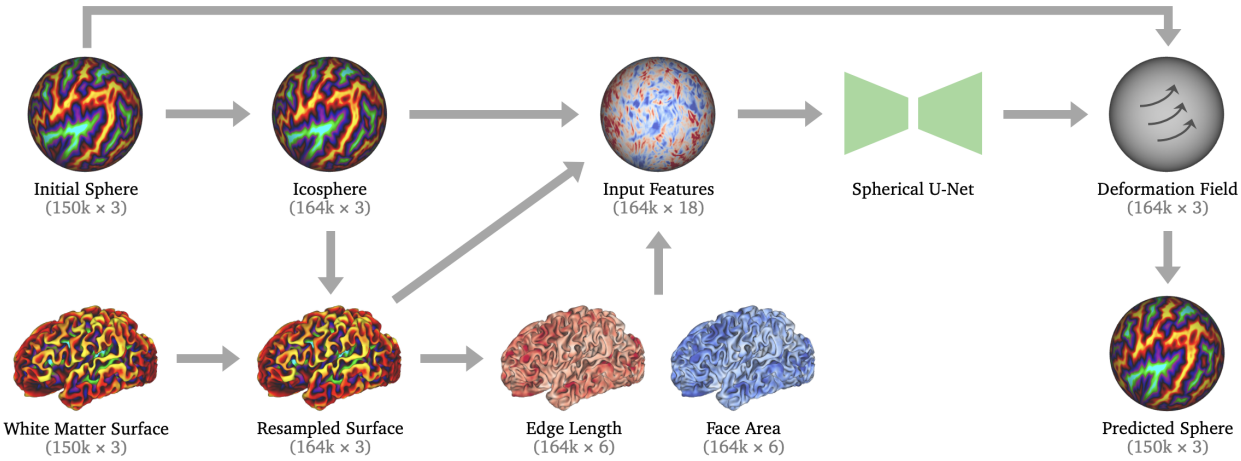


Figure 5.7: Our unsupervised learning-based spherical projection framework. Given an input white matter surface and an initial sphere, we resample the white matter surface mesh such that it has the same connectivity as a standard icosphere with 164k vertices. Then, we compute the metrics including edge length and face area of the resampled surface mesh, of which each vertex contains the metrics of its adjacent edges or faces. The computed metrics are concatenated with the coordinates of the resampled surface and the icosphere as input features. A Spherical U-Net learns from the input features to predict a diffeomorphic deformation field, which deforms the initial sphere to a predicted sphere. An unsupervised loss function is defined as the metric distortion between the predicted sphere and the original white matter surface.

coordinates of the resampled white matter surface and the icosphere, as well as computed metrics as input features of a Spherical U-Net (Zhao et al., 2019a), which has been widely used in cortical surface tasks such as cortical surface registration (Zhao et al., 2021a), parcellation (Zhao et al., 2019b) and development prediction (Zhao et al., 2021b). We train the Spherical U-Net to learn a diffeomorphic deformation field in the spherical domain by integrating a SVF using scaling and squaring method (Higham, 2005; Arsigny et al., 2006). Rather than adding regularisation terms to the loss function (Zhao et al., 2022), we apply Laplacian mesh smoothing (Desbrun et al., 1999) to the spherical deformation field to encourage smoothness. The predicted sphere is derived by deforming the initial sphere, where the displacement of each vertex is sampled from the deformation field using barycentric interpolation.

Zhao et al. (2022) employed multi-resolution Spherical U-Nets and computed the losses between the deformed icosphere and the resampled white matter surface with 164k ver-

tices. In contrast, our approach minimises the metric distortions end-to-end between the predicted sphere and the original input white matter surface. Hence, a single-scale Spherical U-Net is sufficient to achieve accurate results. Inspired by Zhao et al. (2022), we train the Spherical U-Net by minimising an unsupervised metric distortion loss, which is defined as the minimum root mean square deviation (RMSD) between the metrics of predicted sphere and the original white matter surface:

$$\mathcal{L}(X_{sphere}, X_{surf}) = \min_{\beta \in \mathbb{R}_+} \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(\beta X^i_{sphere} - X^i_{surf}\right)^2}, \tag{5.6}$$

where $X^i_{sphere}, X^i_{surf} \in \mathbb{R}^N$ are the metrics of the i-th vertex and β is a scaling coefficient. The loss (5.6) achieves minimum when

$$\beta = \sum_{i=1}^{N} \left(X^i_{sphere} X^i_{surf}\right) / \sum_{i=1}^{N} \left(X^i_{sphere}\right)^2. \tag{5.7}$$

The metrics in Equation (5.6) refer to the mesh edge length or face area. The overall loss function is defined as $\mathcal{L} = \lambda_e \mathcal{L}_e + \lambda_a \mathcal{L}_a$, where $\mathcal{L}_e, \mathcal{L}_a$ are the edge and area distortions, and λ_e, λ_a are the weights. For model training, we use the Adam optimiser with learning rate 10^{-4} to train for 200 epochs. The weights of losses are set to $\lambda_e = 1.0$ and $\lambda_a = 0.5$. After training, we select the model with lowest metric distortions on the validation set.

However, the proposed unsupervised learning-based spherical projection framework only considers local metric distortions such as edge length and face areas, whereas FreeSurfer (Fischl, 2012) and the original dHCP pipeline (Makropoulos et al., 2018) minimize global distortions of long-range geodesic distances. Therefore, besides unsupervised spherical mapping, we train another Spherical U-Net to minimize the geodesic distortions, supervised by ground truth spheres generated by Spherical MDS (Elad et al., 2005). As an alternative option, the supervised learning-based spherical projection is also involved in the release of our DL-based dHCP pipeline, such that the projected spheres match the results of the original dHCP pipeline.

5.4 Comparison to Existing Neuroimage Pipelines

5.4.1 Comparison to Original dHCP Pipeline

We compare our DL-based dHCP pipeline to the original dHCP pipeline (Makropoulos et al., 2018) with respect to the approach, implementation and runtime. The comparison of surface and sphere quality are provided in the Section 5.5.

Approach. We compare the different approaches used for main processing steps between the original and our DL-based dHCP pipelines in Table 5.1. Our DL-based pipeline integrates novel learning-based approaches for the crucial processing steps, and predicts cortical surfaces end-to-end without the need of tissue segmentation.

	Original dHCP Pipeline	DL-based dHCP Pipeline (Ours)
Brain Extraction	FSL BET (Smith, 2002)	Supervised Learning + U-Net
Tissue Segmentation	Draw-EM (Makropoulos et al., 2014)	—
Surface Reconstruction	Deformable Model (Schuh et al., 2017)	Multiscale Deformation Networks
Spherical Projection	Spherical MDS (Elad et al., 2005)	Unsupervised Learning + Spherical U-Net

Table 5.1: Comparisons of different approaches used for brain extraction, tissue segmentation, cortical surface reconstruction and spherical projection between original dHCP pipeline (Makropoulos et al., 2018) and our DL-based dHCP pipeline.

Implementation. The original dHCP pipeline is implemented with MIRTK (Schuh, 2016), Connectome Workbench (Glasser et al., 2013), and FSL (Jenkinson et al., 2012). Since the installation and configuration of the MIRTK package are complicated, it is difficult to reproduce or modify the original dHCP pipeline. For our DL-based dHCP pipeline, the implementation of each processing step is described in Table 5.2. Our DL-based dHCP pipeline is primarily implemented in Python and PyTorch (Paszke et al., 2019), which simplify deployment, enhance reproducibility, and allow for more straightforward modifications and adaptations. The PyTorch library provides seamless support for parallel computation and acceleration on GPUs (Keckler et al., 2011). The PyTorch3D package (Ravi et al., 2020) is required for the training of cortical surface reconstruction and provides utilities for mesh-based processing. ANTsPy (Avants et al., 2009) is used for N4

bias correction and affine registration. The Connectome Workbench (WB) is used for visualisation and myelin map estimation. Our DL-based pipeline is GPU memory-efficient, which only uses 12GB for training and 7.6GB GPU memory for inference. Therefore, it is simple to adapt the pipeline to a new dataset by fine-tuning the neural network models.

Processing Steps	Implementation	GPU	CPU only
Brain Extraction	PyTorch	0.25s	1.63s
Bias Field Correction	ANTsPy	2.62s	2.64s
Affine Registration	ANTsPy	2.76s	2.77s
Surface Reconstruction	PyTorch3D	2.37s	37.01s
Surface Inflation	PyTorch	4.99s	21.28s
Spherical Projection	PyTorch	0.68s	1.35s
Cortical Thickness	SciPy	8.73s	102.15s
Mean Curvature	PyTorch		
Sulcal Depth	PyTorch		
Myelin Map	WB & PyTorch		
Total Runtime		23.98s	170.29s

Table 5.2: The implementation and runtime of the proposed dHCP DL-based pipeline. Both GPU-accelerated and CPU-only runtime (without GPU acceleration) are reported. In addition to listed processing steps, the total runtime also involves extra processing time such as file I/O operations.

Runtime. To measure the runtime, we run both original and DL-based dHCP pipeline to process the brain MRI of 10 dHCP neonatal subjects, which are randomly selected from the test set. The runtime is measured on an 8-core Intel Core i7-11700K CPUs and a NVIDIA GeForce RTX 3080 GPU with 12GB memory. The original dHCP pipeline (Makropoulos et al., 2018) runs exclusively on CPUs and requires 6h38min to process a single subject. The volume-based and surface-based processing requires 2h9min and 4h29min respectively. For our DL-based pipeline, we report the runtime with and without GPU acceleration in Table 5.2. With GPU acceleration, it only takes 24 seconds to run the entire DL-based pipeline, which is 995× faster than the original dHCP pipeline. The cortical surface reconstruction and spherical projection only needs 3 seconds in total owing to the powerful end-to-end deep learning approaches. Even without GPU acceleration, our DL-based pipeline only requires 170 seconds to complete on the CPU, which is still 140× faster compared to the original dHCP pipeline.

5.4.2 Comparison to DL-based Neuroimage Pipelines

We compare our dHCP DL pipeline to existing DL-based neuroimage processing pipelines, including FastSurfer (Henschel et al., 2020) and iBEAT V2.0 (Wang et al., 2023). The comparisons regarding functionality and approximate runtime on a GPU are reported in Table 5.3. Table 5.3 shows that previous DL-based pipelines focus on learning-based approaches for brain regional and tissue segmentations. However, theses pipelines still adopt traditional approaches for cortical surface reconstruction including computationally intensive topology correction (Bazin and Pham, 2005, 2007; Ségonne et al., 2007), and thus require more than one hour to process a single brain MRI scan. Our DL-based dHCP pipeline introduces fast learning-based methods to accelerate the most time-consuming cortical surface reconstruction and spherical projection. Therefore, our pipeline is 150× and 600× faster than FastSurfer and iBEAT V2.0 respectively. Our DL-based pipeline extracts cortical surfaces end-to-end from the T2w brain MRI without the need of intermediate tissue segmentations. This prevents the propagation of errors from segmentations to cortical surfaces. Therefore, as shown in Table 5.3, currently our pipeline does not provide brain tissue segmentations and cortical surface parcellation. We will develop corresponding DL-based approaches and integrate them into our pipeline in future work.

Processing Steps	FastSurfer	iBEAT V2.0	Ours
Preprocessing		✓	✓
Brain Extraction		✓	✓
Brain Segmentation	✓	✓	
Surface Reconstruction	✓	✓	✓
Surface Parcellation	✓	✓	
Spherical Projection	✓		✓
Feature Estimation	✓	✓	✓
Approx Runtime	1h	4h	**24**s

Table 5.3: Comparisons of functionality and runtime across DL-based neuroimage processing pipelines: FastSurfer (Henschel et al., 2020), iBEAT V2.0 (Wang et al., 2023), and our dHCP DL pipeline. The green check marks indicate DL-based processing steps. We provide the approximated runtime as reported in original papers.

5.5 Quality Control

5.5.1 Cortical Surface Quality

5.5.1.1 Visual Inspection

We conduct manual quality control (QC) via visual inspection to assess the quality of the cortical surfaces generated by our DL-based dHCP pipeline. The cortical surface quality, *i.e.*, the anatomical correctness of cortical surfaces, is measured by the number and severity of anatomical errors and corruptions in cortical surfaces. Due the lack of GT cortical surfaces, we use the cortical surface generated by the original dHCP pipeline (Makropoulos et al., 2018) as the baseline. More specifically, our visual QC aims to examine if our DL-based dHCP pipeline extracts cortical surfaces with better surface quality compared to the original dHCP pipeline.

We randomly select 30 T2w brain MRI samples for QC from 265 test cases. For each test brain MRI, we extract two groups of cortical surfaces by original dHCP pipeline and our DL-based dHCP pipeline respectively. The visual QC is performed by four expert assessors with clinical background or extensive experience in neuroimage analysis to mitigate individual variability. To ensure fairness, all assessors visually evaluate two groups of cortical surfaces independently without knowing the corresponding reconstruction approaches (either original or DL-based pipeline). The assessors are requested to carefully examine and detect corrupted areas in two groups of cortical surfaces, and then select the group with better surface quality for each test sample based on the number and severity of identified defects and artefacts. Equal quality is also an option when both groups of cortical surfaces are anatomically correct or corrupted to the same degree.

We further provide visual comparison between cortical surfaces reconstructed by original and DL-based dHCP pipeline. Figure 5.9 displays the cortical surfaces of 5 different samples. It indicates that the original dHCP pipeline is prone to produce corruptions

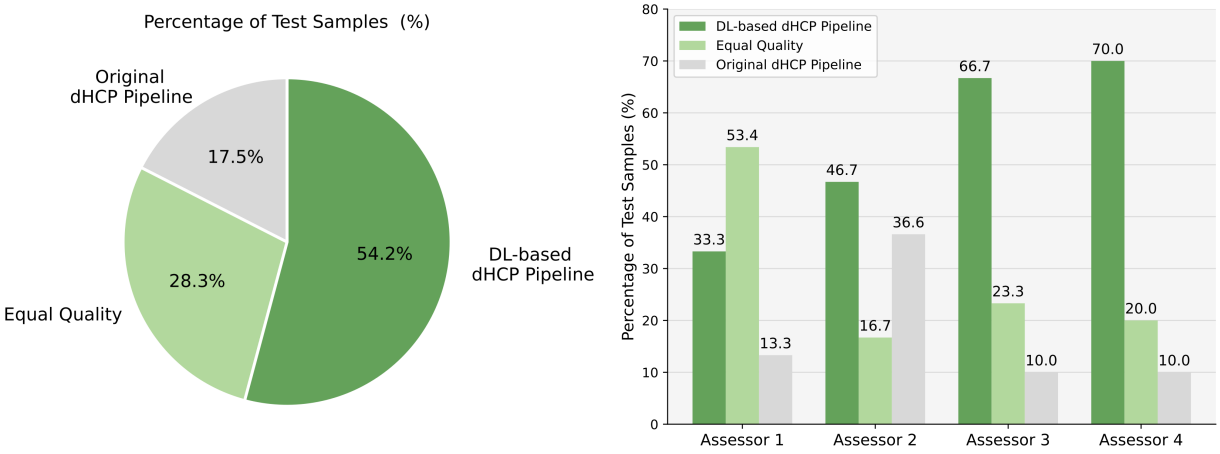


Figure 5.8: The results of manual cortical surface quality control (QC). The assessors visually examine and compare the anatomical correctness of cortical surfaces between the original and our DL-based dHCP pipelines for each test brain MRI. Left: the average surface QC results. For 82.5% of all test samples, our DL-based dHCP pipeline achieves superior (54.2%) or equal (28.3%) surface quality compared to the original dHCP pipeline. Right: the individual surface QC results from all assessors.

on the top and back (parietal and occipital lobes) of the cortical surfaces, due to the low image contrast in the corresponding regions of T2w brain MRI, whereas our DL-based pipeline achieves better anatomical correctness in these regions compared to original dHCP pipeline. In addition, the surfaces generated by the original dHCP pipeline are affected by the imprecise Draw-EM segmentations (Makropoulos et al., 2014, 2018), and thus occasionally produce unexpected "caves" or "holes" as shown in Figure 5.9. Our DL-based pipeline leverages diffeomorphic surface deformations and learns cortical surfaces end-to-end from brain MRI, which can effectively avoid such issues.

5.5.1.2 Brain Development

In addition to manual quality control, we provide qualitative and quantitative evaluation to validate that the cortical surfaces extracted by our dHCP DL pipeline can capture the brain development of neonates. We visualise the cortical surfaces and the corresponding cortical features of different neonatal subjects scanned at the PMA of $\{29, 34, 39, 44\}$ weeks in Figure 5.10. The cortical folding is observed in Figure 5.10 with the increasing of the

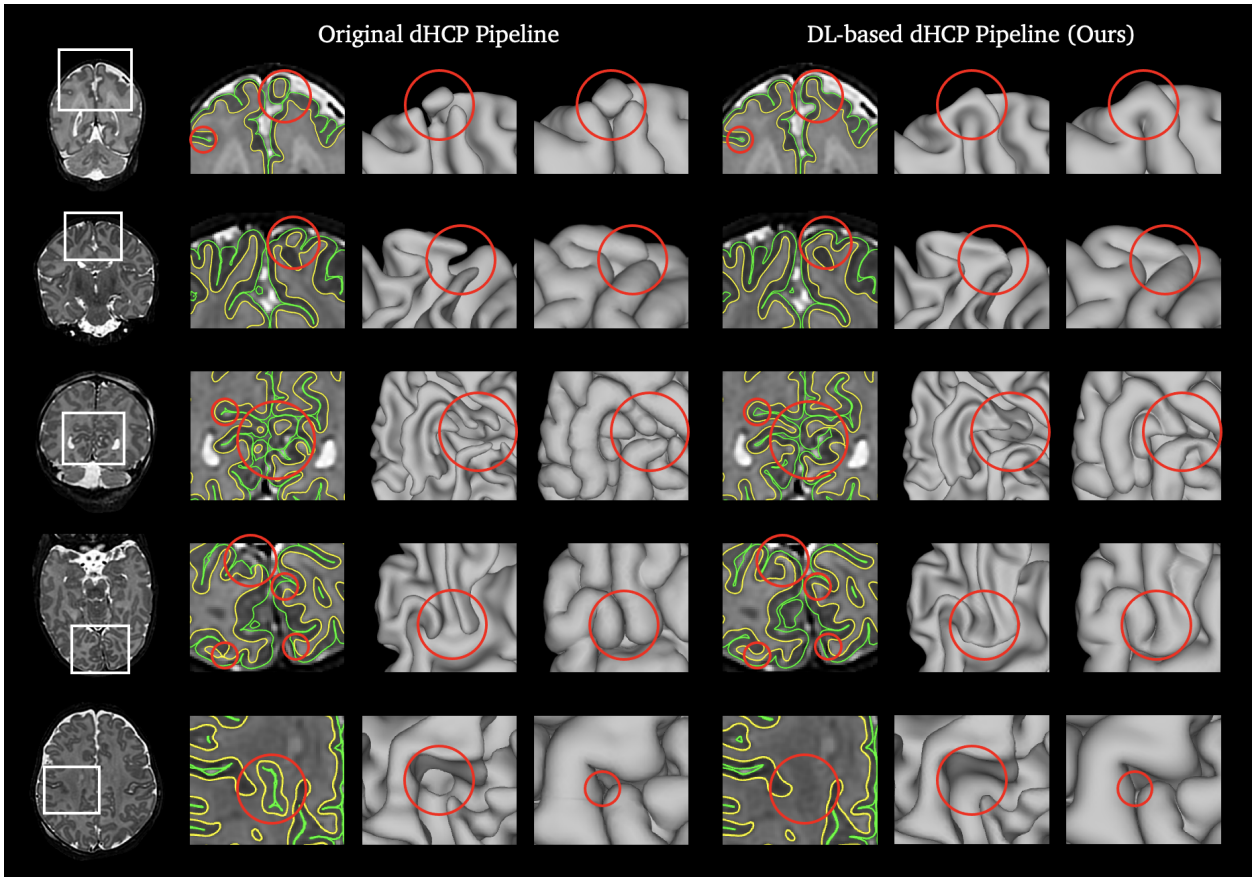


Figure 5.9: The qualitative comparison between original and our DL-based dHCP pipeline on 5 different test samples of the dHCP dataset. The left columns visualise the corruptions in cortical surfaces produced by the original dHCP pipeline, while the right columns show that our DL-based dHCP pipeline effectively reduces these errors. Row 1-2: corruptions on the top of the cortical surfaces. Row 2-4: corruptions on the back of the cortical surfaces. Row 5: the "cave/hole" introduced by the errors of WM segmentations.

PMA. The cortical thickness and sulcal depth are increasing as well with brain growth. We further measure and plot cortical morphological features including cortical volume, surface area, cortical thickness and sulcal depth across all 265 test neonatal subjects in Figure 5.11. The cortical volume is equal to the volumes of pial surfaces minus those of the white matter surfaces for both brain hemispheres. The surface area refers to the mesh area of the pial surfaces. The cortical thickness and sulcal depth for each subject is measured by averaging the cortical thickness and the absolute value of the sulcal depth over all vertices on the surface meshes. Figure 5.11 presents the positive trends of the morphological features with increasing PMA, which reflect the growth and folding of the neonatal cerebral cortex. Moreover, we compute the Pearson correlation coefficients

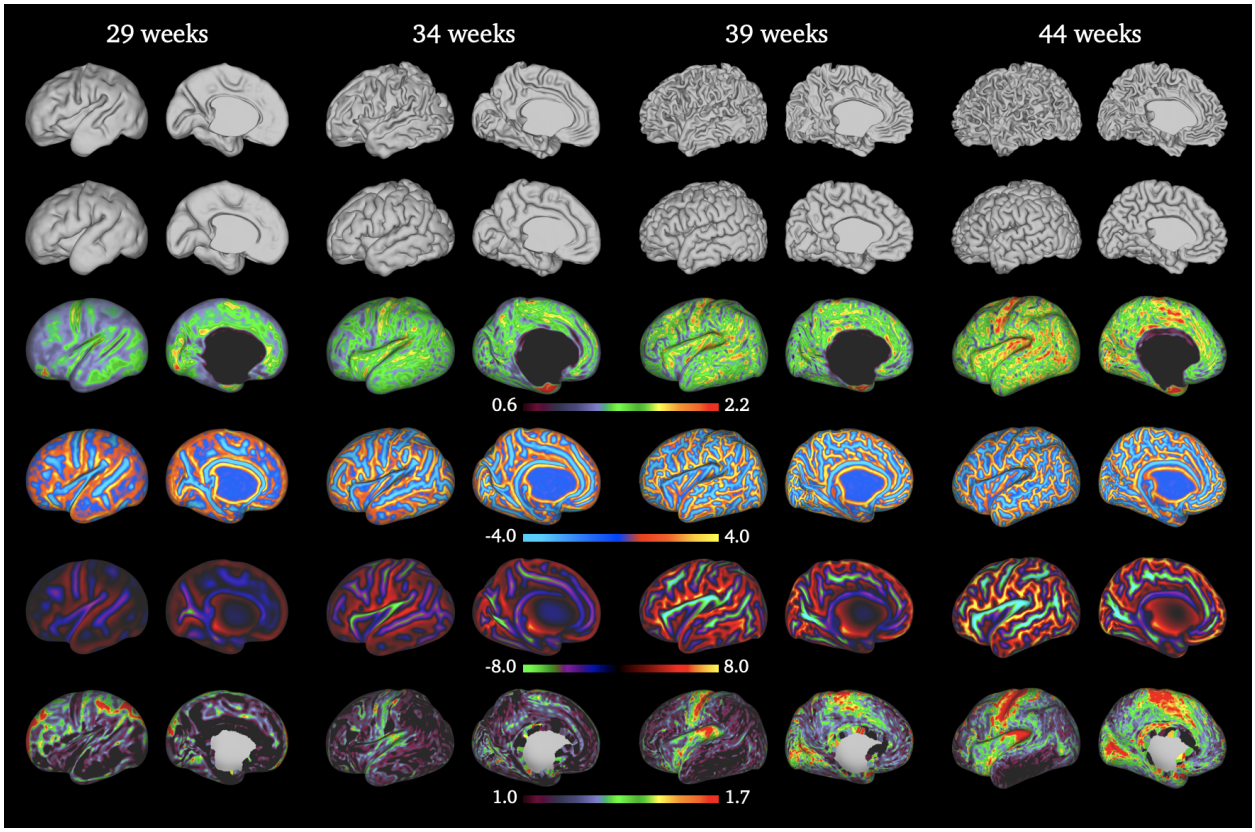


Figure 5.10: The visualisation of cortical surfaces and cortical features of different neonatal subjects at the PMA of 29,34,39,44 weeks. From top to bottom: white matter surface, pial surface, cortical thickness, mean curvature, sulcal depth, and myelin map.

(PCC) r to measure the correlation between the morphological features and the PMA. The PCCs are provided in Figure 5.11, which demonstrate that all cortical morphological features have strong positive correlation with the age of neonates. This verifies that the cortical surfaces extracted by our DL-based pipeline are able to capture the development of neonatal brains.

5.5.1.3 Surface Topology

Our DL-based dHCP neonatal pipeline is able to guarantee the spherical topology of cortical surfaces. The proposed pipeline uses multiscale deformation networks to learn diffeomorphic deformations to deform an initial surface to cortical surfaces. As the diffeomorphic deformations preserve the topology of the surfaces and the initial surface has the

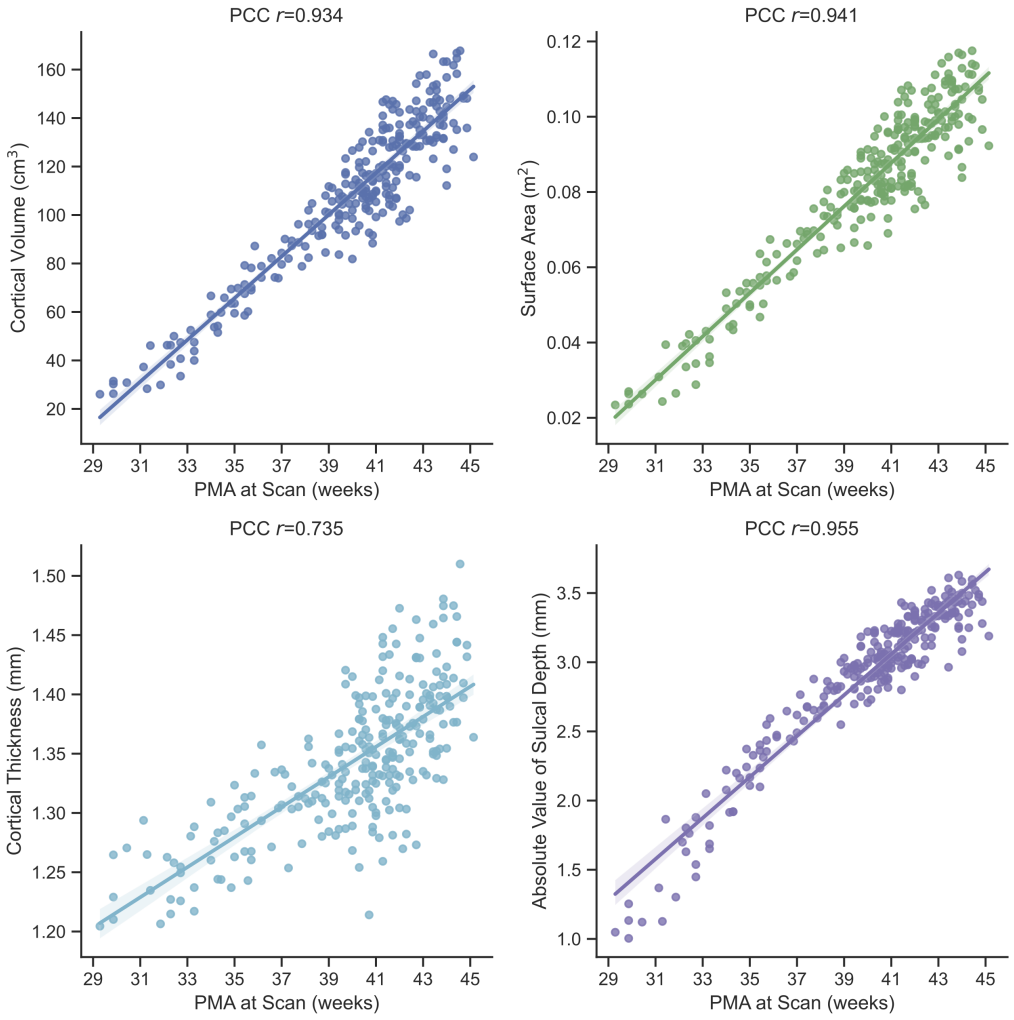


Figure 5.11: The cortical morphological features including cortical volume (cm^3), surface area (m^2), average cortical thickness (mm), and average absolute value of sulcal depth (mm) across 265 test neonatal subjects with different PMA at scan. The Pearson correlation coefficients (PCC) r are reported to measure the correlation between the morphological features and the PMA.

same topology as a sphere, the cortical surfaces are topologically equivalent to a sphere as well. The Euler characteristic number is $\chi = 2$ for all extracted surface meshes as expected. We further examine surface self-intersections, which usually appear in the narrow sulci of pial surfaces and will affect the estimation of cortical morphological features (Ma et al., 2022). We detect and count the number of self-intersecting faces (SIFs) in white matter and pial surface meshes generated by both original and DL-based dHCP pipelines. To verify that the proposed multiscale deformation networks can effectively reduce SIFs, we conduct ablation experiments by removing the Gaussian smoothing layer or replacing

the scaling and squaring method (Higham, 2005; Arsigny et al., 2006) with the forward Euler method for integration. The number of SIFs are reported in Table 5.4.

Table 5.4 shows that our DL-based dHCP pipeline produces only 0.18 SIFs on average in white matter surfaces. The original dHCP pipeline produces marginally fewer SIFs in pial surfaces than ours since it performs collision detection during the iterative surface deformation. The ablation study also validates that SIFs can be reduced effectively by adding the Gaussian smoothing layers and using the scaling and squaring method. Compared to CortexODE (Chapter 3) and CoTAN (Chapter 4), the proposed multiscale deformation network shows notably higher mesh quality with only 0.0005% of SIFs in pial surface meshes, *i.e.*, 1.5 SIFs within 300k faces. This is orders of magnitude less than CortexODE and CoTAN, which produce 0.28% and 0.07% of SIFs respectively in pial surfaces on the same dHCP neonatal dataset.

Method	White Matter Surface	Pial Surface
Original dHCP pipeline	0.457±7.409	**0.517±7.418**
Forward Euler	0.864±5.712	4.925±17.71
w/o Gaussian smoothing	0.613±4.304	3.715±9.824
Ours	**0.179±3.171**	1.500±8.875

Table 5.4: The number of self-intersecting faces (SIFs) on cortical surfaces.

5.5.1.4 Generalisability

In addition to the dHCP neonatal dataset (Edwards et al., 2022), we evaluate the generalisability of our DL-based dHCP pipeline on an unseen dataset. We consider the Evaluation of Preterm Imaging (ePrime) dataset (Edwards et al., 2018; Grigorescu et al., 2021), comprising 486 T2w MRI scans of neonates with gestational age between 23 and 33 weeks, in which 444 subjects are scanned at term-equivalent age between 38 and 45 weeks. The MR images are acquired with a Philips Intera 3T scanner, using a T2w turbo echo sequence (TES) with parameters repetition/echo time TR/TE=8670/160ms and TSE factor of 16. The ePrime dataset has relatively low image resolution with the voxel size of

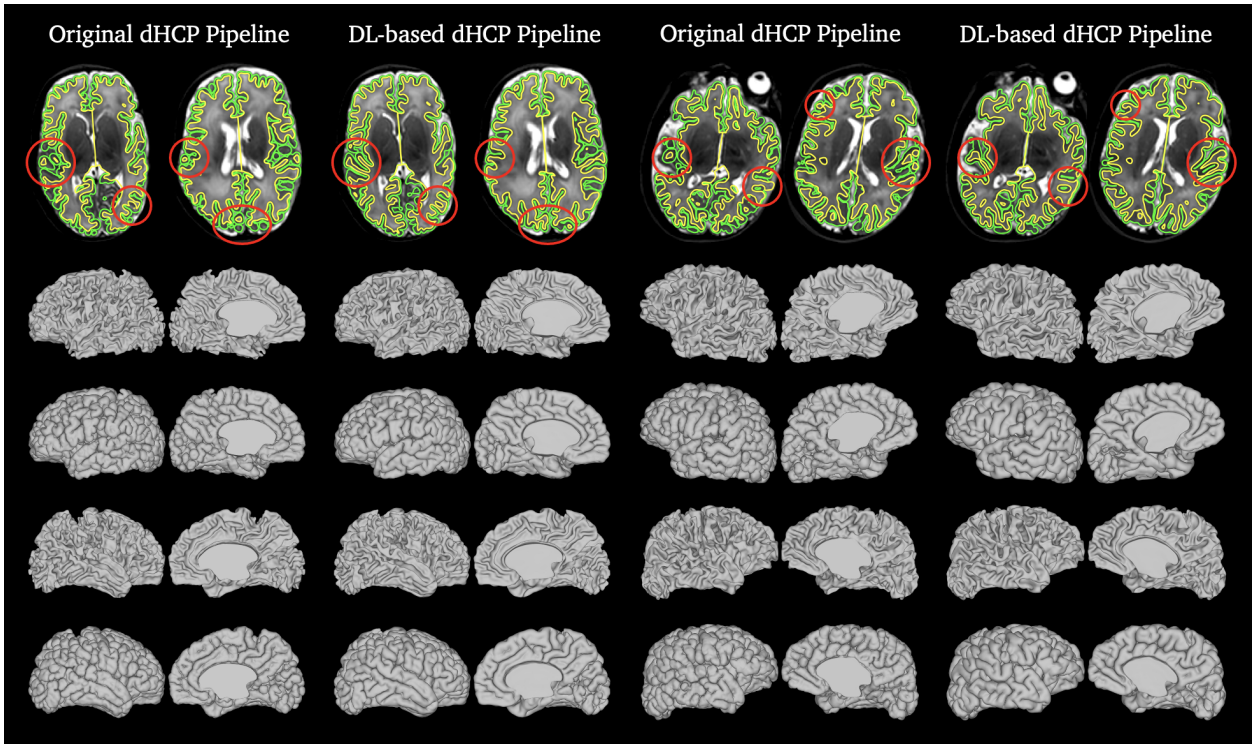


Figure 5.12: The qualitative comparison between original and our DL-based dHCP pipeline on two neonatal subjects in the ePrime dataset. The cortical surfaces generated by original dHCP pipeline are severely corrupted and broken on gyri, whereas our DL-based pipeline has better generalisability. From top to bottom: T2w MRI, left white matter surface, left pial surface, right white matter surface, and right pial surface.

$0.86 \times 0.86 \times 1.0 mm^3$, and thus it is challenging to extract high-quality cortical surfaces. To evaluate the generalisability, we run both original and our DL dHCP pipelines on unseen T2w MRI scans of the ePrime dataset without fine-tuning. The qualitative comparison in Figure 5.12 shows that there are severe corruptions and discontinuity on the gyri of cortical surfaces generated by the original dHCP pipeline, while our DL-based pipeline produces cortical surfaces with acceptable quality. This demonstrates that our DL-based pipeline has better generalisability on the unseen and low-resolution dataset.

5.5.2 Sphere Quality

We evaluate the performance of the spherical projection by the metric distortions defined in Equation (5.6) on 265 test samples in the dHCP neonatal dataset. Similar to the orig-

Method	Edge (mm)	Area (mm^2)	Runtime
Spherical MDS	0.186±0.039	0.064±0.020	74min
FastSurfer	0.284±0.064	0.106±0.033	2.73s
FreeSurfer	0.123±0.025	0.033±0.010	4min12s
Initial Sphere	0.165±0.033	0.055±0.017	–
Zhao et al. (2022)	0.139±0.026	0.035±0.010	2.11s
Ours	**0.119±0.023***	**0.023±0.006***	**0.33s***

Table 5.5: Comparisons of metric distortions between spherical projection approaches. The edge distortion (mm), area distortion (mm^2) and runtime for each brain hemisphere are reported. *Our learning-based approach produces significantly smaller distortions compared to all other approaches ($p < 0.05$).

inal dHCP pipeline (Makropoulos et al., 2018), the quality of the sphere is measured by the distortions of the edge length and face area between the white matter surface and the projected sphere. We compare our unsupervised learning-based approach with existing spherical mapping methods including Spherical MDS (Elad et al., 2005), FreeSurfer (Fischl, 2012), FastSurfer (Henschel et al., 2020), and Zhao et al. (2022). We also provide the metric distortions between the white matter surface and the initial sphere as a baseline. For fair comparison, we only use a single-scale Spherical U-Net in Zhao et al. (2022) like our approach. The edge and area distortions, as well as the average runtime for each brain hemisphere are reported in Table 5.5. We conduct paired t-tests to examine if our unsupervised learning approach performs significantly better than baseline methods.

Table 5.5 shows that our approach achieves significantly better performance ($p < 0.05$) compared to all baseline methods and only requires 0.33 seconds of runtime. As reported in Table 5.5, the runtime of Spherical MDS (Elad et al., 2005) is more than one hour due to the time-consuming pairwise geodesic distance computation. Despite the efficiency of the spherical embedding approach in FastSurfer (Henschel et al., 2020), it does not optimise the metric distortions. FreeSurfer (Fischl, 2012) achieves satisfactory performance but it takes 4 minutes to process each hemisphere. The initial sphere, which is generated by FreeSurfer as a good starting point, effectively reduces the baseline metric distortions without additional deformations.

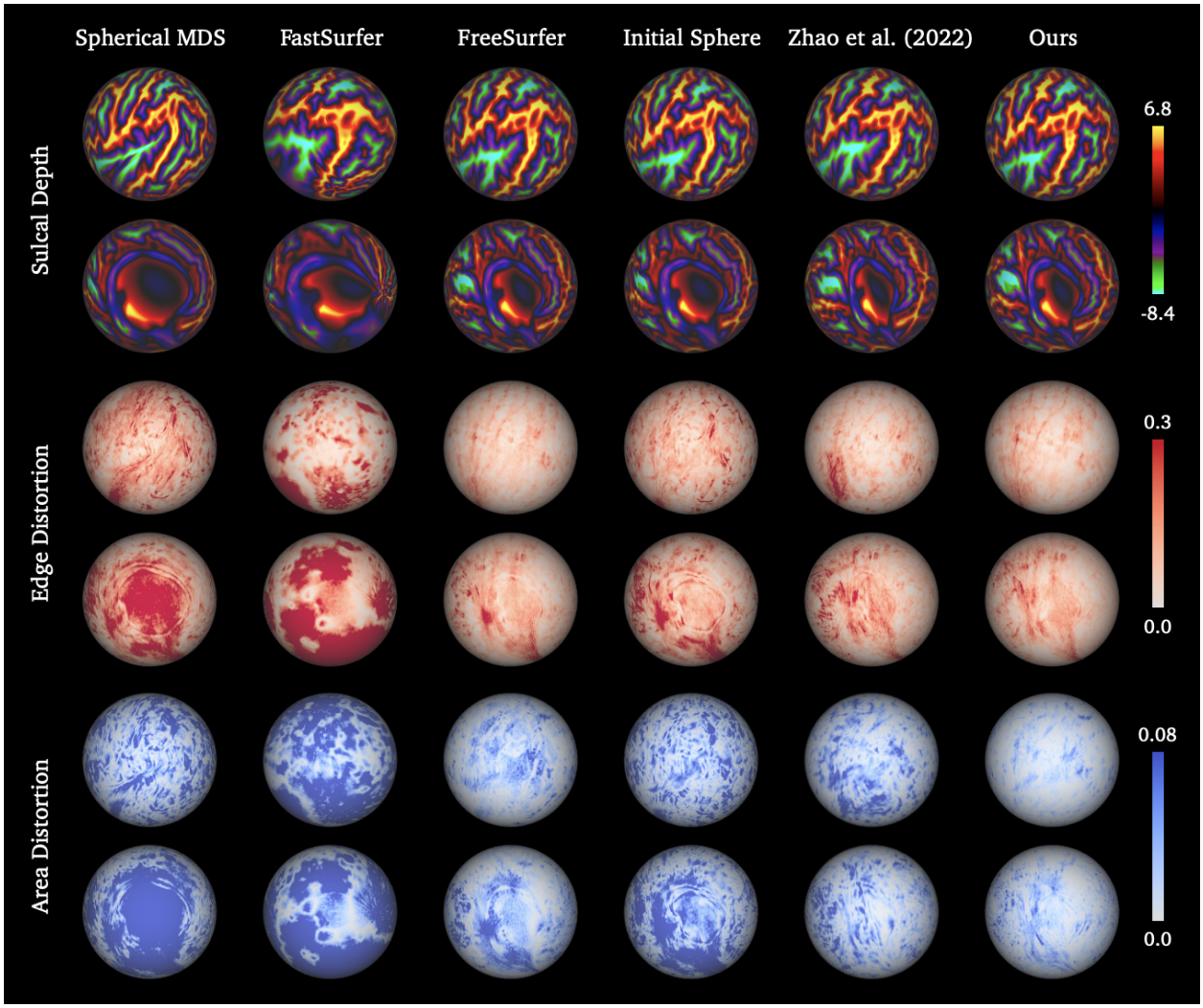


Figure 5.13: The visualisation of projected spheres for all spherical mapping approaches. The projected spheres are overlaid with sulcal depth, edge and area distortions from top to bottom. For each vertex, the distortion is computed by averaging over the distortions of its adjacent edges or faces.

Compared to Zhao et al. (2022), our approach has several advantages. First, Zhao et al. (2022) needs to inflate and project the white matter surfaces to create initial spheres, which requires 1.79 seconds for each brain hemisphere with GPU acceleration in our experiments. Instead, our approach uses an fixed initial sphere for all subjects without extra processing. The initial sphere is generated by FreeSurfer with minimised metric distortions as reported in Table 5.5. This provides a strong prior with small initial distortions for the training. Such a global initial surface also saves the runtime of interpolation and resampling, since the barycentric coordinates can be pre-computed. Hence, our approach

is capable of projecting the cortical surface to a sphere in 0.33 seconds, which is 6× faster than Zhao et al. (2022). Furthermore, Zhao et al. (2022) computes the metric distortions between the resampled white matter surface and the icosphere, and thus requires multi-scale networks to improve the performance. As reported in Table 5.5, Zhao et al. (2022) does not outperform FreeSurfer in the case of single-scale architecture. Our end-to-end metric distortion loss is more effective so that a single-scale Spherical U-Net is sufficient to produce significantly smaller distortions than FreeSurfer.

We also provide qualitative comparison as shown in Figure 5.13, which visualises the projected spheres overlaid with cortical sulcal depth, edge and area distortions for all approaches. Figure 5.13 shows that Spherical MDS and FastSurfer produce large distortions for edge length and face area. Compared to FreeSurfer, which is the best baseline as reported in Table 5.5, our unsupervised approach achieves similar visualisation of sulcal depth, similar edge distortions, much smaller area distortions, and 760× faster runtime.

5.6 Discussion

In this chapter, we presented a fast and complete DL-based cortical surface reconstruction pipeline for structural brain MRI processing and analysis on the dHCP neonatal dataset. The proposed DL-based dHCP pipeline, which is primarily implemented in Python and PyTorch, consists of learning-based brain extraction, cortical surface reconstruction and spherical projection, as well as GPU-accelerated cortical surface inflation and feature estimation. Instead of utilising the DL-based cortical surface reconstruction approaches introduced in previous chapters, we developed a multiscale diffeomorphic deformation network to improve the anatomical correctness and mesh quality of reconstructed cortical surfaces. Our DL-based dHCP pipeline provides rapid and accurate brain MRI preprocessing, cortical surface extraction, and downstream cortical surface analysis for the community of neuroimaging research. The entire pipeline executes within only 24 seconds

accelerated by GPU and takes less than 3 minutes even without GPU support. This is orders of magnitude faster than traditional neuroimage pipelines (MacDonald et al., 2000; Han et al., 2004; Fischl, 2012; Glasser et al., 2013; Makropoulos et al., 2018) that require 6 hours of runtime, as well as existing DL-based cortical surface reconstruction pipelines (Henschel et al., 2020; Wang et al., 2023) that need more than one hour.

The results of visual surface quality control validate that for more than 80% of test samples, our DL-based dHCP pipeline produces cortical surfaces with higher (54.2%) or equal (28.3%) surface quality compared to the original dHCP pipeline (Makropoulos et al., 2018). Particularly, the visual inspection shows that our learning-based cortical surface reconstruction approach can effectively reduces the artefacts on the top and occipital lobe of the cortical surfaces produced by original dHCP pipeline. We also verify that the proposed DL-based dHCP pipeline is capable of generalising well on unseen and low-resolution ePrime neonatal brain MRI data (Edwards et al., 2018; Grigorescu et al., 2021). In addition, our unsupervised learning-based spherical projection framework significantly reduces both edge and area distortions compared to previous work while only requires 0.33 seconds of runtime for each brain hemisphere.

However, despite the high efficacy and efficiency of the proposed dHCP DL pipeline, the anatomical correctness of predicted cortical surfaces is inevitably restricted by the surface quality of pseudo GT cortical surfaces generated by the original dHCP pipeline, due to the limitations of supervised learning. Besides, the proposed multiscale deformation network predicts cortical surfaces from the input T2w brain MRI directly without the reliance on segmentations. Such an end-to-end framework facilitates the surface reconstruction procedure and avoids the potential errors in cortical surfaces introduced by imprecise segmentations, while this also means that our dHCP DL pipeline does not provide brain tissue segmentations and cortical surface parcellation.

In future work, semi-supervised learning techniques could be considered to further enhance the anatomical correctness of cortical surfaces. In addition to pseudo GT cortical

surfaces, unsupervised brain MRI intensity could be leveraged to deform the cortical surfaces towards the WM/cGM or cGM/CSF boundary where the MRI achieves maximum image contrast. Furthermore, the functionality of our DL-based dHCP pipeline could be enriched by incorporating more GPU-accelerated processing steps, *e.g.*, brain regional and tissue segmentations, cortical parcellation, and local gyrification index estimation.

Chapter 6

Conclusion

This thesis presented a series of deep learning (DL) based frameworks for cortical surface reconstruction from brain MR images. We introduce explicit and diffeomorphic surface deformations to reconstruct cortical surfaces with high geometric accuracy and surface quality. Our approaches accelerate the traditional cortical surface reconstruction pipelines (Fischl, 2012; Glasser et al., 2013; Makropoulos et al., 2018) from several hours of runtime to only a few seconds. This facilitates fast cortical surface-based brain MRI processing in large-scale neuroimaging studies (Marcus et al., 2007, 2010; Jack Jr et al., 2008; Van Essen et al., 2013; Alfaro-Almagro et al., 2018; Howell et al., 2019; Edwards et al., 2022). The proposed frameworks have been fully validated on brain MRI datasets involving subjects in various age groups, from fetuses (22-38 weeks) and neonates (25-45 weeks), to young (22-36 years) and old adults (55-90 years). In conclusion, we summarise the achievements of this thesis and discuss potential future directions.

6.1 Summary of Thesis Achievements

Explicit diffeomorphic cortical surface reconstruction by Neural ODEs. Inspired by recent advances in geometric deep learning for 3D surface reconstruction (Wang et al., 2018; Wen et al., 2019; Wickramasinghe et al., 2020), in Section 3.1, we developed a novel deep neural network model, called *PialNN*, to learn explicit and end-to-end cortical pial surface reconstruction from brain MR images. PialNN employs multiple deformation blocks to deform an input white matter surface into a pial surface, supervised by pseudo ground

truth (pGT) cortical surfaces generated by traditional surface-based brain MRI processing pipelines (Fischl, 2012; Glasser et al., 2013). In Section 3.2, we extended the architecture of PialNN and presented *CortexODE* framework for both inner and outer cortical surface reconstruction. CortexODE leverages Neural ODEs (Chen et al., 2018) to learn diffeomorphic surface deformations, with theoretical guarantees and numerical conditions to prevent self-intersections in cortical surfaces. We provided a fast re-implementation of topology correction algorithms (Bazin and Pham, 2005, 2007) to extract an initial white matter surface with spherical topology. Two CortexODE models were trained to explicitly deform the initial surface into white matter and pial surfaces respectively. The performance of CortexODE has been evaluated on various age groups including neonates (Edwards et al., 2022), young (Van Essen et al., 2013) and old adults (Jack Jr et al., 2008).

Neonatal cortical surface reconstruction by conditional temporal attention networks. Cortical surface reconstruction of fetal and neonatal brains is more challenging than adults as a consequence of rapid brain development during the perinatal period, leading to vast differences in scale and appearance of brains across fetal and neonatal subjects at different gestational and post-menstrual ages (Makropoulos et al., 2018). In Chapter 4, we presented customised DL-based frameworks to tackle fetal and neonatal cortical surface reconstruction. In Section 4.1, we introduced a *conditional temporal attention network* (*CoTAN*) for neonatal cortical surface reconstruction. CoTAN incorporates a channel-wise attention mechanism to learn a probabilistic attention map conditioned on input integration time step and the age of neonatal subjects. The predicted attention map measures the importance of multiple stationary velocity fields, which are combined to model a conditional time-varying velocity field for diffeomorphic surface deformations. Such an attention mechanism enables CoTAN to capture considerable variations in neonatal brains across different ages and to learn coarse-to-fine surface deformations automatically. CoTAN accelerates the reconstruction of cortical surfaces on both brain hemispheres to less than 0.5 seconds. The predicted surfaces achieve state-of-the-art geometric accuracy with less than 0.07% self-intersecting faces.

Fetal cortical surface reconstruction weakly supervised by segmentations. Despite the superior performance of our DL-based approaches on adult and neonatal cortical surface reconstruction, one limitation of these approaches is that they heavily rely on pseudo ground truth (pGT) cortical surfaces generated by traditional neuroimage processing pipelines (Fischl, 2012; Glasser et al., 2013; Makropoulos et al., 2018), which are usually time-consuming and may fail to produce acceptable pGT cortical surfaces, especially for low-resolution brain images such as fetal brain MRI data. In Section 4.2, we introduced *CoSeg*, a weakly supervised learning framework for fetal cortical surface reconstruction. CoSeg follows the architecture of CoTAN, supervised by pGT cortical ribbon segmentations without the need for pGT cortical surfaces. In order to alleviate partial volume effects in pGT cGM segmentations (Ballester et al., 2002), we proposed a novel weakly supervised loss to inflate the white matter surface towards the cGM segmentation boundary along normal directions, such that the reconstructed pial surfaces effectively captured deep cortical sulci. The experiments demonstrate that CoSeg predicts cortical surfaces with high anatomical and morphological accuracy on both low-resolution fetal (Price et al., 2019) and adult brain MRI datasets (Glasser et al., 2013).

Learning-based cortical surface reconstruction pipeline for the developing brain. To provide complete cortical surface-based analysis for neuroimaging studies, in Chapter 5, we developed a DL-based brain MRI processing pipeline for neonatal cortical surface reconstruction on the dHCP dataset (Edwards et al., 2022). Our DL-based dHCP pipeline is composed of DL-based brain extraction, cortical surface reconstruction and spherical projection, as well as GPU-accelerated cortical surface inflation and cortical feature estimation. A multiscale diffeomorphic deformation network is proposed to extract high-quality cortical surfaces with negligible number of self-intersections. The entire workflow of our DL-based dHCP pipeline only requires 24 seconds with GPU acceleration and 170 seconds on CPUs only. This is nearly 1000$\times$ faster than the original dHCP structural pipeline (Makropoulos et al., 2018) and 150$\times$ faster than previous DL-based cortical surface reconstruction pipelines (Henschel et al., 2020; Cruz et al., 2021; Wang et al., 2023).

Comparison and association between proposed DL-based approaches. In this thesis, we have introduced several geometric deep learning approaches for cortical surface reconstruction, which can be applied to different scenarios. PialNN (Section 3.1) is developed in particular for cortical pial surface reconstruction. CortexODE (Section 3.2) extends PialNN by learning diffeomorphic surface reconstruction of both white matter and pial surfaces via Neural ODEs. CoTAN (Section 4.1) is customised for neonatal cortical surface reconstruction and generalisable to adult subjects. CoTAN demonstrates superior geometric accuracy and computational efficiency than CortexODE, while CortexODE is more advantageous to process brain MRI at very high resolution. CoSeg (Section 4.2) is weakly supervised by cortical ribbon segmentations only without the dependence on pGT cortical surfaces. It is preferable when pGT cortical surfaces are not available, *e.g.*, for low-resolution fetal brain MRI on which traditional neuroimage processing pipelines fail. However, CortexODE and CoTAN would be better choices if accurate pGT cortical surfaces are provided. In addition to learning cortical surfaces end-to-end from brain MRI, the DL-based dHCP pipeline presented in Chapter 5 offers an entire workflow for cortical surface-based neuroimage processing and analysis. The pipeline aims to produce anatomically accurate cortical surfaces and corresponding attributes for neurodevelopmental study of newborn infants.

6.2 Future Work

In this section, we provide potential future directions to address existing limitations in this thesis and improve DL-based approaches for cortical surface reconstruction.

Probabilistic cortical surface reconstruction with uncertainty estimation. Due to complicated geometric structures and large variations of cortical surfaces across different subjects and ages, uncertainty is an important metric to measure the confidence and reliability of reconstructed cortical surfaces. Existing cortical surface reconstruction approaches primarily concentrate on deterministic approaches and seldom measure the uncertainty of produced cortical surfaces. Therefore, one of the future research directions is probabilistic cortical surface reconstruction with uncertainty estimation. The randomness could be introduced by Monte Carlo dropout modules (Gal and Ghahramani, 2016) or Bayesian neural networks (Kendall and Gal, 2017). Particularly, in the framework of diffeomorphic surface deformation, we could jointly learn the mean and variance of a stochastic velocity field via variational inference (Dalca et al., 2018, 2019), which models stochastic surface deformations for cortical surface reconstruction. The volumetric and surface-based uncertainties could be estimated by computing the variances of the velocity field and the deformation of vertices respectively (Kong et al., 2020). The uncertainties are expected to have a strong correlation to anatomical errors of the cortical surfaces, and thus potential corruptions could be indicated in areas with high uncertainty (Tóthová et al., 2020).

Semi-supervised cortical surface reconstruction. In Section 4.2, we have proposed a weakly supervised learning framework for cortical surface reconstruction from fetal brain MRI, of which the pGT cortical surfaces are not available when the traditional pipeline fails (Makropoulos et al., 2018). Such a framework learns cortical surfaces from pGT segmentations predicted by DL-based approaches. However, the anatomical accuracy of predicted cortical surfaces can be affected by imprecise pGT segmentations. The partial volume effects in pGT cGM segmentations (Ballester et al., 2002), which are not fully

addressed by current weakly supervised frameworks, further limit the accuracy of cortical pial surfaces. Alternatively, semi-supervised cortical surface reconstruction could be considered in future work. Motivated by existing automated cortical surface reconstruction pipelines (MacDonald et al., 2000; Shattuck and Leahy, 2002; Han et al., 2004; Fischl, 2012; Makropoulos et al., 2018), the objective is to first learn an initial white matter surface from pGT WM segmentations, which are the only supervision and relatively insusceptible to partial volume effects. Then, following conventional pipelines, an unsupervised loss function could be devised based on the intensity and contrast of brain MR images, encouraging the initial white matter surface to deform towards the WM/cGM and cGM/CSF boundaries for white matter and pial surface reconstruction respectively. The semi-supervised learning framework could fully utilise the brain MRI intensity, reduce the dependence on supervision, and predict cortical surfaces with high accuracy by learning to fit the tissue boundary directly in brain MRI.

Generalisable cortical surface reconstruction across all age groups. In this thesis, we have proposed several DL-based cortical surface reconstruction frameworks, of which the effectiveness has been verified on fetal, neonatal, and adult brain MRI datasets. However, our frameworks need to be re-trained to adapt to an unseen dataset, and a single deep neural network model does not generalise well to all age groups. Domain adaptation is an open problem in medical image analysis (Dou et al., 2019; Guan and Liu, 2021), which aims to address domain shifts and generalise to different modalities for medical images. For neuroimage analysis, domain shifts exist in brain MRI scans with different image acquisition and reconstruction protocols. Moreover, the infant and adult cortical surface reconstruction requires brain MRI with different modalities, *i.e.*, T2 and T1 weighted images, due to inverted brain MRI contrast. Inspired by state-of-the-art domain adaptation techniques for medical image segmentation (Dou et al., 2018; Billot et al., 2023a), one promising direction is generalisable cortical surface reconstruction across all age groups and acquisition protocols, which could be achieved, for example, by learning a shape prior or emphasising on relative image contrast instead of intensity values.

In this thesis, we introduced accurate, efficient, and versatile DL techniques for rapid cortical surface reconstruction from fetal, neonatal, and adult brain MRI scans, accelerating the runtime of traditional neuroimage processing pipelines from several hours to a few seconds. Our approaches enable fast and robust surface-based neuroimage processing and analysis, facilitating both neurodevelopmental and neurodegenerative studies. Looking ahead, we will develop reliable, adaptable, and generalisable DL-based methods, incorporating uncertainty quantification, unsupervised information, and domain adaptation, for cortical surface reconstruction across diverse populations, modalities, and imaging protocols. We aim to provide a universal surface-based brain MRI processing tool to the neuroimaging community for a better understanding of brain anatomy and connectome across the lifespan. Moreover, the DL-based approaches presented in this thesis could be further extended for surface reconstruction from various organs and tissue types, making a broad impact on medical image analysis and healthcare.

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. (2016). TensorFlow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Aël Chetelat, G. and Baron, J.-C. (2003). Early diagnosis of Alzheimer's disease: contribution of structural neuroimaging. *NeuroImage*, 18(2):525–541.

Alfaro-Almagro, F., Jenkinson, M., Bangerter, N. K., Andersson, J. L., Griffanti, L., Douaud, G., Sotiropoulos, S. N., Jbabdi, S., Hernandez-Fernandez, M., Vallee, E., et al. (2018). Image processing and quality control for the first 10,000 brain imaging datasets from UK Biobank. *NeuroImage*, 166:400–424.

Aljabar, P., Heckemann, R. A., Hammers, A., Hajnal, J. V., and Rueckert, D. (2009). Multi-atlas based segmentation of brain images: atlas selection and its effect on accuracy. *NeuroImage*, 46(3):726–738.

Allen, J. S., Damasio, H., and Grabowski, T. J. (2002). Normal neuroanatomical variation in the human brain: An MRI-volumetric study. *American Journal of Physical Anthropology*, 118(4):341–358.

Angenent, S., Haker, S., Tannenbaum, A., and Kikinis, R. (1999). On the Laplace-Beltrami operator and brain surface flattening. *IEEE Transactions on Medical Imaging*, 18(8):700–711.

Aoki, Y., Goforth, H., Srivatsan, R. A., and Lucey, S. (2019). PointNetLK: Robust & efficient point cloud registration using PointNet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7163–7172.

Arnold, V. I. (1992). *Ordinary differential equations*. Springer Science & Business Media.

Arsigny, V. (2004). *Processing data in Lie groups: An algebraic approach. Application to non-linear registration and diffusion tensor MRI*. PhD thesis, Citeseer.

Arsigny, V., Commowick, O., Pennec, X., and Ayache, N. (2006). A log-Euclidean framework for statistics on diffeomorphisms. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 924–931. Springer.

Arun, K. S., Huang, T. S., and Blostein, S. D. (1987). Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (5):698–700.

Ashburner, J. (2007). A fast diffeomorphic image registration algorithm. *NeuroImage*, 38(1):95–113.

Ashburner, J. (2009). Computational anatomy with the SPM software. *Magnetic Resonance Imaging*, 27(8):1163–1174.

Ashburner, J. and Friston, K. J. (2005). Unified segmentation. *NeuroImage*, 26(3):839–851.

Avants, B. B., Tustison, N., Song, G., et al. (2009). Advanced normalization tools (ANTS). *Insight J*, 2(365):1–35.

Avants, B. B., Tustison, N. J., Stauffer, M., Song, G., Wu, B., and Gee, J. C. (2014). The Insight ToolKit image registration framework. *Frontiers in Neuroinformatics*, 8:44.

Balakrishnan, G., Zhao, A., Sabuncu, M. R., Guttag, J., and Dalca, A. V. (2019). VoxelMorph: a learning framework for deformable medical image registration. *IEEE Transactions on Medical Imaging*, 38(8):1788–1800.

Ballester, M. A. G., Zisserman, A. P., and Brady, M. (2002). Estimation of the partial volume effect in MRI. *Medical Image Analysis*, 6(4):389–405.

Bazin, P.-L. and Pham, D. L. (2005). Topology correction using fast marching methods and its application to brain segmentation. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 484–491. Springer.

Bazin, P.-L. and Pham, D. L. (2007). Topology correction of segmented medical images using a fast marching algorithm. *Computer Methods and Programs in Biomedicine*, 88(2):182–190.

Bazin, P.-L. and Pham, D. L. (2008). Homeomorphic brain image segmentation with topological and statistical atlases. *Medical Image Analysis*, 12(5):616–625.

Beg, M. F., Miller, M. I., Trouvé, A., and Younes, L. (2005). Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International Journal of Computer Vision*, 61(2):139–157.

Bergen, T. and Wittenberg, T. (2014). Stitching and surface reconstruction from endoscopic image sequences: a review of applications and methods. *IEEE Journal of Biomedical and Health Informatics*, 20(1):304–321.

Besl, P. and McKay, N. D. (1992). A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(02):239–256.

Bethlehem, R. A., Seidlitz, J., White, S. R., Vogel, J. W., Anderson, K. M., Adamson, C., Adler, S., Alexopoulos, G. S., Anagnostou, E., Areces-Gonzalez, A., et al. (2022). Brain charts for the human lifespan. *Nature*, 604(7906):525–533.

Billot, B., Greve, D. N., Puonti, O., Thielscher, A., Van Leemput, K., Fischl, B., Dalca, A. V., Iglesias, J. E., et al. (2023a). SynthSeg: Segmentation of brain MRI scans of any contrast and resolution without retraining. *Medical Image Analysis*, 86:102789.

Billot, B., Magdamo, C., Cheng, Y., Arnold, S. E., Das, S., and Iglesias, J. E. (2023b). Robust machine learning segmentation for large-scale analysis of heterogeneous clinical brain MRI datasets. *Proceedings of the National Academy of Sciences*, 120(9):e2216399120.

Bongratz, F., Rickmann, A.-M., Pölsterl, S., and Wachinger, C. (2022). Vox2Cortex: Fast explicit reconstruction of cortical surfaces from 3D MRI scans with geometric deep neu-

ral networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20773–20783.

Bongratz, F., Rickmann, A.-M., and Wachinger, C. (2024). Neural deformation fields for template-based reconstruction of cortical surfaces from MRI. *Medical Image Analysis*, 93:103093.

Botsch, M. and Sorkine, O. (2007). On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):213–230.

Bozek, J., Makropoulos, A., Schuh, A., Fitzgibbon, S., Wright, R., Glasser, M. F., Coalson, T. S., O'Muircheartaigh, J., Hutter, J., Price, A. N., et al. (2018). Construction of a neonatal cortical surface atlas using multimodal surface matching in the developing human connectome project. *NeuroImage*, 179:11–29.

Bronstein, M. M., Bruna, J., Cohen, T., and Veličković, P. (2021). Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*.

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P. (2017). Geometric deep learning: going beyond Euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42.

Butcher, J. C. (1963). Coefficients for the study of runge-kutta integration processes. *Journal of the Australian Mathematical Society*, 3(2):185–201.

Butcher, J. C. (1987). *The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods*. Wiley-Interscience.

Butcher, J. C. (2016). *Numerical methods for ordinary differential equations*. John Wiley & Sons.

Butt, M. A. and Maragos, P. (1998). Optimum design of chamfer distance transforms. *IEEE Transactions on Image Processing*, 7(10):1477–1484.

Cabezas, M., Oliver, A., Lladó, X., Freixenet, J., and Cuadra, M. B. (2011). A review of atlas-based segmentation for magnetic resonance brain images. *Computer Methods and Programs in Biomedicine*, 104(3):e158–e177.

Cappart, Q., Chételat, D., Khalil, E. B., Lodi, A., Morris, C., and Veličković, P. (2023). Combinatorial optimization and reasoning with graph neural networks. *Journal of Machine Learning Research*, 24(130):1–61.

Caselles, V., Kimmel, R., and Sapiro, G. (1997). Geodesic active contours. *International Journal of Computer Vision*, 22:61–79.

Chan, T. F. and Vese, L. A. (2001). Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277.

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al. (2015). ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*.

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. (2018). Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, pages 6572–6583.

Chen, S. and Zhang, D. (2004). Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(4):1907–1916.

Chen, X., Zhao, J., Liu, S., Ahmad, S., and Yap, P.-T. (2023). SurfFlow: A flow-based approach for rapid and accurate cortical surface reconstruction from infant brain MRI. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 380–388. Springer.

Cheng, J., Dalca, A. V., Fischl, B., Zöllei, L., Initiative, A. D. N., et al. (2020). Cortical surface registration using unsupervised learning. *NeuroImage*, 221:117161.

Cho, K., van Merrienboer, B., Gulcehre, C., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing*.

Chui, H. and Rangarajan, A. (2000). A new algorithm for non-rigid point matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 44–51. IEEE.

Ciarlet, P. G. (2013). *Linear and nonlinear functional analysis with applications*. SIAM.

Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., Ranzuglia, G., et al. (2008). MeshLab: an open-source mesh processing tool. In *Eurographics Italian Chapter Conference*, volume 2008, pages 129–136. Salerno, Italy.

Cocosco, C. A., Zijdenbos, A. P., and Evans, A. C. (2003). A fully automatic and robust brain MRI tissue classification method. *Medical Image Analysis*, 7(4):513–527.

Coddington, E. A., Levinson, N., and Teichmann, T. (1956). *Theory of ordinary differential equations*. American Institute of Physics.

Cohen, T. S., Geiger, M., Köhler, J., and Welling, M. (2018). Spherical CNNs. In *International Conference on Learning Representations*.

Collins, D. L., Holmes, C. J., Peters, T. M., and Evans, A. C. (1995). Automatic 3-D model-based neuroanatomical segmentation. *Human Brain Mapping*, 3(3):190–208.

Collins, D. L., Neelin, P., Peters, T. M., and Evans, A. C. (1994). Automatic 3D intersubject registration of MR volumetric data in standardized Talairach space. *Journal of Computer Assisted Tomography*, 18(2):192–205.

Cordero-Grande, L., Hughes, E. J., Hutter, J., Price, A. N., and Hajnal, J. V. (2018). Three-dimensional motion corrected sensitivity encoding reconstruction for multi-shot multi-slice MRI: application to neonatal brain imaging. *Magnetic Resonance in Medicine*, 79(3):1365–1376.

Crane, K. (2018). Discrete differential geometry: An applied introduction. *Notices of the AMS, Communication*, 1153.

Cruz, R. S., Lebrat, L., Bourgeat, P., Fookes, C., Fripp, J., and Salvado, O. (2021). DeepCSR: A 3D deep learning approach for cortical surface reconstruction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 806–815.

Curless, B. and Levoy, M. (1996). A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 303–312.

Dahnke, R., Yotter, R. A., and Gaser, C. (2013). Cortical thickness and central surface estimation. *NeuroImage*, 65:336–348.

Dai, Y., Shi, F., Wang, L., Wu, G., and Shen, D. (2013). iBEAT: a toolbox for infant brain magnetic resonance image processing. *Neuroinformatics*, 11:211–225.

Dalca, A. V., Balakrishnan, G., Guttag, J., and Sabuncu, M. R. (2018). Unsupervised learning for fast probabilistic diffeomorphic registration. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 729–738. Springer.

Dalca, A. V., Balakrishnan, G., Guttag, J., and Sabuncu, M. R. (2019). Unsupervised learning of probabilistic diffeomorphic registration for images and surfaces. *Medical Image Analysis*, 57:226–236.

Dale, A. M., Fischl, B., and Sereno, M. I. (1999). Cortical surface-based analysis I: Segmentation and surface reconstruction. *NeuroImage*, 9(2):179–194.

Dale, A. M. and Sereno, M. I. (1993). Improved localizadon of cortical activity by combining EEG and MEG with MRI cortical surface reconstruction: a linear approach. *Journal of Cognitive Neuroscience*, 5(2):162–176.

Davatzikos, C. and Bryan, N. (1996). Using a deformable surface model to obtain a shape representation of the cortex. *IEEE Transactions on Medical Imaging*, 15(6):785–795.

Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in Neural Information Processing Systems*, 29.

Deisenroth, M. P., Faisal, A. A., and Ong, C. S. (2020). *Mathematics for machine learning*. Cambridge University Press.

del Fresno, M., Vénere, M., and Clausse, A. (2009). A combined region growing and deformable model method for extraction of closed surfaces in 3D CT and MRI scans. *Computerized Medical Imaging and Graphics*, 33(5):369–376.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee.

Desbrun, M., Meyer, M., Schröder, P., and Barr, A. H. (1999). Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pages 317–324.

Desikan, R. S., Ségonne, F., Fischl, B., Quinn, B. T., Dickerson, B. C., Blacker, D., Buckner, R. L., Dale, A. M., Maguire, R. P., Hyman, B. T., et al. (2006). An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. *NeuroImage*, 31(3):968–980.

Despotović, I., Goossens, B., and Philips, W. (2015). MRI segmentation of the human brain: challenges, methods, and applications. *Computational and Mathematical Methods in Medicine*, 2015(1):450341.

Destrieux, C., Fischl, B., Dale, A., and Halgren, E. (2010). Automatic parcellation of human cortical gyri and sulci using standard anatomical nomenclature. *NeuroImage*, 53(1):1–15.

Di Martino, A., Yan, C.-G., Li, Q., Denio, E., Castellanos, F. X., Alaerts, K., Anderson, J. S., Assaf, M., Bookheimer, S. Y., Dapretto, M., et al. (2014). The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. *Molecular Psychiatry*, 19(6):659–667.

Do Carmo, M. P. (1992). *Riemannian geometry*, volume 2. Springer.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., et al. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*.

Dou, Q., Coelho de Castro, D., Kamnitsas, K., and Glocker, B. (2019). Domain generalization via model-agnostic learning of semantic features. *Advances in Neural Information Processing Systems*, 32.

Dou, Q., Ouyang, C., Chen, C., Chen, H., and Heng, P.-A. (2018). Unsupervised cross-modality domain adaptation of ConvNets for biomedical image segmentations with adversarial loss. In *International Joint Conference on Artificial Intelligence*, page 691.

Duvenaud, D. K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., and Adams, R. P. (2015). Convolutional networks on graphs for learning molecular fingerprints. *Advances in Neural Information Processing Systems*, 28.

Edelsbrunner, H. and Harer, J. L. (2010). *Computational topology: an introduction*. American Mathematical Society.

Edwards, A. D., Redshaw, M. E., Kennea, N., Rivero-Arias, O., Gonzales-Cinca, N., Nongena, P., Ederies, M., Falconer, S., Chew, A., Omar, O., et al. (2018). Effect of MRI on preterm infants and their families: a randomised trial with nested diagnostic and economic evaluation. *Archives of Disease in Childhood-Fetal and Neonatal Edition*, 103(1):F15–F21.

Edwards, A. D., Rueckert, D., Smith, S. M., Seada, S. A., Alansary, A., Almalbis, J., Allsop, J., Andersson, J., Arichi, T., Arulkumaran, S., et al. (2022). The developing human connectome project neonatal data release. *Frontiers in Neuroscience*, 16.

Eickhoff, S. B., Yeo, B. T., and Genon, S. (2018). Imaging-based parcellations of the human brain. *Nature Reviews Neuroscience*, 19(11):672–686.

Elad, A., Keller, Y., and Kimmel, R. (2005). Texture mapping via spherical multi-dimensional scaling. In *International Conference on Scale-Space Theories in Computer Vision*, pages 443–455. Springer.

Fan, H., Su, H., and Guibas, L. J. (2017). A point set generation network for 3D object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 605–613.

Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J., and Yin, D. (2019). Graph neural networks for social recommendation. In *The World Wide Web Conference*, pages 417–426.

Fawaz, A., Williams, L. Z., Alansary, A., Bass, C., Gopinath, K., da Silva, M., Dahan, S., Adamson, C., Alexander, B., Thompson, D., et al. (2021). Benchmarking geometric deep learning for cortical segmentation and neurodevelopmental phenotype prediction. *bioRxiv*, pages 2021–12.

Fetit, A. E., Alansary, A., Cordero-Grande, L., Cupitt, J., Davidson, A. B., Edwards, A. D., Hajnal, J. V., Hughes, E., Kamnitsas, K., Kyriakopoulou, V., et al. (2020). A deep learning approach to segmentation of the developing cortex in fetal brain MRI with minimal manual labeling. In *Medical Imaging with Deep Learning*, pages 241–261. PMLR.

Fischl, B. (2012). FreeSurfer. *NeuroImage*, 62(2):774–781.

Fischl, B. and Dale, A. M. (2000). Measuring the thickness of the human cerebral cortex from magnetic resonance images. *Proceedings of the National Academy of Sciences*, 97(20):11050–11055.

Fischl, B., Liu, A., and Dale, A. M. (2001). Automated manifold surgery: constructing geometrically accurate and topologically correct models of the human cerebral cortex. *IEEE Transactions on Medical Imaging*, 20(1):70–80.

Fischl, B., Salat, D. H., Busa, E., Albert, M., Dieterich, M., Haselgrove, C., Van Der Kouwe, A., Killiany, R., Kennedy, D., Klaveness, S., et al. (2002). Whole brain segmentation: automated labeling of neuroanatomical structures in the human brain. *Neuron*, 33(3):341–355.

Fischl, B., Sereno, M. I., and Dale, A. M. (1999a). Cortical surface-based analysis II: Inflation, flattening, and a surface-based coordinate system. *NeuroImage*, 9(2):195–207.

Fischl, B., Sereno, M. I., Tootell, R. B., and Dale, A. M. (1999b). High-resolution intersubject averaging and a coordinate system for the cortical surface. *Human Brain Mapping*, 8(4):272–284.

Fischl, B., Van Der Kouwe, A., Destrieux, C., Halgren, E., Ségonne, F., Salat, D. H., Busa, E., Seidman, L. J., Goldstein, J., Kennedy, D., et al. (2004). Automatically parcellating the human cerebral cortex. *Cerebral Cortex*, 14(1):11–22.

Fonov, V., Evans, A. C., Botteron, K., Almli, C. R., McKinstry, R. C., Collins, D. L., Group, B. D. C., et al. (2011). Unbiased average age-appropriate atlases for pediatric studies. *NeuroImage*, 54(1):313–327.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. PMLR.

Garcia, K., Kroenke, C., and Bayly, P. (2018a). Mechanics of cortical folding: stress, growth and stability. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 373(1759):20170321.

Garcia, K. E., Robinson, E. C., Alexopoulos, D., Dierker, D. L., Glasser, M. F., Coalson, T. S., Ortinau, C. M., Rueckert, D., Taber, L. A., Van Essen, D. C., et al. (2018b). Dynamic patterns of cortical expansion during folding of the preterm human brain. *Proceedings of the National Academy of Sciences*, 115(12):3156–3161.

Garland, M. and Heckbert, P. S. (1997). Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, pages 209–216.

Gaudelet, T., Day, B., Jamasb, A. R., Soman, J., Regep, C., Liu, G., Hayter, J. B., Vickers, R., Roberts, C., Tang, J., et al. (2021). Utilizing graph machine learning within drug discovery and development. *Briefings in Bioinformatics*, 22(6):bbab159.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017). Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pages 1263–1272. PMLR.

Gkioxari, G., Malik, J., and Johnson, J. (2019). Mesh R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9785–9795.

Glasser, M. F., Coalson, T. S., Robinson, E. C., Hacker, C. D., Harwell, J., Yacoub, E., Ugurbil, K., Andersson, J., Beckmann, C. F., Jenkinson, M., et al. (2016). A multi-modal parcellation of human cerebral cortex. *Nature*, 536(7615):171–178.

Glasser, M. F., Sotiropoulos, S. N., Wilson, J. A., Coalson, T. S., Fischl, B., Andersson, J. L., Xu, J., Jbabdi, S., Webster, M., Polimeni, J. R., et al. (2013). The minimal preprocessing pipelines for the Human Connectome Project. *NeuroImage*, 80:105–124.

Glasser, M. F. and Van Essen, D. C. (2011). Mapping human cortical areas in vivo based on myelin content as revealed by T1-and T2-weighted MRI. *Journal of Neuroscience*, 31(32):11597–11616.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Gopinath, K., Desrosiers, C., and Lombaert, H. (2021). SegRecon: Learning joint brain surface reconstruction and segmentation from images. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 650–659. Springer.

Gopinath, K., Desrosiers, C., and Lombaert, H. (2023a). Learning joint surface reconstruction and segmentation, from brain images to cortical surface parcellation. *Medical Image Analysis*, 90:102974.

Gopinath, K., Greve, D. N., Das, S., Arnold, S., Magdamo, C., and Iglesias, J. E. (2023b). Cortical analysis of heterogeneous clinical brain MRI scans for large-scale neuroimaging studies. In *International Conference on Medical Image Computing and Computer Assisted Intervention*. Springer.

Grigorescu, I., Vanes, L., Uus, A., Batalle, D., Cordero-Grande, L., Nosarti, C., Edwards, A. D., Hajnal, J. V., Modat, M., and Deprez, M. (2021). Harmonized segmentation of neonatal brain MRI. *Frontiers in Neuroscience*, 15:662005.

Groueix, T., Fisher, M., Kim, V. G., Russell, B. C., and Aubry, M. (2018). A Papier-Mâché approach to learning 3D surface generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 216–224.

Guan, H. and Liu, M. (2021). Domain adaptation for medical image analysis: a survey. *IEEE Transactions on Biomedical Engineering*, 69(3):1173–1185.

Guo, M.-H., Cai, J.-X., Liu, Z.-N., Mu, T.-J., Martin, R. R., and Hu, S.-M. (2021). PCT: Point cloud transformer. *Computational Visual Media*, 7:187–199.

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., and Bennamoun, M. (2020). Deep learning for 3D point clouds: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4338–4364.

Gupta, K. and Chandraker, M. (2020). Neural Mesh Flow: 3D manifold mesh generation

via diffeomorphic flows. In *Advances in Neural Information Processing Systems*, pages 1747–1758.

Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, 30.

Han, X. and Fischl, B. (2007). Atlas renormalization for improved brain MR image segmentation across scanner platforms. *IEEE Transactions on Medical Imaging*, 26(4):479–486.

Han, X., Pham, D. L., Tosun, D., Rettmann, M. E., Xu, C., and Prince, J. L. (2004). CRUISE: cortical reconstruction using implicit surface evolution. *NeuroImage*, 23(3):997–1012.

Han, X., Xu, C., Braga-Neto, U., and Prince, J. L. (2002). Topology correction in brain cortex segmentation using a multiscale, graph-based algorithm. *IEEE Transactions on Medical Imaging*, 21(2):109–121.

Hanocka, R., Metzer, G., Giryes, R., and Cohen-Or, D. (2020). Point2Mesh: a self-prior for deformable meshes. *ACM Transactions on Graphics*, 39(4):126–1.

Hatcher, A. (2002). *Algebraic Topology*. Cambridge University Press.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

Heckemann, R. A., Hajnal, J. V., Aljabar, P., Rueckert, D., and Hammers, A. (2006). Automatic anatomical brain MRI segmentation combining label propagation and decision fusion. *NeuroImage*, 33(1):115–126.

Henschel, L., Conjeti, S., Estrada, S., Diers, K., Fischl, B., and Reuter, M. (2020). Fastsurfer-a fast and accurate deep learning based neuroimaging pipeline. *NeuroImage*, 219:117012.

Higham, N. J. (2005). The scaling and squaring method for the matrix exponential revisited. *SIAM Journal on Matrix Analysis and Applications*, 26(4):1179–1193.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Hong, Y., Ahmad, S., Wu, Y., Liu, S., and Yap, P.-T. (2021). Vox2Surf: Implicit surface reconstruction from volumetric data. In *International Workshop on Machine Learning in Medical Imaging*, pages 644–653. Springer.

Hoopes, A., Iglesias, J. E., Fischl, B., Greve, D., and Dalca, A. V. (2022). TopoFit: Rapid reconstruction of topologically-correct cortical surfaces. In *Medical Imaging with Deep Learning*.

Howell, B. R., Styner, M. A., Gao, W., Yap, P.-T., Wang, L., Baluyot, K., Yacoub, E., Chen, G., Potts, T., Salzwedel, A., et al. (2019). The UNC/UMN Baby Connectome Project (BCP): An overview of the study design and protocol development. *NeuroImage*, 185:891–905.

Hu, J., Shen, L., and Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7132–7141.

Hughes, E. J., Winchman, T., Padormo, F., Teixeira, R., Wurie, J., Sharma, M., Fox, M., Hutter, J., Cordero-Grande, L., Price, A. N., et al. (2017). A dedicated neonatal brain imaging system. *Magnetic Resonance in Medicine*, 78(2):794–804.

Hui, K.-H., Li, R., Hu, J., and Fu, C.-W. (2022). Neural Template: Topology-aware reconstruction and disentangled generation of 3D meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18572–18582.

Huo, Y., Plassard, A. J., Carass, A., Resnick, S. M., Pham, D. L., Prince, J. L., and Landman, B. A. (2016). Consistent cortical reconstruction and multi-atlas brain segmentation. *NeuroImage*, 138:197–210.

Isensee, F., Jaeger, P. F., Kohl, S. A., Petersen, J., and Maier-Hein, K. H. (2021). nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18(2):203–211.

Jack Jr, C. R., Bernstein, M. A., Fox, N. C., Thompson, P., Alexander, G., Harvey, D., Borowski, B., Britson, P. J., L. Whitwell, J., Ward, C., et al. (2008). The Alzheimer's disease neuroimaging initiative (ADNI): MRI methods. *Journal of Magnetic Resonance Imaging*, 27(4):685–691.

Jenkinson, M., Beckmann, C. F., Behrens, T. E., Woolrich, M. W., and Smith, S. M. (2012). FSL. *NeuroImage*, 62(2):782–790.

Jiang, C., Sud, A., Makadia, A., Huang, J., Nießner, M., Funkhouser, T., et al. (2020). Local implicit grid representations for 3D scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010.

Jordan, M. I. and Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260.

Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331.

Keckler, S. W., Dally, W. J., Khailany, B., Garland, M., and Glasco, D. (2011). GPUs and the future of parallel computing. *IEEE Micro*, 31(5):7–17.

Kendall, A. and Gal, Y. (2017). What uncertainties do we need in Bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, 30.

Khalil, E., Dai, H., Zhang, Y., Dilkina, B., and Song, L. (2017). Learning combinatorial optimization algorithms over graphs. *Advances in Neural Information Processing Systems*, 30.

Kim, H., Lepage, C., Evans, A. C., Barkovich, A. J., and Xu, D. (2015). NEOCIVET: Extraction of cortical surface and analysis of neonatal gyrification using a modified CIVET

pipeline. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 571–579. Springer.

Kim, J. S., Singh, V., Lee, J. K., Lerch, J., Ad-Dab'bagh, Y., MacDonald, D., Lee, J. M., Kim, S. I., and Evans, A. C. (2005). Automated 3-D extraction and evaluation of the inner and outer cortical surfaces using a Laplacian map and partial volume effect classification. *NeuroImage*, 27(1):210–221.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Klein, A. and Tourville, J. (2012). 101 labeled brain images and a consistent human cortical labeling protocol. *Frontiers in Neuroscience*, 6:171.

Kong, F. and Shadden, S. C. (2022). Learning whole heart mesh generation from patient images for computational simulations. *IEEE Transactions on Medical Imaging*, 42(2):533–545.

Kong, F., Wilson, N., and Shadden, S. (2021). A deep-learning approach for direct whole-heart mesh reconstruction. *Medical Image Analysis*, 74:102222.

Kong, L., Sun, J., and Zhang, C. (2020). SDE-Net: Equipping deep neural networks with uncertainty estimates. In *International Conference on Machine Learning*, pages 5405–5415. PMLR.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25.

Kuklisova-Murgasova, M., Quaghebeur, G., Rutherford, M. A., Hajnal, J. V., and Schnabel, J. A. (2012). Reconstruction of fetal brain MRI with intensity matching and complete outlier removal. *Medical Image Analysis*, 16(8):1550–1564.

Lam, S. K., Pitrou, A., and Seibert, S. (2015). Numba: A LLVM-based Python JIT compiler. In *2nd Workshop on the LLVM Compiler Infrastructure in HPC*, pages 1–6.

LaMontagne, P. J., Benzinger, T. L., Morris, J. C., Keefe, S., Hornbeck, R., Xiong, C., Grant, E., Hassenstab, J., Moulder, K., Vlassenko, A. G., et al. (2019). OASIS-3: longitudinal neuroimaging, clinical, and cognitive dataset for normal aging and Alzheimer disease. *medRxiv*, pages 2019–12.

Lebrat, L., Santa Cruz, R., de Gournay, F., Fu, D., Bourgeat, P., Fripp, J., Fookes, C., and Salvado, O. (2021). CorticalFlow: A diffeomorphic mesh transformer network for cortical surface reconstruction. *Advances in Neural Information Processing Systems*, 34:29491–29505.

LeCun, Y., Bengio, Y., et al. (1995). Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, 3361(10):1995.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551.

Lee, J. M. (2012). *Introduction to smooth manifolds*. Springer.

Lerch, J. P., Pruessner, J., Zijdenbos, A. P., Collins, D. L., Teipel, S. J., Hampel, H., and Evans, A. C. (2008). Automated cortical thickness measurements from MRI can accurately separate Alzheimer's patients from normal elderly controls. *Neurobiology of Aging*, 29(1):23–30.

Li, C., Huang, R., Ding, Z., Gatenby, J. C., Metaxas, D. N., and Gore, J. C. (2011). A level set method for image segmentation in the presence of intensity inhomogeneities with application to MRI. *IEEE Transactions on Image Processing*, 20(7):2007–2016.

Li, C., Kao, C.-Y., Gore, J. C., and Ding, Z. (2008). Minimization of region-scalable fitting energy for image segmentation. *IEEE Transactions on Image Processing*, 17(10):1940–1949.

Liang, L., Wei, M., Szymczak, A., Petrella, A., Xie, H., Qin, J., Wang, J., and Wang, F. L. (2018). Nonrigid iterative closest points for registration of 3D biomedical surfaces. *Optics and Lasers in Engineering*, 100:141–154.

Liang, Y., Song, W., Yang, J., Qiu, L., Wang, K., and He, L. (2020). X2Teeth: 3D teeth reconstruction from a single panoramic radiograph. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 400–409. Springer.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., Van Der Laak, J. A., Van Ginneken, B., and Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88.

Liu, T., Nie, J., Tarokh, A., Guo, L., and Wong, S. T. (2008). Reconstruction of central cortical surface from brain MRI images: method and application. *NeuroImage*, 40(3):991–1002.

Lorensen, W. E. and Cline, H. E. (1998). Marching Cubes: A high resolution 3D surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353.

Lötjönen, J. M., Wolz, R., Koikkalainen, J. R., Thurfjell, L., Waldemar, G., Soininen, H., Rueckert, D., Initiative, A. D. N., et al. (2010). Fast and robust multi-atlas segmentation of brain magnetic resonance images. *NeuroImage*, 49(3):2352–2365.

Lu, W., Wan, G., Zhou, Y., Fu, X., Yuan, P., and Song, S. (2019). DeepVCP: An end-to-end deep neural network for point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–21.

Lyttelton, O., Boucher, M., Robbins, S., and Evans, A. (2007). An unbiased iterative group registration template for cortical surface analysis. *NeuroImage*, 34(4):1535–1544.

Ma, Q., Li, L., Kyriakopoulou, V., Hajnal, J. V., Robinson, E. C., Kainz, B., and Rueckert, D. (2023). Conditional temporal attention networks for neonatal cortical surface reconstruction. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 312–322. Springer.

Ma, Q., Li, L., Robinson, E. C., Kainz, B., and Rueckert, D. (2024). Weakly supervised learning of cortical surface reconstruction from segmentations. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 766–777. Springer.

Ma, Q., Li, L., Robinson, E. C., Kainz, B., Rueckert, D., and Alansary, A. (2022). CortexODE: Learning cortical surface reconstruction by neural ODEs. *IEEE Transactions on Medical Imaging*, 42(2):430–443.

Ma, Q., Liang, K., Li, L., Masui, S., Guo, Y., Nosarti, C., Robinson, E. C., Kainz, B., and Rueckert, D. (2025). The Developing Human Connectome Project: A fast deep learning-based pipeline for neonatal cortical surface reconstruction. *Medical Image Analysis*, 100:103394.

Ma, Q., Robinson, E. C., Kainz, B., Rueckert, D., and Alansary, A. (2021). PialNN: A fast deep learning framework for cortical pial surface reconstruction. In *International Workshop on Machine Learning in Clinical Neuroimaging*, pages 73–81. Springer.

MacDonald, D., Kabani, N., Avis, D., and Evans, A. C. (2000). Automated 3-D extraction of inner and outer surfaces of cerebral cortex from MRI. *NeuroImage*, 12(3):340–356.

Maclaren, J., Han, Z., Vos, S. B., Fischbein, N., and Bammer, R. (2014). Reliability of brain volume measurements: a test-retest dataset. *Scientific Data*, 1(1):1–9.

Madre, M., Canales-Rodríguez, E. J., Fuentes-Claramonte, P., Alonso-Lana, S., Salgado-Pineda, P., Guerrero-Pedraza, A., Moro, N., Bosque, C., Gomar, J. J., Ortíz-Gil, J., et al. (2020). Structural abnormality in schizophrenia versus bipolar disorder: a whole brain

cortical thickness, surface area, volume and gyrification analyses. *NeuroImage: Clinical*, 25:102131.

Mahmoud, N., Collins, T., Hostettler, A., Soler, L., Doignon, C., and Montiel, J. M. M. (2018). Live tracking and dense reconstruction for handheld monocular endoscopy. *IEEE Transactions on Medical Imaging*, 38(1):79–89.

Maillot, J., Yahia, H., and Verroust, A. (1993). Interactive texture mapping. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, pages 27–34.

Makropoulos, A., Aljabar, P., Wright, R., Hüning, B., Merchant, N., Arichi, T., Tusor, N., Hajnal, J. V., Edwards, A. D., Counsell, S. J., et al. (2016). Regional growth and atlasing of the developing human brain. *NeuroImage*, 125:456–478.

Makropoulos, A., Gousias, I. S., Ledig, C., Aljabar, P., Serag, A., Hajnal, J. V., Edwards, A. D., Counsell, S. J., and Rueckert, D. (2014). Automatic whole brain MRI segmentation of the developing neonatal brain. *IEEE Transactions on Medical Imaging*, 33(9):1818–1831.

Makropoulos, A., Robinson, E. C., Schuh, A., Wright, R., Fitzgibbon, S., Bozek, J., Counsell, S. J., Steinweg, J., Vecchiato, K., Passerat-Palmbach, J., et al. (2018). The developing human connectome project: A minimal processing pipeline for neonatal cortical surface reconstruction. *NeuroImage*, 173:88–112.

Malladi, R., Sethian, J. A., and Vemuri, B. C. (1995). Shape modeling with front propagation: A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175.

Marcus, D. S., Fotenos, A. F., Csernansky, J. G., Morris, J. C., and Buckner, R. L. (2010). Open access series of imaging studies: longitudinal MRI data in nondemented and demented older adults. *Journal of Cognitive Neuroscience*, 22(12):2677–2684.

Marcus, D. S., Wang, T. H., Parker, J., Csernansky, J. G., Morris, J. C., and Buckner, R. L. (2007). Open Access Series of Imaging Studies (OASIS): cross-sectional MRI data in young, middle aged, nondemented, and demented older adults. *Journal of Cognitive Neuroscience*, 19(9):1498–1507.

McEvoy, L. K., Fennema-Notestine, C., Roddey, J. C., Hagler Jr, D. J., Holland, D., Karow, D. S., Pung, C. J., Brewer, J. B., and Dale, A. M. (2009). Alzheimer disease: quantitative structural neuroimaging for detection and prediction of clinical and structural changes in mild cognitive impairment. *Radiology*, 251(1):195–205.

McInerney, T. and Terzopoulos, D. (1996). Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 1(2):91–108.

Meng, Q., Bai, W., O'Regan, D. P., and Rueckert, D. (2023). DeepMesh: Mesh-based cardiac motion tracking using deep learning. *IEEE Transactions on Medical Imaging*.

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. (2019). Occupancy networks: Learning 3D reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. (2021). NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106.

Mittal, P., Cheng, Y.-C., Singh, M., and Tulsiani, S. (2022). AutoSDF: Shape priors for 3D completion, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 306–315.

Modat, M., Cash, D. M., Daga, P., Winston, G. P., Duncan, J. S., and Ourselin, S. (2014). Global image registration using a symmetric block-matching approach. *Journal of Medical Imaging*, 1(2):024003–024003.

Moeskops, P., Viergever, M. A., Mendrik, A. M., De Vries, L. S., Benders, M. J., and Išgum, I. (2016). Automatic segmentation of MR brain images with a convolutional neural network. *IEEE Transactions on Medical Imaging*, 35(5):1252–1261.

Möller, T. (1997). A fast triangle-triangle intersection test. *Journal of Graphics Tools*, 2(2):25–30.

Nazari, M., Oroojlooy, A., Snyder, L., and Takác, M. (2018). Reinforcement learning for solving the vehicle routing problem. *Advances in Neural Information Processing Systems*, 31.

Nealen, A., Igarashi, T., Sorkine, O., and Alexa, M. (2006). Laplacian mesh optimization. In *Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia*, pages 381–389.

Nie, J., Li, G., Wang, L., Gilmore, J. H., Lin, W., and Shen, D. (2012). A computational growth model for measuring dynamic cortical development in the first year of life. *Cerebral Cortex*, 22(10):2272–2284.

Niemeyer, M., Mescheder, L., Oechsle, M., and Geiger, A. (2020). Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3504–3515.

Orasanu, E., Melbourne, A., Cardoso, M. J., Modat, M., Taylor, A. M., Thayyil, S., and Ourselin, S. (2014). Brain volume estimation from post-mortem newborn and fetal MRI. *NeuroImage: Clinical*, 6:438–444.

Owens, J. D., Houston, M., Luebke, D., Green, S., Stone, J. E., and Phillips, J. C. (2008). GPU computing. *Proceedings of the IEEE*, 96(5):879–899.

Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S. (2019). DeepSDF:

Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174.

Park, J.-Y., McInerney, T., Terzopoulos, D., and Kim, M.-H. (2001). A non-self-intersecting adaptive deformable surface for complex boundary extraction from volumetric images. *Computers & Graphics*, 25(3):421–440.

Park, K., Sinha, U., Barron, J. T., Bouaziz, S., Goldman, D. B., Seitz, S. M., and Martin-Brualla, R. (2021). Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32.

Patel, Y., Parker, N., Shin, J., Howard, D., French, L., Thomopoulos, S. I., Pozzi, E., Abe, Y., Abé, C., Anticevic, A., et al. (2021). Virtual histology of cortical thickness and shared neurobiology in 6 psychiatric disorders. *JAMA Psychiatry*, 78(1):47–63.

Peng, S., Jiang, C., Liao, Y., Niemeyer, M., Pollefeys, M., and Geiger, A. (2021). Shape as points: A differentiable Poisson solver. *Advances in Neural Information Processing Systems*, 34:13032–13044.

Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., and Geiger, A. (2020). Convolutional occupancy networks. In *Proceedings of the European Conference on Computer Vision*, pages 523–540. Springer.

Perko, L. (2013). *Differential equations and dynamical systems*, volume 7. Springer Science & Business Media.

Pham, D., Prince, J. L., Xu, C., and Dagher, A. P. (1997). An automated technique for

statistical characterization of brain tissues in magnetic resonance imaging. *International Journal of Pattern Recognition and Artificial Intelligence*, 11(08):1189–1211.

Pham, D. L. and Prince, J. L. (1999a). An adaptive fuzzy C-means algorithm for image segmentation in the presence of intensity inhomogeneities. *Pattern Recognition Letters*, 20(1):57–68.

Pham, D. L. and Prince, J. L. (1999b). Adaptive fuzzy segmentation of magnetic resonance images. *IEEE Transactions on Medical Imaging*, 18(9):737–752.

Pham, D. L., Xu, C., and Prince, J. L. (2000). Current methods in medical image segmentation. *Annual Review of Biomedical Engineering*, 2(1):315–337.

Pohl, K. M., Fisher, J., Grimson, W. E. L., Kikinis, R., and Wells, W. M. (2006). A Bayesian model for joint segmentation and registration. *NeuroImage*, 31(1):228–239.

Prastawa, M., Gilmore, J. H., Lin, W., and Gerig, G. (2005). Automatic segmentation of MR images of the developing newborn brain. *Medical Image Analysis*, 9(5):457–466.

Price, A. N., Cordero-Grande, L., et al. (2019). The developing human connectome project (dHCP): fetal acquisition protocol. In *Proceedings of the annual meeting of the International Society of Magnetic Resonance in Medicine*, volume 244.

Pumarola, A., Corona, E., Pons-Moll, G., and Moreno-Noguer, F. (2021). D-NeRF: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017a). PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017b). PointNet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, 30.

Qian, G., Li, Y., Peng, H., Mai, J., Hammoud, H., Elhoseiny, M., and Ghanem, B. (2022). PointNext: Revisiting PointNet++ with improved training and scaling strategies. *Advances in Neural Information Processing Systems*, 35:23192–23204.

Querbes, O., Aubry, F., Pariente, J., Lotterie, J.-A., Démonet, J.-F., Duret, V., Puel, M., Berry, I., Fort, J.-C., Celsis, P., et al. (2009). Early diagnosis of Alzheimer's disease using cortical thickness: impact of cognitive reserve. *Brain*, 132(8):2036–2047.

Ravi, N., Reizenstein, J., Novotny, D., Gordon, T., Lo, W.-Y., Johnson, J., and Gkioxari, G. (2020). Accelerating 3D deep learning with PyTorch3D. *arXiv preprint arXiv:2007.08501*.

Remelli, E., Lukoianov, A., Richter, S., Guillard, B., Bagautdinov, T., Baque, P., and Fua, P. (2020). MeshSDF: Differentiable iso-surface extraction. *Advances in Neural Information Processing Systems*, 33:22468–22478.

Rimol, L. M., Nesvåg, R., Hagler Jr, D. J., Bergmann, Ø., Fennema-Notestine, C., Hartberg, C. B., Haukvik, U. K., Lange, E., Pung, C. J., Server, A., et al. (2012). Cortical volume, surface area, and thickness in schizophrenia and bipolar disorder. *Biological Psychiatry*, 71(6):552–560.

Robinson, E. C., Garcia, K., Glasser, M. F., Chen, Z., Coalson, T. S., Makropoulos, A., Bozek, J., Wright, R., Schuh, A., Webster, M., et al. (2018). Multimodal surface matching with higher-order smoothness constraints. *NeuroImage*, 167:453–465.

Robinson, E. C., Jbabdi, S., Glasser, M. F., Andersson, J., Burgess, G. C., Harms, M. P., Smith, S. M., Van Essen, D. C., and Jenkinson, M. (2014). MSM: a new flexible framework for multimodal surface matching. *NeuroImage*, 100:414–426.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 234–241. Springer.

Roy, A. G., Conjeti, S., Navab, N., Wachinger, C., Initiative, A. D. N., et al. (2019). QuickNAT: A fully convolutional network for quick and accurate segmentation of neuroanatomy. *NeuroImage*, 186:713–727.

Roy, A. G., Navab, N., and Wachinger, C. (2018). Concurrent spatial and channel 'squeeze & excitation' in fully convolutional networks. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 421–429. Springer.

Ruelle, D. and Sullivan, D. (1975). Currents, flows and diffeomorphisms. *Topology*, 14(4):319–327.

Sahoo, P. K., Soltani, S., and Wong, A. K. (1988). A survey of thresholding techniques. *Computer Vision, Graphics, and Image Processing*, 41(2):233–260.

Santa Cruz, R., Lebrat, L., Fu, D., Bourgeat, P., Fripp, J., Fookes, C., and Salvado, O. (2022). CorticalFlow++: Boosting cortical surface reconstruction accuracy, regularity, and interoperability. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 496–505. Springer.

Schuh, A. (2016). Medical image registration toolkit (MIRTK).

Schuh, A., Makropoulos, A., Robinson, E. C., Cordero-Grande, L., Hughes, E., Hutter, J., Price, A. N., Murgasova, M., Teixeira, R. P. A., Tusor, N., et al. (2018). Unbiased construction of a temporally consistent morphological atlas of neonatal brain development. *bioRxiv*, page 251512.

Schuh, A., Makropoulos, A., Wright, R., Robinson, E. C., Tusor, N., Steinweg, J., Hughes, E., Grande, L. C., Price, A., Hutter, J., et al. (2017). A deformable model for the reconstruction of the neonatal cortex. In *IEEE 14th International Symposium on Biomedical Imaging*, pages 800–803. IEEE.

Ségonne, F., Pacheco, J., and Fischl, B. (2007). Geometrically accurate topology-correction

of cortical surfaces using nonseparating loops. *IEEE Transactions on Medical Imaging*, 26(4):518–529.

Sereno, M. I., Dale, A., Reppas, J., Kwong, K., Belliveau, J., Brady, T., Rosen, B., and Tootell, R. (1995). Borders of multiple visual areas in humans revealed by functional magnetic resonance imaging. *Science*, 268(5212):889–893.

Shattuck, D. W. and Leahy, R. M. (2001). Automated graph-based analysis and correction of cortical volume topology. *IEEE Transactions on Medical Imaging*, 20(11):1167–1177.

Shattuck, D. W. and Leahy, R. M. (2002). BrainSuite: an automated cortical surface identification tool. *Medical Image Analysis*, 6(2):129–142.

Shattuck, D. W., Mirza, M., Adisetiyo, V., Hojatkashani, C., Salamon, G., Narr, K. L., Poldrack, R. A., Bilder, R. M., and Toga, A. W. (2008). Construction of a 3D probabilistic atlas of human cortical structures. *NeuroImage*, 39(3):1064–1080.

Shattuck, D. W., Sandor-Leahy, S. R., Schaper, K. A., Rottenberg, D. A., and Leahy, R. M. (2001). Magnetic resonance image tissue classification using a partial volume model. *NeuroImage*, 13(5):856–876.

Shen, D., Wu, G., and Suk, H.-I. (2017). Deep learning in medical image analysis. *Annual Review of Biomedical Engineering*, 19(1):221–248.

Sitzmann, V., Martel, J., Bergman, A., Lindell, D., and Wetzstein, G. (2020). Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473.

Sled, J. G., Zijdenbos, A. P., and Evans, A. C. (1998). A nonparametric method for automatic correction of intensity nonuniformity in MRI data. *IEEE Transactions on Medical Imaging*, 17(1):87–97.

Smith, S. M. (2002). Fast robust automated brain extraction. *Human Brain Mapping*, 17(3):143–155.

Somerville, L. H., Bookheimer, S. Y., Buckner, R. L., Burgess, G. C., Curtiss, S. W., Dapretto, M., Elam, J. S., Gaffrey, M. S., Harms, M. P., Hodge, C., et al. (2018). The Lifespan Human Connectome Project in Development: A large-scale study of brain connectivity development in 5–21 year olds. *NeuroImage*, 183:456–468.

Sorkine, O., Cohen-Or, D., Lipman, Y., Alexa, M., Rössl, C., and Seidel, H.-P. (2004). Laplacian surface editing. In *Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pages 175–184.

Sotiras, A., Davatzikos, C., and Paragios, N. (2013). Deformable medical image registration: A survey. *IEEE Transactions on Medical Imaging*, 32(7):1153–1190.

Sudlow, C., Gallacher, J., Allen, N., Beral, V., Burton, P., Danesh, J., Downey, P., Elliott, P., Green, J., Landray, M., et al. (2015). UK Biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLoS Medicine*, 12(3):e1001779.

Süli, E. and Mayers, D. F. (2003). *An introduction to numerical analysis*. Cambridge University Press.

Suliman, M. A., Williams, L. Z., Fawaz, A., and Robinson, E. C. (2022a). A deep-discrete learning framework for spherical surface registration. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 119–129. Springer.

Suliman, M. A., Williams, L. Z. J., Fawaz, A., and Robinson, E. C. (2022b). GeoMorph: Geometric deep learning for cortical surface registration. In *Geometric Deep Learning in Medical Image Analysis*.

Sun, M., Zhao, S., Gilvary, C., Elemento, O., Zhou, J., and Wang, F. (2020). Graph convolutional networks for computational drug development and discovery. *Briefings in Bioinformatics*, 21(3):919–935.

Sun, S., Han, K., Kong, D., Tang, H., Yan, X., and Xie, X. (2022). Topology-preserving shape reconstruction and registration via neural diffeomorphic flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20845–20855.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9.

Talairach, J. (1988). Co-planar stereotaxic atlas of the human brain.

Taubin, G. (1995a). Curve and surface smoothing without shrinkage. In *Proceedings of IEEE International Conference on Computer Vision*, pages 852–857. IEEE.

Taubin, G. (1995b). A signal processing approach to fair surface design. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, pages 351–358.

Teo, P. C., Sapiro, G., and Wandell, B. A. (1997). Creating connected representations of cortical gray matter for functional MRI visualization. *IEEE Transactions on Medical Imaging*, 16(6):852–863.

Terzopoulos, D. and Fleischer, K. (1988). Deformable models. *The Visual Computer*, 4(6):306–331.

Tohka, J., Zijdenbos, A., and Evans, A. (2004). Fast and robust parameter estimation for statistical partial volume models in brain MRI. *NeuroImage*, 23(1):84–97.

Tootell, R. B., Mendola, J. D., Hadjikhani, N. K., Ledden, P. J., Liu, A. K., Reppas, J. B., Sereno, M. I., and Dale, A. M. (1997). Functional analysis of V3A and related areas in human visual cortex. *Journal of Neuroscience*, 17(18):7060–7078.

Tosun, D. and Prince, J. L. (2001). Hemispherical map for the human brain cortex. In *Medical Imaging: Image Processing*, volume 4322, pages 290–300. SPIE.

Tóthová, K., Parisot, S., Lee, M., Puyol-Antón, E., King, A., Pollefeys, M., and Konukoglu, E. (2020). Probabilistic 3D surface reconstruction from sparse MRI information. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 813–823. Springer.

Tustison, N. J., Avants, B. B., Cook, P. A., Zheng, Y., Egan, A., Yushkevich, P. A., and Gee, J. C. (2010). N4ITK: improved N3 bias correction. *IEEE Transactions on Medical Imaging*, 29(6):1310–1320.

Uus, A. U., Kyriakopoulou, V., et al. (2023a). BOUNTI: Brain volumetry and automated parcellation for 3D fetal MRI. *Elife*.

Uus, A. U., Kyriakopoulou, V., et al. (2023b). Multi-channel spatio-temporal MRI atlas of the normal fetal brain development from the developing human connectome project. *G-Node*.

Vahdat, A., Williams, F., Gojcic, Z., Litany, O., Fidler, S., Kreis, K., et al. (2022). LION: Latent point diffusion models for 3D shape generation. *Advances in Neural Information Processing Systems*, 35:10021–10039.

Van Essen, D. C., Glasser, M. F., Dierker, D. L., Harwell, J., and Coalson, T. (2012). Parcellations and hemispheric asymmetries of human cerebral cortex analyzed on surface-based atlases. *Cerebral Cortex*, 22(10):2241–2262.

Van Essen, D. C., Smith, S. M., Barch, D. M., Behrens, T. E., Yacoub, E., Ugurbil, K., Consortium, W.-M. H., et al. (2013). The WU-Minn human connectome project: an overview. *NeuroImage*, 80:62–79.

Van Leemput, K., Maes, F., Vandermeulen, D., and Suetens, P. (2003). A unifying framework for partial volume segmentation of brain MR images. *IEEE Transactions on Medical Imaging*, 22(1):105–119.

Varadarajan, V. S. (2013). *Lie groups, Lie algebras, and their representations*, volume 102. Springer Science & Business Media.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph attention networks. In *International Conference on Learning Representations*.

Vercauteren, T., Pennec, X., Perchant, A., and Ayache, N. (2009). Diffeomorphic demons: Efficient non-parametric image registration. *NeuroImage*, 45(1):S61–S72.

Vese, L. A. and Chan, T. F. (2002). A multiphase level set framework for image segmentation using the Mumford and Shah model. *International Journal of Computer Vision*, 50:271–293.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272.

Walter, W. (2013). *Ordinary differential equations*, volume 182. Springer Science & Business Media.

Wang, L., Wu, Z., Chen, L., Sun, Y., Lin, W., and Li, G. (2023). iBEAT v2.0: a multisite-applicable, deep learning-based pipeline for infant cerebral cortical surface reconstruction. *Nature Protocols*, 18(5):1488–1509.

Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., and Jiang, Y.-G. (2018). Pixel2Mesh: Generating 3D mesh models from single RGB images. In *Proceedings of the European Conference on Computer Vision*, pages 52–67.

Wang, X., Ang Jr, M. H., and Lee, G. H. (2020). Cascaded refinement network for point

cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 790–799.

Wang, Y. and Solomon, J. M. (2019). Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3523–3532.

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., and Solomon, J. M. (2019). Dynamic graph CNN for learning on point clouds. *ACM Transactions on Graphics*, 38(5):1–12.

Wang, Z. M., Soh, Y. C., Song, Q., and Sim, K. (2009). Adaptive spatial information-theoretic clustering for image segmentation. *Pattern Recognition*, 42(9):2029–2044.

Warfield, S. K., Kaus, M., Jolesz, F. A., and Kikinis, R. (2000). Adaptive, template moderated, spatially varying statistical classification. *Medical Image Analysis*, 4(1):43–55.

Wells, W. M., Grimson, W. E. L., Kikinis, R., and Jolesz, F. A. (1996). Adaptive segmentation of MRI data. *IEEE Transactions on Medical Imaging*, 15(4):429–442.

Wen, C., Zhang, Y., Li, Z., and Fu, Y. (2019). Pixel2Mesh++: Multi-view 3D mesh generation via deformation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1042–1051.

Wickramasinghe, U., Remelli, E., Knott, G., and Fua, P. (2020). Voxel2Mesh: 3D mesh model generation from volumetric data. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 299–308. Springer.

Williams, L. Z., Fawaz, A., Glasser, M. F., Edwards, A. D., and Robinson, E. C. (2021). Geometric deep learning of the human connectome project multimodal cortical parcellation. In *Machine Learning in Clinical Neuroimaging*, pages 103–112. Springer.

Wright, R., Kyriakopoulou, V., Ledig, C., Rutherford, M. A., Hajnal, J. V., Rueckert, D.,

and Aljabar, P. (2014). Automatic quantification of normal cortical folding patterns from fetal brain MRI. *NeuroImage*, 91:21–32.

Wu, C., Bradley, D., Garrido, P., Zollhöfer, M., Theobalt, C., Gross, M. H., and Beeler, T. (2016). Model-based teeth reconstruction. *ACM Transactions on Graphics*, 35(6):220–1.

Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., and Weinberger, K. (2019a). Simplifying graph convolutional networks. In *International Conference on Machine Learning*, pages 6861–6871. PMLR.

Wu, S., Sun, F., Zhang, W., Xie, X., and Cui, B. (2022). Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, 55(5):1–37.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24.

Wu, Z., Zhao, F., Xia, J., Wang, L., Lin, W., Gilmore, J. H., Li, G., and Shen, D. (2019b). Intrinsic patch-based cortical anatomical parcellation using graph convolutional neural network on surface manifold. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 492–500. Springer.

Xu, C., Pham, D. L., Rettmann, M. E., Yu, D. N., and Prince, J. L. (1999). Reconstruction of the human cerebral cortex from magnetic resonance images. *IEEE Transactions on Medical Imaging*, 18(6):467–480.

Xu, C. and Prince, J. L. (1998). Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. (2018). How powerful are graph neural networks? *International Conference on Learning Representations*.

Xue, H., Srinivasan, L., Jiang, S., Rutherford, M., Edwards, A. D., Rueckert, D., and Hajnal, J. V. (2007). Automatic segmentation and reconstruction of the cortex from neonatal MRI. *NeuroImage*, 38(3):461–477.

Yeo, B. T., Sabuncu, M., Vercauteren, T., Ayache, N., Fischl, B., and Golland, P. (2008). Spherical demons: Fast surface registration. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 745–753. Springer.

Yeo, B. T., Sabuncu, M. R., Vercauteren, T., Ayache, N., Fischl, B., and Golland, P. (2009). Spherical demons: fast diffeomorphic landmark-free surface registration. *IEEE Transactions on Medical Imaging*, 29(3):650–668.

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., and Leskovec, J. (2018). Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 974–983.

Yotter, R. A., Dahnke, R., Thompson, P. M., and Gaser, C. (2011). Topological correction of brain surface meshes using spherical harmonics. *Human Brain Mapping*, 32(7):1109–1124.

Younes, L. (2010). *Shapes and diffeomorphisms*, volume 171. Springer.

Yu, A., Ye, V., Tancik, M., and Kanazawa, A. (2021a). pixelNeRF: Neural radiance filds from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587.

Yu, X., Rao, Y., Wang, Z., Liu, Z., Lu, J., and Zhou, J. (2021b). PoinTr: Diverse point cloud completion with geometry-aware transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12498–12507.

Yu, X., Tang, L., Rao, Y., Huang, T., Zhou, J., and Lu, J. (2022). Point-BERT: Pre-training 3D

point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19313–19322.

Yuan, W., Khot, T., Held, D., Mertz, C., and Hebert, M. (2018). PCN: Point completion network. In *International Conference on 3D Vision*, pages 728–737. IEEE.

Yushkevich, P. A., Piven, J., Hazlett, H. C., Smith, R. G., Ho, S., Gee, J. C., and Gerig, G. (2006). User-guided 3D active contour segmentation of anatomical structures: significantly improved efficiency and reliability. *NeuroImage*, 31(3):1116–1128.

Zha, R., Cheng, X., Li, H., Harandi, M., and Ge, Z. (2023). EndoSurf: Neural surface reconstruction of deformable tissues with stereo endoscope videos. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 13–23. Springer.

Zhang, J., Yao, Y., and Deng, B. (2021). Fast and robust iterative closest point. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3450–3466.

Zhang, Z. (1994). Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152.

Zhao, F., Wu, Z., Wang, F., Lin, W., Xia, S., Shen, D., Wang, L., and Li, G. (2021a). S3Reg: superfast spherical surface registration based on deep learning. *IEEE Transactions on Medical Imaging*, 40(8):1964–1976.

Zhao, F., Wu, Z., Wang, L., Lin, W., Gilmore, J. H., Xia, S., Shen, D., and Li, G. (2021b). Spherical deformable U-Net: Application to cortical surface parcellation and development prediction. *IEEE Transactions on Medical Imaging*, 40(4):1217–1228.

Zhao, F., Wu, Z., Wang, L., Lin, W., and Li, G. (2022). Fast spherical mapping of cortical surface meshes using deep unsupervised learning. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 163–173. Springer.

Zhao, F., Xia, S., Wu, Z., Duan, D., Wang, L., Lin, W., Gilmore, J. H., Shen, D., and Li, G.

(2019a). Spherical U-Net on cortical surfaces: methods and applications. In *International Conference on Information Processing in Medical Imaging*, pages 855–866. Springer.

Zhao, F., Xia, S., Wu, Z., Wang, L., Chen, Z., Lin, W., Gilmore, J. H., Shen, D., and Li, G. (2019b). Spherical U-Net for infant cortical surface parcellation. In *IEEE 16th International Symposium on Biomedical Imaging*, pages 1882–1886. IEEE.

Zhao, H., Jiang, L., Jia, J., Torr, P. H., and Koltun, V. (2021c). Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268.

Zheng, H., Li, H., and Fan, Y. (2024). Coupled reconstruction of cortical surfaces by diffeomorphic mesh deformation. *Advances in Neural Information Processing Systems*, 36.

Zhou, S. K., Greenspan, H., and Shen, D. (2023). *Deep learning for medical image analysis*. Academic Press.

Zöllei, L., Iglesias, J. E., Ou, Y., Grant, P. E., and Fischl, B. (2020). Infant FreeSurfer: An automated segmentation and surface extraction pipeline for T1-weighted neuroimaging data of infants 0–2 years. *NeuroImage*, 218:116946.