SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Informatics

# Fast Surface Reconstruction and Mesh Correspondence for Longitudinal 3D Brain MRI Analysis

Jan Fecht

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Informatics

# Fast Surface Reconstruction and Mesh Correspondence for Longitudinal 3D Brain MRI Analysis

## Schnelle Oberflächenrekonstruktion und Gitterkorrespondenz für Longitudinale 3D-MRT-Analyse des Gehirns

| | |
|---|---|
| Author: | Jan Fecht |
| Supervisor: | Prof. Dr. Christian Wachinger |
| Advisor: | Fabian Bongratz |
| Submission Date: | 15th of December 2023 |

I confirm that this master's thesis is my own work and I have documented all sources and material used.

Munich, 15th of December 2023 Jan Fecht

# Acknowledgments

I want to thank Fabi for being an incredible advisor, always having an open ear for my questions and supporting me throughout this journey. I also want to thank Anne, for her support with setting up V2CC and fruitful discussions surrounding the topic of this thesis. Next, I want to thank Professor Wachinger for giving me the opportunity to work on this thesis and the whole lab for the making the time in the office more fun. This thesis would not have been possible without the compute-power provided by the Leibniz Supercomputing Centre (LRZ) and the CORE project at the LMU clinic, for which I am grateful. Finally, I want to thank my family and friends, in particular my girlfriend Viktorija, for their support and encouragement.

# Abstract

Deep cortical surface reconstruction methods have shown promising results in recent years, reconstructing mesh representations of the cortical surface boundaries with high accuracy, typically within seconds [101]. However, a critical gap remains: these methods do not focus on producing intra-subject aligned meshes in longitudinal studies. For downstream analysis of morphological changes, vertex correspondence – where the same anatomical location is consistently represented by identical vertices across all meshes – is crucial. This consistency is key to reducing noise and enabling smooth interpolation between meshes in both vertex and region-based analysis.

To bridge this gap, we present a new method called V2C-Long, which generates intra-subject aligned meshes through a novel three-step process. Initially, the surfaces of all time points of a subject are reconstructed independently, from which a median template for the subject is then created. Finally, a dedicated model deforms the template to the target surfaces, requiring only small adjustments in vertex positions, leading to highly aligned meshes. Our method builds on the V2C-Flow method, which produces state-of-the-art meshes in terms of reconstruction accuracy [6], but the general approach is adaptable to other mesh deformation-based cortical surface reconstruction methods. Given the lack of related work on this topic, we introduce a set of novel metrics to quanitfy the vertex correspondence, most of which are based on the variance of selected mesh properties across time points. Utilizing these metrics, we rigorously evaluate our methods against related mesh deformation methods, including FreeSurfer's longitudinal pipeline [75, 74], and show that our method produces meshes with the best correspondence scores, with a notable margin to other deep learning methods. In addition, our method significantly reduces the number of self-intersections compared to V2C-Flow. Based on our metrics, we anticipate that meshes produced by our method are well suited for downstream analysis, as their alignment is at least on par with, if not slightly better than the alignment of meshes produced by FreeSurfer's longitudinal pipeline, the de facto standard in the field.

# Contents

# 1. Introduction

## 1.1. Motivation

Recently, deep learning [43] methods in Cortical Surface Reconstruction (CSR) – transforming Magnetic Resonance Imaging (MRI) images into mesh-based cortical boundary representations – have yielded promising results in terms of accuracy and inference speed [101]. In longitudinal studies, multiple MRI scans of a subject are taken over time. Many downstream applications based on such longitudinal brain imaging studies require aligned cortical surface meshes across multiple, intra-subject surfaces to accurately compare vertex-wise features [39, 64, 56]. Cortical thickness is an example of such a relevant feature that can be used as an indicator for neurodegenerative diseases, such as Alzheimer's Disease (AD) [21, 85]. Unaligned meshes can result in high levels of noise in longitudinal measurements. In contrast, well-aligned meshes, i.e. meshes where a specific vertex is located at the same anatomical location in each mesh of a patient, lead to smoother trajectories of vertex-wise features along the time axis, which aids downstream analysis.

The de facto standard tool for CSR is FreeSurfer [25], which includes a dedicated pipeline for longitudinal analysis [48, 75, 76, 74]. However, to align the reconstructed meshes, FreeSurfer uses a spherical registration method that registers the meshes to a common space [76]. This process can take multiple hours per MRI image [72].

To the best of the author's knowledge, no work has investigated longitudinal mesh alignment in deep CSR methods so far, with only one work discussing general mesh correspondence for deep CSR methods [77]. Because of that, current deep learning-based methods typically reconstruct the cortical surfaces independently without considering longitudinal information, and do not aim to produce aligned meshes. For integrating these methods into longitudinal pipelines, the predicted meshes must thus be aligned in a post-processing step, for example using FreeSurfer's spherical registration [27]. Although there have been advances in deep learning-based spherical registration [78, 87, 102], the cortical surfaces still need to be inflated to a sphere first, which requires long computation times [72] and is prone to introduce errors such as distortions [18].

A subset of the recent deep learning methods use graph convolutions to deform a template mesh into the target surface [34, 17, 5]. Since the template mesh is shared across all reconstructions, the outputs of these methods are naturally aligned to some extent. In particular, the reconstructed surfaces have the same number of vertices and the same vertex connectivity. Also, due to the deformation process, corresponding predicted vertices may be located at similar anatomical positions. However, the often-employed Chamfer loss does not explicitly enforce mesh correspondence.

Figure 1.1 shows parts of reconstructed meshes from both FreeSurfer's longitudinal pipeline and V2C-Flow [6], the method we build upon. The two images on the right show the predicted meshes of two consecutive scans of a subject each, with the meshes on top being predicted by V2C-Flow, and the ones on the bottom by longitudinal FreeSurfer. In both cases, the same set of faces has been highlighted in both meshes to show the alignment of the meshes. It can be seen that the corresponding faces on the V2C-Flow meshes are in proximity, but do not align perfectly. Related deep CSR methods have similar problems. On the other hand, the corresponding faces on the FreeSurfer meshes are much better aligned.

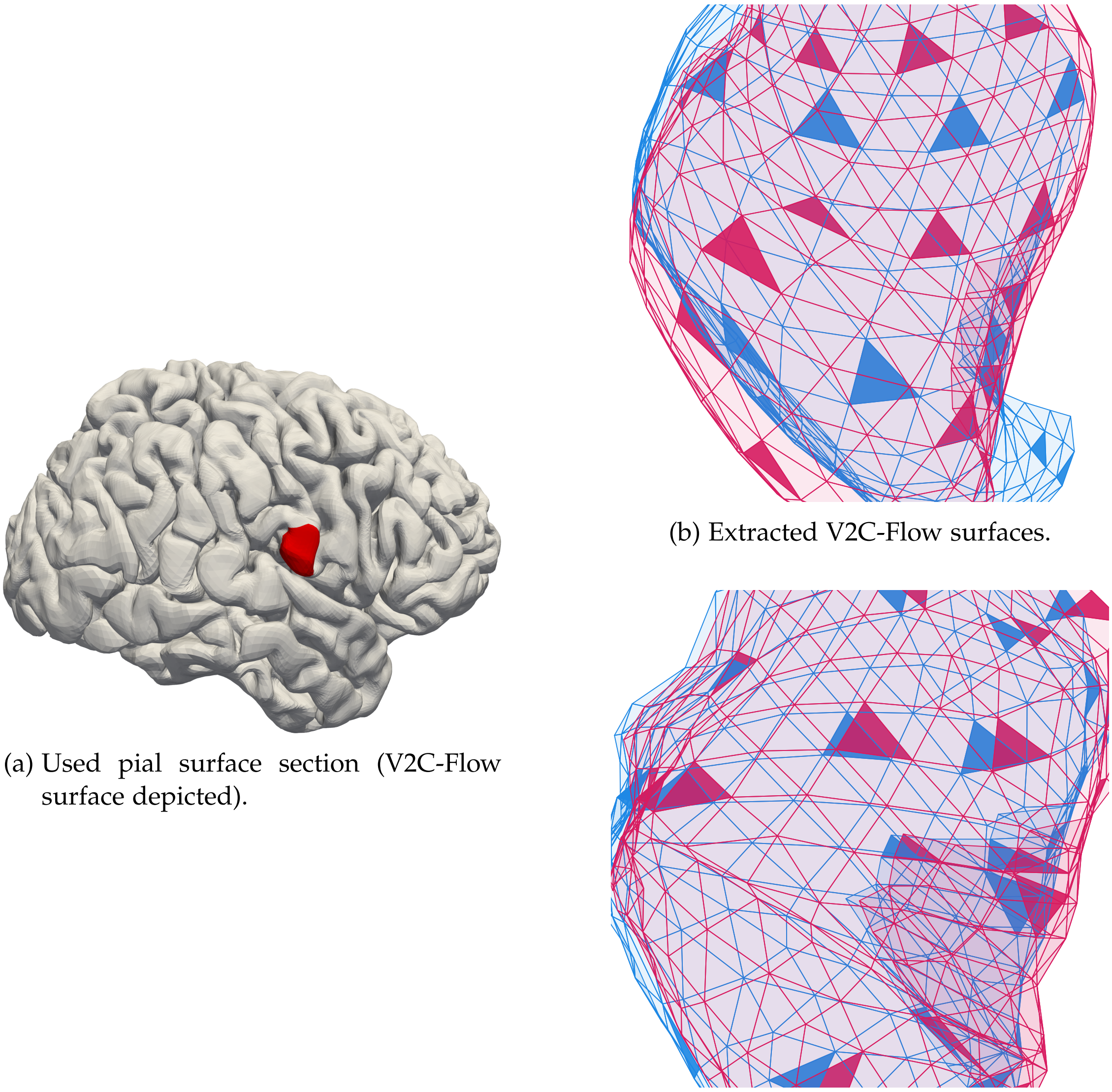
## 1.2. Contribution

In this thesis, we investigate the problem of longitudinal mesh alignment in deep CSR methods. Our main goal is to measure and improve the mesh correspondence, specifically in V2C-Flow [6], a state-of-the-art deep CSR method. To this end, our contributions are:

1. We introduce a new set of metrics to quantify the correspondence between meshes.
2. We present a new method, V2C-Long, that significantly improves mesh correspondence, even outperforming FreeSurfer. In addition, V2C-Long improves the reconstruction accuracy and reduces the number of self-intersecting faces compared to V2C-Flow. Our approach can be generalized to various other mesh-deformation based methods.
3. We perform two extensive ablation studies to investigate the effect of the different configurations of V2C-Long. We evaluate a final version of V2C-Long against related work on three longitudinal datasets. Finally, we show that V2C-Long outperforms related work in terms of mesh correspondence and lowers the number of self-intersections compared to V2C-Flow.

## 1.3. Thesis Outline

This chapter motivates and explains the relevance of longitudinal intra-subject mesh alignment. The next chapter introduces relevant mathematical and neuroimaging concepts and provides an overview over existing work related to CSR. Chapter 3 introduces our method, V2C-Long, and explain the training procedure and possible configurations of the method. Then, the chapter presents and motivates various new evaluation metrics used to quantify the extent of mesh misalignment. In Chapter 4, an extensive overview over the datasets, training and model setup, and evaluation procedures is given. In particular, the chapter talks about the setup of two ablation studies and a final comparison of our method to related work. The results of the ablation studies and comparison with related methods are presented and discussed in Chapter 5. Finally, Chapter 6 summarizes our findings, discusses potential limitations of the method, and gives some suggestions for future work.



(a) Used pial surface section (V2C-Flow surface depicted).

(b) Extracted V2C-Flow surfaces.

(c) Extracted FreeSurfer longitudinal pipeline surfaces.

Figure 1.1.: Mesh correspondence visualized on two pial surfaces of two intra-subject scans taken within 24 hours (TRT s1_01, s1_02). Image (a) highlights the portion of the pial surface where the submeshes were extracted. Images (b) and (c) show each the surfaces from both scans in the region with some of the faces highlighted. The better the highlighted faces overlap, the better the mesh correspondence. The meshes in (b) were produced by V2C-Flow. The meshes in (c) were produced by FreeSurfer's longitudinal pipeline. FreeSurfer achieves better correspondence than V2C-Flow, but V2C-Flow meshes are more regular and smooth due to regularization terms in the loss function.

# 2. Background and Related Work

This chapter presents related background knowledge essential for understanding this work. The first part focuses on the mathematical background, followed by a section on relevant medical imaging concepts. The third section gives an overview of existing methods in the field of cortical surface reconstruction. Finally, the last section touches on existing related work on deep learning in the context of longitudinal studies.

## 2.1. Mathematical Background

### 2.1.1. Polygon Meshes, Surfaces, and Graphs

Polygon meshes are a common representation of 3D objects in computer graphics. They provide an explicit, discrete representation of the surface of an object, which can be manipulated and rendered efficiently [8]. In this work, our focus is on 2-manifold polygon meshes with a triangular tessellation of the surface, since the meshes generated by Vox2Cortex and related methods (cf. Subsection 2.3.3) and our new model use this type of mesh, due to its flexibility and uniformity.

A mesh is said to be 2-manifold if it is topologically equivalent to a sphere (this topology is often called $S^2$ [22]). This further implies that the mesh is *watertight*, i.e., it does not contain holes [38, p. 25, Definition 2.8]. A triangular tessellation divides the surface into triangles faces, ensuring that there are no gaps between the faces. This type of tessellation has the Schläfli symbol $(3, 6)$, which means that the faces are triangular and each vertex is surrounded by 6 faces [94]. Figure 2.1 shows an example of a mesh with such a tesselation.

We define this type of mesh as a tuple $M = (V, F)$ where $V = \{v_1, \ldots, v_n\}$ is the set of vertices and $F$ is the set of faces. The vertices are typically represented as 3D vectors $v_i \in \mathbb{R}^3$ and the faces as triplets of vertex ids $f_k = (i, j, l) \in \mathbb{N}^3$ with $v_i, v_j, v_l \in V$. The edges of the mesh can be derived from the faces, since each face is composed of three edges. The edges are typically represented as pairs of vertex ids $e_k = (i, j) \in \mathbb{N}^2$ with $v_i, v_j \in V$.

Given the vertices and edges, a graph $G = (V, E)$ can be derived from the mesh. From the previous definitions, we can obtain the following properties of the graph using Euler's polyhedral formula [93]:

1. $deg(v_i) = 6$ 2. $|F| = 2|V| - 4$ 3. $|E| = \frac{3}{2}|F|$

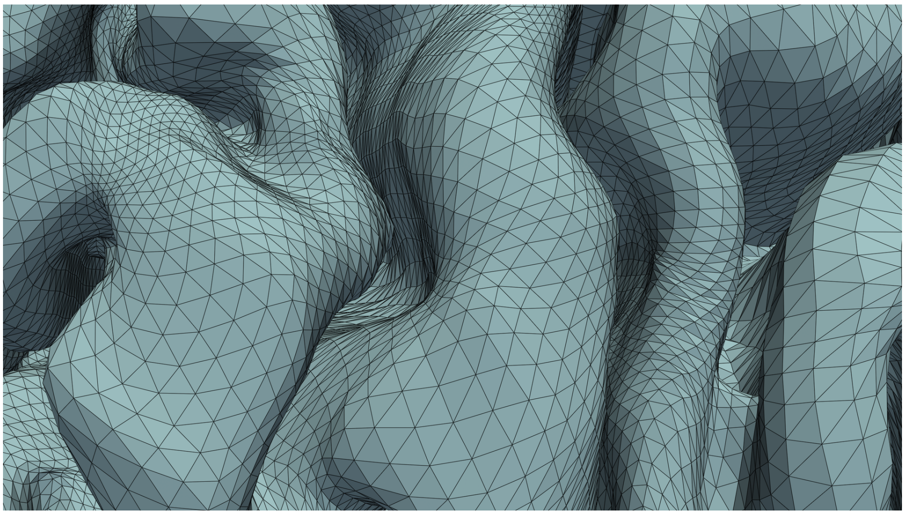


Figure 2.1.: Example cortical surface mesh with a $(3, 6)$ tessellation.

### 2.1.2. The Laplace-Beltrami Operator and the Mean Curvature

The *Laplace operator* $\Delta$ is defined as the divergence of the gradient of a function $f$ [38, p. 21]:

$$\Delta f = div \cdot \nabla f(x) = \frac{\partial^2 f}{\partial x_1^2} + \frac{\partial^2 f}{\partial x_2^2} + \frac{\partial^2 f}{\partial x_3^2} = \mathrm{tr}(\mathrm{Hess}(f)) \tag{2.1}$$

The Laplace-Beltrami operator generalizes the Laplace operator from Euclidean space to manifolds/curved spaces:

$$\Delta_{\mathcal{M}} f = \mathrm{div}_{\mathcal{M}} \cdot \nabla_{\mathcal{M}} f \tag{2.2}$$

Intuitively, it measures how much a function changes at a point, given the local curvature of the manifold $\mathcal{M}$.

To apply the Laplace-Beltrami operator to a mesh, it first needs to be discretized. This can be accomplished by approximating the gradient and divergence operators with a weighted average of neighboring vertices' function values. The weights are typically derived from the cotangent of the angles of the faces of the mesh [38, p. 29].

The *mean curvature* $H$ of a surface is one of many curvature measures used in shape analysis. It is defined as the average of the principal curvatures of a surface:

$$H = \frac{\kappa_1 + \kappa_2}{2} \tag{2.3}$$

The principal curvatures $\kappa_1$ and $\kappa_2$ are the minimum and maximum curvatures of a surface at a given point.

The mean curvature can also be expressed using the normal vector $\vec{n}$ of the surface:

$$H = \frac{1}{2}\mathrm{div}_{\mathcal{M}}\vec{n} \tag{2.4}$$

Similar to the Laplace-Beltrami operator, the mean curvature can be discretized by weighted averaging of neighboring normal vectors of the mesh vertices.

The discretized mean curvature is prone to outliers and is often smoothed by iteratively applying a weighted averaging of neighboring vertex function values.

### 2.1.3. Image Registration

Image registration is the process of aligning a set of images (referred to as *moving images*) to a common space using a reference image (called *fixed image*). It is a common task in medical image analysis, necessary for comparing images of different patients or for longitudinal comparisons of the same patient.

Image registration can be divided into three types, based on the transformation used. Each type can be used depending on requirements such as speed and distance-preservation.

1. **Affine Registration:** The transformation model is affine. It consists of a rotation, a translation and a scaling.

2. **Rigid Registration:** The transformation model is a special case of affine registration where the scaling is fixed to 1, which means that distances are preserved.

3. **Non-rigid Registration:** The transformation model is non-linear and can model more complex transformations.

Many medical imaging tools and libraries ship with implementations of image registration algorithms, such as FreeSurfer (see Subsection 2.3.1), `NiftyReg` [61] and `ANTs` [3].

Since we work with 3D images in this thesis, these transformations operate on voxels, which are the 3D equivalent of pixels. However, the transformation obtained from registering 3D Magnetic Resonance Imaging (MRI) images can also be applied to cortical meshes (as long as the coordinate systems match), allowing us to consistently align meshes with the underlying images.

A commonly used reference image is the *MNI152 template*, which is a reference image of a human brain that is commonly used in brain imaging research [24]. In this work, we use the 1-millimeter version of the template, which means that each voxel spans a volume of $1mm^3$. The resulting coordinate space is called *MNI152 space*.

## 2.2. Medical Imaging Background

### 2.2.1. Cortical Surfaces

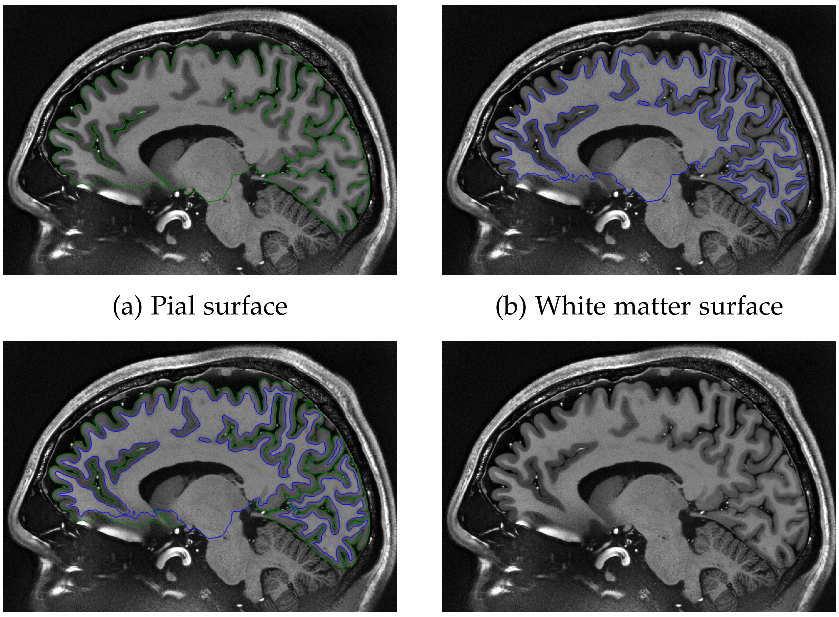
The cerebral cortex is a large, tightly folded region of the cerebrum responsible for higher cognitive functions [7]. It is subdivided into left and right hemispheres by the cerebral fissure. Because the cerebral cortex is located at the outer layer of the brain, it is surrounded by the Cerebrospinal Fluid (CSF), a colorless fluid that typically appears black on T1-weighted MRI images. The outer layer of the cerebral cortex consists of gray matter, while the inner region of the cortex consists of white matter.

Two types of boundaries are commonly used in the context of cortical thickness, which is defined as the distance between them:

1. The *pial surface* that separates the outer CSF from the gray matter.

2. The *white (matter) surface* that separates the gray matter from the white matter.

These surfaces exist for both hemispheres and are depicted in Figure 2.2. Consequently, there are four total surfaces that are typically used for thickness estimation: the left/right pial surface and the left/right white matter surface. In this work, however, we will focus on the right hemisphere only.



(a) Pial surface

(b) White matter surface

(c) Both surfaces

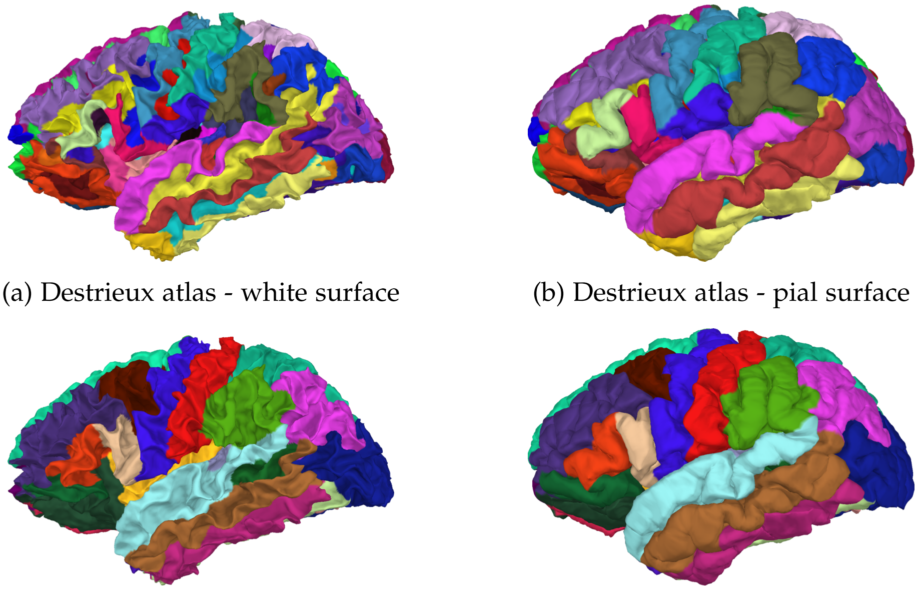
(d) Original image

Figure 2.2.: Sagittal slice view of the cortex highlighting the two types of surfaces. (a) Pial surface depicted in green. (b) White surface in blue. (c) Both surfaces, showcasing the gray matter sandwiched between them. (d) The original MRI from [52]. The surfaces were extracted using FreeSurfer v.7.2 [28, 25].

### 2.2.2. Brain Parcellations

The term *brain parcellation* refers to a process by which the cerebral cortex is divided into regions (called *parcels* or *regions of interest*) based on certain characteristics, often related to higher-level function or geometry. The specific set of regions is typically defined by a *brain atlas*, which is a collection of anatomical labels that are assigned to each region.

Commonly used atlases are the Desikan-Killiany atlas [19], and the Destrieux atlas [20]. The Desikan-Killiany atlas contains 34 regions per hemisphere, while the Destrieux atlas contains 74 regions per hemisphere. Both are shown in Figure 2.3.



(a) Destrieux atlas - white surface

(b) Destrieux atlas - pial surface

(c) Desikan-Killiany atlas - white surface

(d) Desikan-Killiany atlas - pial surface

Figure 2.3.: Destrieux vs. Desikan-Killiany atlas mapped to the white and pial surface. The images were rendered using MNE-Python v1.6.0 [32]

### 2.2.3. Cortical Thickness Estimation

The thickness of the human cerebral cortex is of great interest in neuroimaging research, as changes in it are closely related to neurodegenerative diseases such as AD [44, 69]. On average, it is approximately 2.5 mm thick and its thickness can vary widely between cortical regions [26]. The goal of Cortical Thickness Estimation (CTE) is to estimate the cortical thickness of the cerebral cortex from MRI images.

For accurate tracking of sub-millimeter cortical thickness changes, mesh-based representations of the cortex are required, as these changes are well below the typical 1 mm resolution of voxel-based images. In this work, we define cortical thickness as the distance between the white matter and pial surface and compute it vertex-wise by measuring the distance between each white matter vertex and the pial mesh.

### 2.2.4. Cross-Sectional and Longitudinal Studies

Cross-sectional studies are studies that are conducted at a single point in time providing a snapshot of a population at that time. Often, a group of subjects with a specific condition is compared to a healthy control group in order to find differences between them [36]. For example, the cortical thickness of a group of patients with Alzheimer's Disease (AD) may be compared to the cortical thickness of a control group of healthy subjects at a single point in time, as in [44].

Longitudinal studies, on the other hand, are conducted over a longer period of time, often several years. The same group of subjects is examined at multiple points in time, offering insight into the development of their brains or the progression of a disease. In the context of CTE, longitudinal studies look into the progression of cortical atrophy over time [80, 89, 1, 81]. Because patients are scanned multiple times over a period of time, a typical longitudinal dataset contains multiple MRI images for each patient. In this work, we refer to these images and the time at which they were acquired as *time points* starting with time point 1, which is the first scan of a patient (often called *baseline* scan) and then counting upward for each subsequent scan of the patient (often called *follow-up* scans).

## 2.3. Cortical Surface Reconstruction

The goal of CSR is to generate a three-dimensional, mesh-based representation of the cortical surface from a voxel-based MRI image. Having a mesh-based representation of the cortex allows for a more detailed analysis of the cortical surface compared to voxel-based representations. For example, cortical thickness, the degree of folding, and overall connectivity can be computed more accurately [18].

Traditionally, CSR is performed using hand-crafted statistical algorithms, which often involve time-consuming feature engineering and parameter tuning. In addition, due to their long computation times, these algorithms may not be suitable for large-scale studies [72]. In recent years, deep learning has been applied to CSR and has shown promising results in both inference speed and reconstruction accuracy [101]. In the following, we give an overview of relevant methods for CSR.

### 2.3.1. FreeSurfer

**Overview**

FreeSurfer [25] is currently the most widely used software-based tool for CSR and is the de facto standard in the field. It consists of different tools that can be used to perform various tasks related to CSR and CTE.

Typically, these tools are chained together in a pipeline that performs all necessary steps to generate cortical surfaces from an MRI image. Each pipeline step reads a set of relevant input files and produces a set of output files. For example, the spherical registration command (called `mris_register`) reads the inflated surfaces called `lh.sphere` and `rh.sphere` and produces the registered surfaces `lh.sphere.reg` and `rh.sphere.reg`.

### Cross-Sectional Processing

The cross-sectional processing pipeline (the `recon-all` pipeline) consists among other things of the following stages [71]:

- **Image Normalization:** There are several steps to normalize the image, such as intensity normalization, image registration, bias field correction, and skull stripping.
- **Tissue Segmentation:** The normalized image voxels are classified into different tissue types.
- **WM Cortical Surface Reconstruction:** The white matter surface is reconstructed using a combination of the segmentation results, tessellation and topology correction.
- **WM Spherical Registration:** The white matter surface is inflated to a sphere and registered to a common space using spherical registration.
- **WM Surface Parcellation:** The surface is parcellated using a brain atlas (cf. Subsection 2.2.2).
- **Pial Surface Reconstruction:** The pial surface is reconstructed using information from the white matter surface and the image.
- **Thickness Measurements and Parcellation Statistics:** The cortical thickness and other statistics are calculated.

The cortical thickness estimates computed by FreeSurfer are based on the distance between the pial and white matter surfaces. In a validation study, Cardinale et al. [9] compared FreeSurfer's cortical thickness estimates to in-vivo histologic measurements They found that FreeSurfer's closely matched the histologic data, with a difference in mean thickness of only 0.02 mm.

### Longitudinal Processing

For longitudinal analysis, FreeSurfer provides a special longitudinal pipeline that reconstructs the surfaces of multiple time points of a patient using information from all time points to improve reconstruction accuracy, reduce processing variability, and avoid over-regularization [76]. As a by-product, the reconstructed meshes have a higher correspondence across time points, meaning they have the same number of vertices and the same vertex connectivity and are generally better aligned (cf. Figure 1.1).

The longitudinal pipeline consists of the following stages [48]:

- **Base Template Creation:** The intensity-normalized images of each of the patient's time points are registered and averaged to create an unbiased patient base template. The median of the intensities of the images is used for averaging [76]. This average image is then processed using the standard `recon-all` pipeline to create cortical surfaces and other files.
- **Longitudinal Processing of Time Points:** A modified version of the `recon-all` pipeline is used to process each time point of the patient. Some operations are modified to use output from the base template creation step. In particular, the cortical surfaces are initialized using the base template surfaces. In addition, the target surface for the spherical registration step is the base template surface.

The algorithms used in FreeSurfer's longitudinal pipeline are described in detail in [75, 74, 76]. One of the main features of the pipeline is the creation of an unbiased patient template that avoids bias in both registration and in algorithm initialization [74].

#### Longitudinal Analysis

FreeSurfer includes several ways to perform statistical analysis on the data extracted from the longitudinal pipeline. The most advanced tool offered by FreeSurfer is a Matlab library for LinearMixedEffects models [49, 4], which has been used in multiple studies to investigate the effects of cortical thinning [31, 23].

#### The *fsaverage* Subject

The *fsaverage* subject is a special subject that is included with FreeSurfer. It was created by averaging the cortical surfaces of 40 subjects [86]. FreeSurfer also provides a parcellation for the *fsaverage* subject, which maps each vertex of the *fsaverage* subject to a region of the Desikan-Killiany atlas and the Destrieux atlas (cf. Subsection 2.2.2). It is commonly used as a reference subject for registration or for visualization [37, 96].

### 2.3.2. Deep Cortical Surface Reconstruction Methods

CNNs have successfully solved many computer vision tasks, due to the widespread availability of large datasets and the increased computational power of GPUs [43]. Architectures such as U-Nets [79] have been used to solve a wide range of segmentation tasks in medical imaging [46]. Based on this, several deep learning methods for CSR have been proposed in recent years. Since there is little ground truth data available for training, most of these methods use synthetic data generated from FreeSurfer's `recon-all` pipeline [101] (cf. Subsection 2.3.1).

Deep CSR methods can be roughly divided into three categories: voxel-based, implicit surface-based and explicit mesh-based methods. In the following, we give an overview of these methods.

### Voxel-Based Methods

In voxel-based methods, the output of a segmentation network is directly converted into a mesh using a marching cubes or similar algorithm, which can often be implemented in an efficient way [45, 50]. However, marching cubes reconstructions are known to be prone to artifacts and topology errors that require expensive post-processing to fix [5, 88]. In addition, the accuracy of the reconstruction is limited by the resolution of the input image. An example of a voxel-based CSR method can be found in [73].

### Implicit Surface-Based Methods

Implicit surface-based methods represent the cortical surface as a Signed Distance Function (SDF) [97, 59] which is a function that maps any point in space to a signed distance from the surface. To obtain an explicit mesh from the SDF, which may be needed for downstream tasks, the SDF must be converted using specialized algorithms, such as marching cubes. These types of algorithms often lead to similar problems as with the voxel-based methods [5]. Examples of methods using implicit surfaces are SurfNN [88], DeepCSR [73], PialNN [54] and SegRecon [88].

### Explicit Mesh-Based Methods

Explicit mesh-based methods represent the cortical surface as a mesh and use a neural network, typically based on graph convolutions, to deform a template mesh into the target mesh. These methods typically produce smoother meshes than voxel-based and implicit surface-based methods, and preserve the topology of the template mesh, but suffer from self-intersecting faces. Another advantage of template deformation is that the resulting meshes have the same number of vertices and the same vertex connectivity as the template mesh, which allows for vertex-to-vertex comparison between meshes.

Voxel2Mesh [95] is an early example of this approach, where a spherical template mesh is deformed into the target mesh, although it does not focus on cortical surfaces. Topofit [34] is another example of a mesh-based method that uses a template mesh and a set of features extracted from the MRI image to deform the template mesh into the target mesh. Topofit is shipped with FreeSurfer and can be conveniently integrated into its pipeline [90]. Vox2Cortex and its variants (cf. Subsection 2.3.3), which will be discussed in the next section, are other examples of explicit mesh-based methods.

### Diffeomorphic Deformation Fields

A common approach to mesh-based CSR is to model a deformation field that transforms a template mesh into the target mesh. A special category of deformation fields are *diffeomorphic* deformation fields, which are bijective and smooth [53]. These diffeomorphic deformation fields provide theoretical guarantees that the transformed mesh is topologically equivalent to the template mesh and contains no self-intersecting faces [42, Theorem 3.1.]. Multiple deep CSR methods use diffeomorphic deformation fields to deform a template mesh into the target mesh by integrating a neural ordinary differential equation [10]. Examples of these methods are CortexODE [53], CorticalFlow++ [17] and DD-SWD [41].

CortexODE uses a hybrid mesh representation. It starts by constructing an SDF from an initial segmentation to estimate an explicit template mesh. This template mesh is then deformed into a white matter surface using a diffeomorphic deformation field. In a final step, the white matter surface is deformed into a pial surface using a second diffeomorphic deformation field [53]. Both the CortexODE and DD-SWD authors report that CortexODE is the leading model in terms of reconstruction accuracy (ASSD scores) [41, 53], and Bongratz, Rickmann, and Wachinger [6] report that it is on par with V2C-Flow.

CorticalFlow++ is another explicit mesh deformation method that instead of deforming vertices directly, learns a diffeomorphic deformation field in image space that transforms the template mesh into the target mesh [17, 42]. The deformation is guided by a set of features extracted from the MRI image by a U-Net.

Although there exist theoretical guarantees, because of discretization, limited mesh resolution, and the usage of the Chamfer loss, the resulting meshes of CortexODE and CorticalFlow++ still contain self-intersecting faces, although significantly less than Vox2Cortex [41, 53]. DD-SWD, on the other hand, uses probability measures to represent the meshes and the sliced Wasserstein distance [68] as a loss function, resulting in meshes with almost no self-intersecting faces (less than $10^{-4}$%) [41].

### 2.3.3. Vox2Cortex, V2C-Flow and V2CC

**Vox2Cortex**

Vox2Cortex (V2C) is a deep learning method for CSR that was first introduced by Bongratz et al. [5]. The proposed method uses a U-Net architecture [79] to segment an input MRI image into different tissue types. Additionally, a set of template meshes are deformed using graph convolutions into the pial and white matter surfaces for both hemispheres. The latent U-Net features are interpolated to the mesh vertices and used to guide the graph convolutions.

The model is trained using a combination of a segmentation loss and a mesh reconstruction loss. The mesh reconstruction loss includes regularization terms and a modified Chamfer distance [5]. The ground truth segmentations and meshes are generated using FreeSurfer’s cross-sectional reconstruction pipeline.

The template meshes used as input are generated by iteratively applying a Laplacian smoothing operation to the vertex positions of randomly chosen FreeSurfer surfaces. Each template mesh contains approximately 168,000 vertices, making the model computationally expensive to train and use.

The meshes produced by the model achieved state-of-the-art performance in terms of accuracy and computation time compared to other methods at the time of publication [5]. However, the generated meshes contain self-intersecting faces which can be removed in a post-processing step using mesh repair tools such as MeshFix [60].

Bongratz et al. also show that the meshes produced by the model have a higher reconstruction consistency than other methods, including FreeSurfer, when reconstructing MRI images of the same subject acquired within a short period of time [5].

### V2C-Flow

V2C-Flow is an extension of V2C introduced by Bongratz, Rickmann, and Wachinger [6]. It models the deformation of the template meshes as a continuous flow field, computed using blocks of neural ordinary differential equations [10]. Instead of using a random, smoothed FreeSurfer surface as input, the surfaces from the *fsaverage* FreeSurfer subject are taken, which allows for easier integration of the model into existing FreeSurfer pipelines and automatically yields the parcellation labels for the predictions. V2C-Flow predictions achieve better ASSD and Hausdorff distance scores (cf. Subsection 4.6.2) on most surfaces than the original V2C model and other state-of-the-art methods [6].

### V2CC

Another extension of Vox2Cortex is V2CC, which was introduced by the same authors as V2C-Flow in [77]. The model architecture used by V2CC is very similar to that used in Vox2Cortex. As in V2C-Flow, the template meshes are smoothed *fsaverage* FreeSurfer meshes. In addition, the authors evaluate their method with FreeSurfer's *fsaverage6* meshes, which are significantly smaller than the *fsaverage* meshes (40,962 vs 163,842 respectively) and therefore computationally more efficient. V2CC introduces a new loss function which is based on a vertex-wise L1 distance between the predicted mesh and a resampled ground truth mesh. Resampling is necessary, because the meshes produced by FreeSurfer's cross-section reconstruction pipeline do not have the same number of vertices.

Given fsaverage's vertex-to-parcellation label mapping, V2CC predictions achieve a significantly better parcellation label overlap when computing a DICE score against the FreeSurfer ground truth compared to other mesh deformation methods such as Vox2Cortex and CorticalFlow++ [77]. Furthermore, V2CC meshes have a lower root mean square deviation of the vertex positions for intra-subject predictions in scans from a longitudinal test-retest study [55] (cf. Section 4.1) compared to Vox2Cortex and CorticalFlow++. This indicates that meshes generated by V2CC are more aligned across time points than meshes generated by previous methods. However, the V2CC pial surface meshes contain on average more self-intersecting faces than those generated by Vox2Cortex and CorticalFlow++ [77]. Since V2CC is, to the best of our knowledge, the only deep CSR method directly targeting mesh correspondence, it is an important baseline for our work and will be compared to our final method (cf. Subsection 4.5.4).

Figures 2.4 and 2.5 show the different models and their inputs and outputs. An asterisk (★) marks meshes that contain the same number of vertices and have the same vertex connectivity. This allows for a direct vertex-to-vertex comparison between the meshes. However, as shown in Figure 1.1, the meshes produced by V2C-like models are not aligned sufficiently for precise vertex-wise comparison.

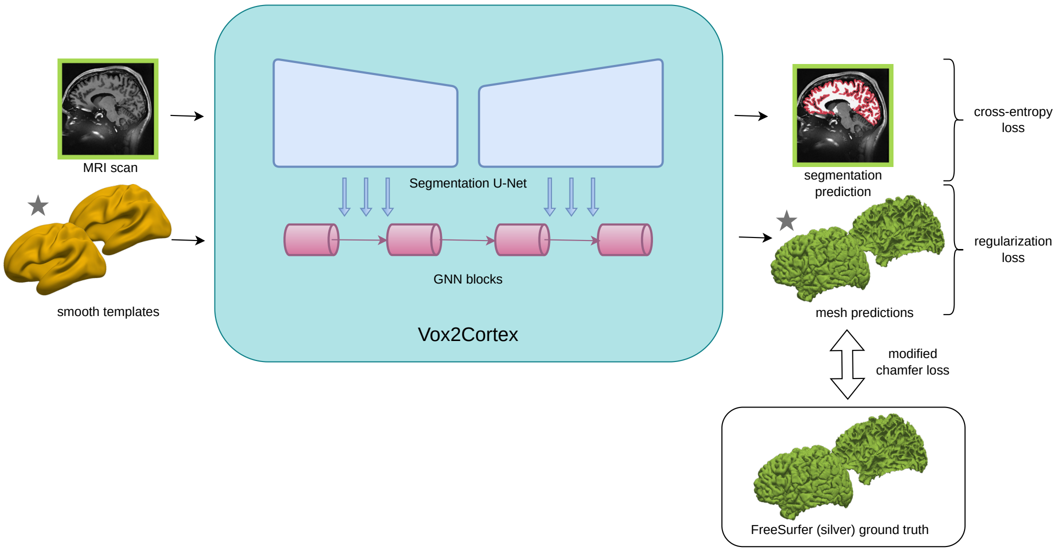


Figure 2.4.: Vox2Cortex model. The model takes a 3D MRI image as input and can predict pial and white surfaces simultaneously. Vox2Cortex uses a fixed set of four GCN blocks to deform the template mesh. ★: Meshes marked with an asterisk contain the same number of vertices and the same vertex connectivity.

## 2.4. Deep Learning in Longitudinal Neuroimaging

There is little literature on deep learning methods specifically focused on longitudinal neuroimaging data. Ouyang et al. [65] use a self-supervised approach to learn a latent space of longitudinal MRI images called Longitudinal Neighborhood Embedding, where trajectories in the latent space represent morphological changes in the brain. This is done through a combination of contrastive learning and a similarity score inspired by pedestrian trajectory prediction. The embeddings are then shown to lead to better results in downstream tasks.

Liu et al. [47] use an explicit mesh deformation approach to predict missing infant scans from longitudinal datasets. This is achieved by creating local geodesic coordinate grids with uniform, interpolated virtual vertices, and defining convolution operations on them. These convolutions are then used to predict vertex-wise growth trajectories that can be used to determine the pial and white surface at the next time point. The model is trained using a combination of a vertex-wise $L^2$ loss and a cortical thickness loss. They show that the model outperforms a linear affine transformation model.

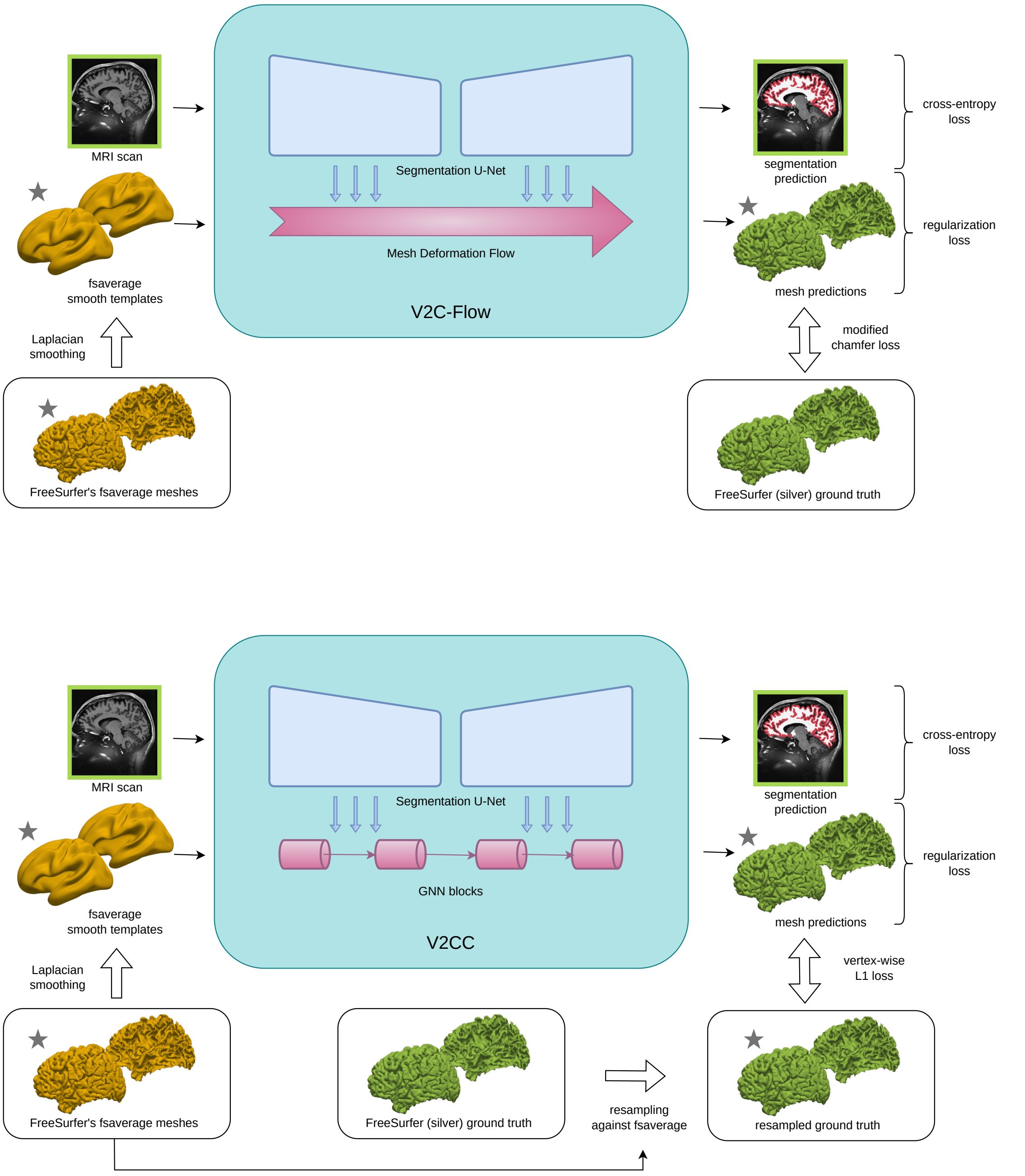


Figure 2.5.: V2C-Flow and V2CC models. The V2C-Flow model uses blocks of continuous flow fields while the V2CC model uses a vertex-wise L1 loss against resampled ground truth meshes. ★: Meshes marked with an asterisk contain the same number of vertices and the same vertex connectivity.

# 3. Method

## 3.1. Overview of V2C-Long

The main idea behind our method is to leverage longitudinal information by replacing the static templates in previous V2C variants with a patient-specific and potentially time point-specific mesh generated by another V2C-style model (the *Template Generation Model* (TGM)). This significantly shortens the deformations needed to transform the template to the target mesh, which we hypothesize results in better vertex/mesh correspondence between the template and the prediction. In addition, the final model can incorporate patient-specific longitudinal information as additional input to the model, including features extracted from the template generation process. This is possible because we use V2C-Flow [6] as the underlying model on which our method is built which can be easily modified to accept additional input with each vertex. We call this final model *V2C-Long*.

The complete process behind this model requires three separate steps:

1. **Template Generation:** The necessary templates are generated from the Template Generation Model (TGM) (a V2C-Flow model) and a static input mesh.

2. **Pairing Generation:** Template-target time point pairs are generated for each patient based on a pairing mode.

3. **Training and Inference:** The final model predicts the target mesh with the respective individual input template. The predicted mesh contains correspondences to the template mesh.

All of these steps are performed separately, which means that the TGM is trained independently of V2C-Long. Each step is described in more detail below.

### 3.1.1. Template Generation

An important property of the templates for the V2C-Long model is that the number of vertices and their connectivity remain the same for all templates used in the final model. This is necessary because the final V2C-Long model outputs meshes with the same number of vertices and connectivity as the input template meshes. The shared vertex connectivity and vertex number are fundamental requirements for vertex correspondence.

One way to achieve this is to run a mesh-deformation Cortical Surface Reconstruction (CSR) method on a single, static template mesh and predict meshes for each MRI scan in the dataset. In this work, a V2C-Flow (cf. Subsection 2.3.3) model is used, as it predicts more accurate meshes than Vox2Cortex [6] and allows for vertex-wise addition of additional input features. The template generation process is illustrated in Figure 3.1.

In addition, inspired by Reuter and Fischl [74], median and mean patient templates are derived from the generated templates. We compute these templates by patient-wise averaging the vertices of the registered meshes. No additional registration is performed.

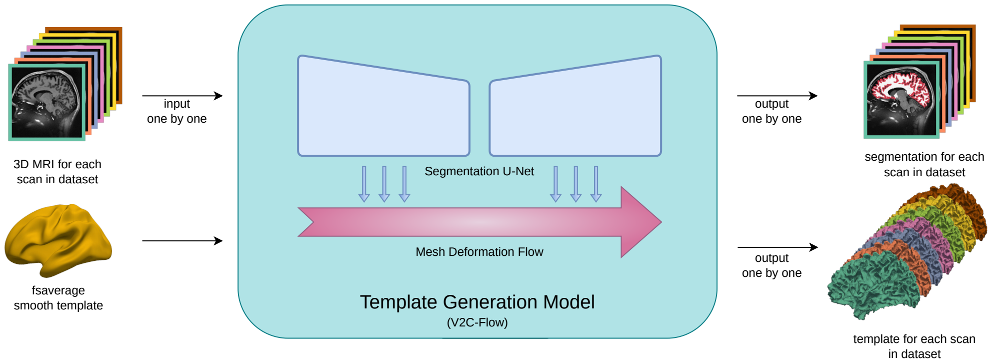


Figure 3.1.: Template generation process. The Template Generation Model (TGM) is trained by deforming the same, static template as in the V2C-Flow method [6] and predicts a mesh for each MRI scan in the dataset (including all splits). The predicted meshes are then used as templates for the final, V2C-Long model.

### 3.1.2. Pairing Generation

#### FIRST/MEDIAN/MEAN Mode

In the FIRST/MEDIAN/MEAN modes, the template used per patient remains constant across all scans of the patient. The template is either the template mesh corresponding to the first scan of the patient (FIRST) or the mean/median mesh of the patient calculated by taking the vertex-wise mean/median coordinates of all the templates for all the scans of the patient (MEAN/MEDIAN). Since the template is shared across all scans of the patient, the vertex displacements learned by the model start from the same set of vertices and which results in in strong vertex correspondence across all final predictions of a patient. However, the FIRST mode may be biased toward the first time point, and the MEDIAN/MEAN modes may be biased toward the middle.

### PREV/PREV_CHAIN Mode

In the PREV and PREV_CHAIN modes, the template used for a scan is a mesh predicted for previous scan/time point of the patient. As a result, the first time point is not predicted by the final model and instead, the first template mesh can be interpreted as the first prediction. The exact input mesh used in later steps differs depending on the mode: In PREV mode, the input mesh used for each prediction is the template mesh generated by the TGM. This means that the input vertices do not correspond well across the predictions, resulting in weak vertex correspondence between the predicted vertices. In PREV_CHAIN mode, however, the input mesh used is the actual mesh prediction of the previous time point. This leads to a chain of predictions where each prediction is based on the previous one, starting with the first mesh generated by the TGM. In this mode, the model actually learns the trajectory of the mesh over multiple time points which might lead to strong vertex correspondence across all of a patient's predictions. However, the variance in vertex positions may accumulate over time, which could also weaken the correspondences over multiple time points.

### NxN/NxN_SORTED Mode

The NxN and NxN_SORTED modes are special in that they do not use a single template mesh per time point, but instead generate multiple template-target pairings for a single time point. The NxN mode generates all possible combinations of template-target pairings, while the NxN_SORTED mode only generates pairings where the template is from an earlier or the same time point as the target time point.

This means that only a subset of the predicted meshes share the same template mesh, which means that the vertex correspondences across all predictions of a patient are weaker than in the FIRST/MEDIAN/MEAN modes. However, the NxN modes generate significantly more pairings than the other modes, leading to more training data for the model which , in turn, may improve training performance. On the other hand, this makes both training and inference computationally expensive, as for a patient, the number of pairings grows quadratically with the total number of scans.

The mesh predictions for a given time point can be aggregated to a single mesh using mean/median vertex positions. This may potentially restore vertex correspondence across these aggregated meshes and may also lead to more accurate and regular mesh predictions.

Table 3.1 provides an overview of the different pairing modes including a mathematical definition of the set of pairings for each mode. Note that we allow different numbers of total scans per patient in a dataset, which means that the number of pairings generated by each mode may vary across patients in the dataset. For a visual overview of the pairing modes, see Figure 3.3, which shows the template-target pairings generated by each mode for a patient with 4 scans. Figure 3.2 shows the pairing generation process for a patient with 4 scans and the NxN mode. It also shows the additional features generated for each pair that can be used for the final V2C-Long model.

| mode | pairings | # pairings | Notes |
|---|---|---|---|
| FIRST | $\{(T_1, I_i) \mid i \in \{1, \dots, N_{\text{scans}}\}\}$ | $N_{\text{scans}}$ | |
| MEAN | $\{(T_{\text{mean}}, I_i) \mid i \in \{1, \dots, N_{\text{scans}}\}\}$ | $N_{\text{scans}}$ | |
| MEDIAN | $\{(T_{\text{median}}, I_i) \mid i \in \{1, \dots, N_{\text{scans}}\}\}$ | $N_{\text{scans}}$ | |
| PREV | $\{(T_{i-1}, I_i) \mid i \in \{2, \dots, N_{\text{scans}}\}\}$ | $N_{\text{scans}} - 1$ | |
| PREV_CHAIN | $\{(P_{i-1}, I_i) \mid i \in \{2, \dots, N_{\text{scans}}\}\}$ | $N_{\text{scans}} - 1$ | $P_1 \leftarrow T_1$ |
| NxN | $\{(T_i, I_j) \mid i, j \in \{1, \dots, N_{\text{scans}}\}\}$ | $N_{\text{scans}}^2$ | |
| NxN_SORTED | $\{(T_i, I_j) \mid i, j \in \{1, \dots, N_{\text{scans}}\}, i \leq j\}$ | $\frac{N_{\text{scans}}^2 + N_{\text{scans}}}{2}$ | |

Table 3.1.: Overview of the pairing modes. $N_{\text{scans}}$: number of time points (varies across patients). $T_i$: template for $i$-th time point of patient, $I_i$: $i$-th MRI image, $P_i$: prediction of $i$-th time point of patient.

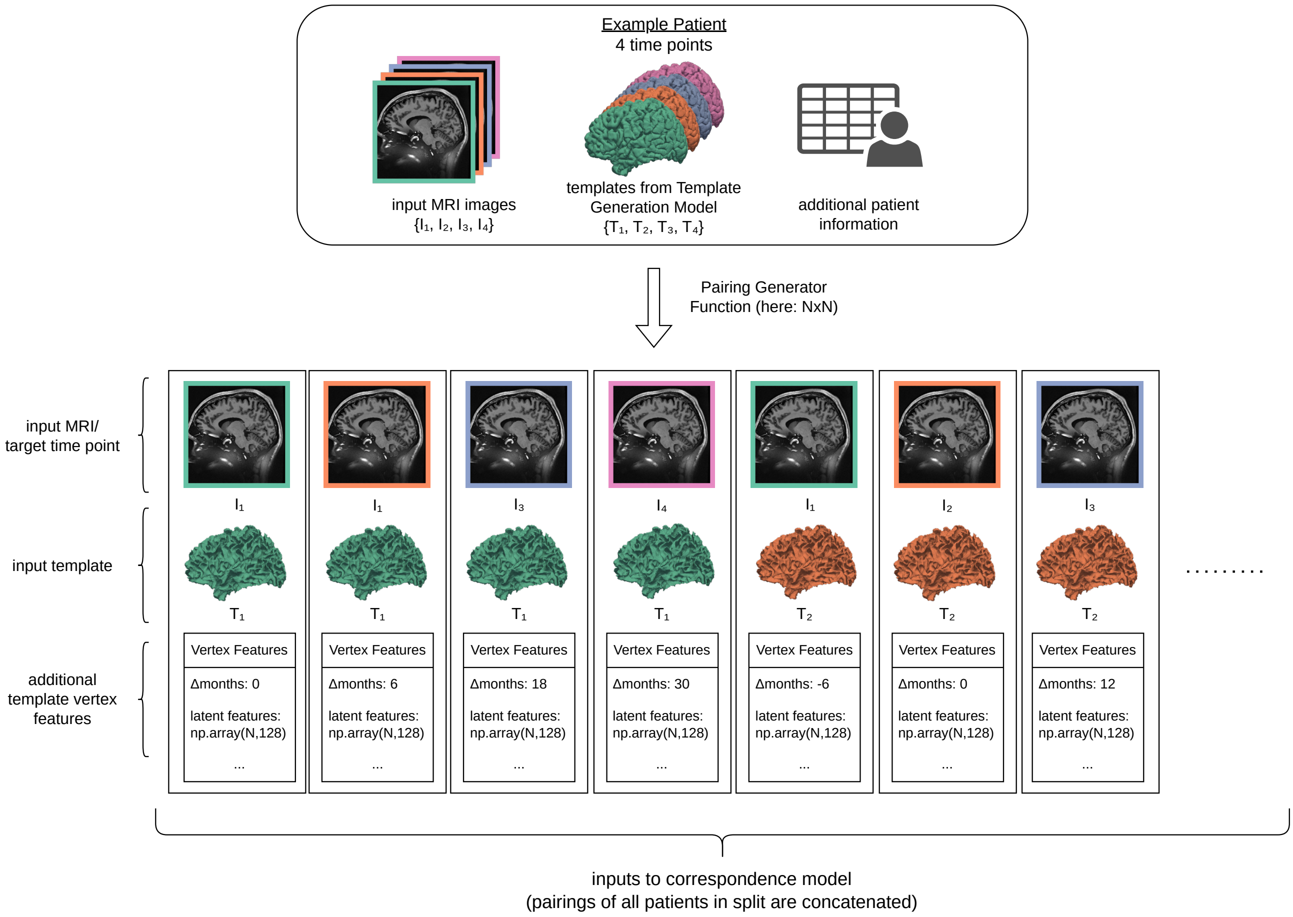

Example Patient
4 time points
input MRI images
{I₁, I₂, I₃, I₄}
templates from Template Generation Model
{T₁, T₂, T₃, T₄}
additional patient information
Pairing Generator Function (here: NxN)
input MRI/ target time point
input template
additional template vertex features
I₁
I₁
I₃
I₄
I₁
I₂
I₃
T₁
T₁
T₁
T₁
T₂
T₂
T₂
Vertex Features
Δmonths: 0
Δmonths: 6
Δmonths: 18
Δmonths: 30
Δmonths: -6
Δmonths: 0
Δmonths: 12
latent features: np.array(N,128)
...
.........
inputs to correspondence model
(pairings of all patients in split are concatenated)

Figure 3.2.: Pairing generation process visualized for a patient with 4 scans and the NxN mode. Only the first 7 pairings of the 16 total pairings are shown. After generating the pairings for all patients, they are concatenated to form the actual dataset for the V2C-Long model.

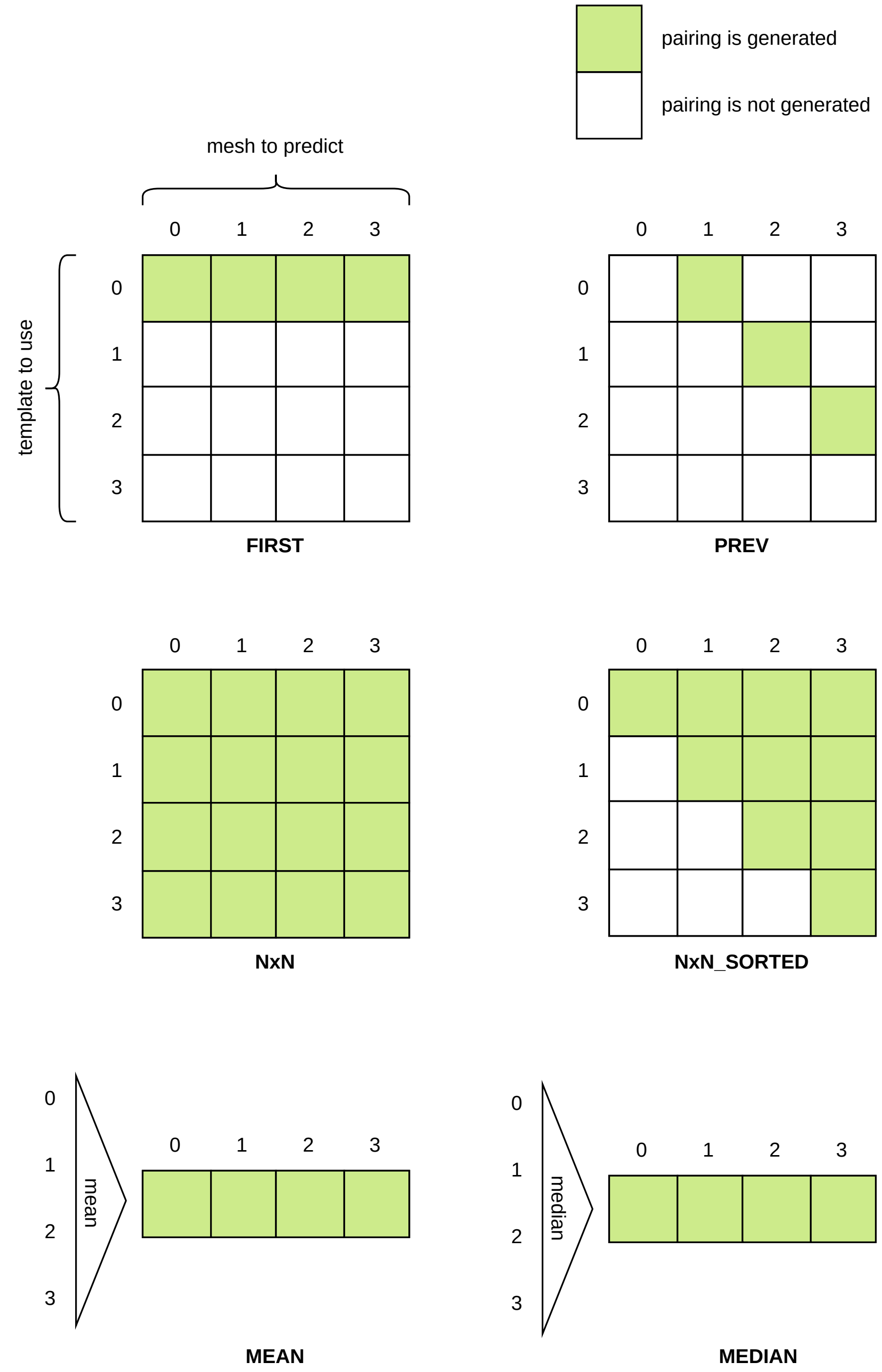


Figure 3.3.: Visual overview of the template-scan pairs generated by each mode for a patient with 4 scans. The numbers represent the ordered time points. The combinations generated by the mode are highlighted in green. The NxN and NxN_SORTED modes generate significantly more pairs and the PREV mode slightly less pairs than the other modes.

### 3.1.3. Training and Inference

In the last step, the V2C-Long model is trained using the generated pairings as training items. This means that for each (template, target) time point pair, we use the generated template mesh of the template time point as the input together with the MRI scan of the target time point. The loss is then computed against the ground truth segmentation and the ground truth mesh of that time point.

We have also extended the architecture of V2C-Flow to optionally attach additional features to the input vertices of the template. These features can be arbitrary, but in our setup we evaluate the model with the following:

- **Time difference**: The (signed) number of months between the template and the target time point is attached to each vertex as an additional input feature. If the template is from a later time point than the target, the number of months is negative. Ideally, the model can use this information to better predict the target mesh by relating time differences to morphological changes in the brain. For the MEAN and MEDIAN templates, we assign each template the mean (in both cases) of all time points of the respective patient.

- **TGM features**: We extract internal latent features of the Template Generation Model (TGM) by concatenating the outputs of each graph deformation layer. The features extracted for the target time point are then added to the template vertices for the V2C-Long model. Since these features encode the brain anatomy of the scan, the idea is that they can help the V2C-Long model to better predict the target mesh. For the MEAN and MEDIAN templates, we use the element-wise mean and median features across all time points.

For inference, the same pairings are generated as for training, but the ground truth segmentation and mesh are not used since no loss is computed. In the case of the NxN and NxN_SORTED modes, there may be multiple predictions for one time point. In this case, we additionally build an ensemble prediction by taking the mean or median vertex coordinates of the those predictions. This aggregation operation is applied along the columns of the NxN and NxN_SORTED plots in Figure 3.3.

A visual overview of the final V2C-Long model is shown in Figure 3.4.

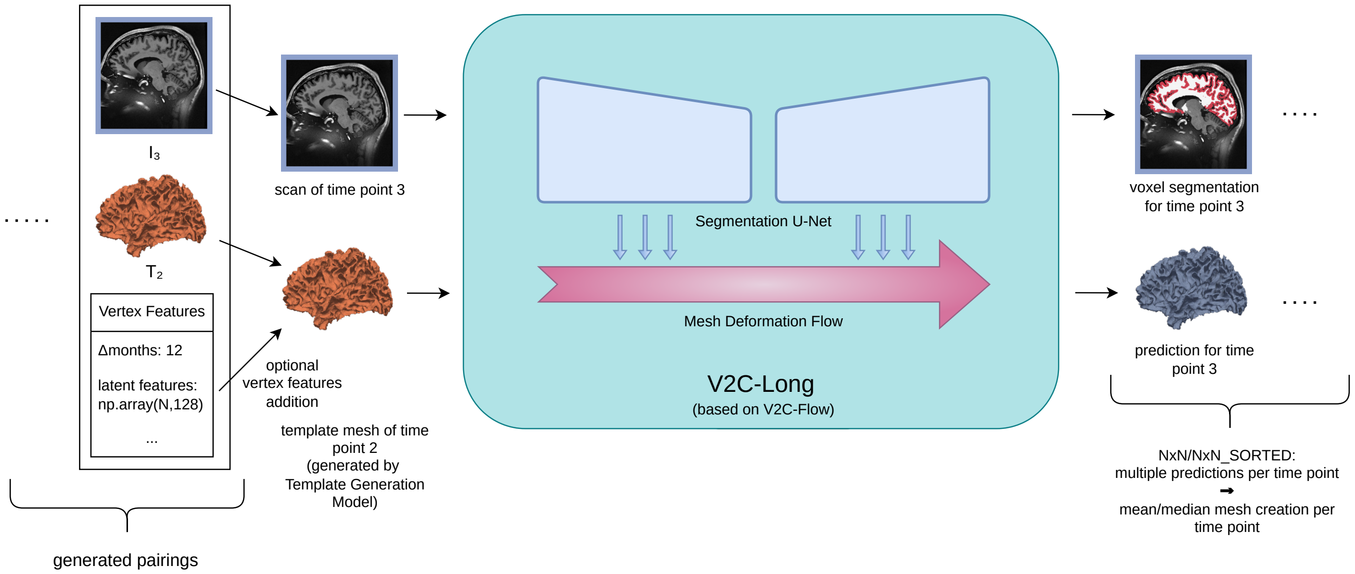

.....
$I_3$
$T_2$
Vertex Features
Δmonths: 12
latent features: np.array(N,128)
...
generated pairings
scan of time point 3
optional vertex features addition
template mesh of time point 2 (generated by Template Generation Model)
Segmentation U-Net
Mesh Deformation Flow
V2C-Long
(based on V2C-Flow)
voxel segmentation for time point 3
....
prediction for time point 3
....
NxN/NxN_SORTED: multiple predictions per time point
➡
mean/median mesh creation per time point

Figure 3.4.: The final V2C-Long model. The generated pairings are used to train the model, which has the same internal architecture as V2C-Flow (cf. Subsection 2.3.3). Additional longitudinal information can be added to the template vertex features. Since the NxN and NxN_sorted modes generate multiple predictions per time point, these predictions can additionally be aggregated using mean or median operations on the vertex coordinates.

## 3.2. Evaluation Metrics for Mesh Correspondence

Due to the lack of literature and existing metrics for mesh correspondence evaluation, we propose a set of new metrics to evaluate the quality of correspondence between a set of meshes. These metrics are mostly based on the comparison vertex-wise features, including features that depend on the absolute position of the mesh (the Mean Vertex Distance ($V_{L2}$) and Longitudinal Parcellation Consistency (ParcF1)) and features that compare the intrinsic shape and structure of the mesh, such as the Cortical Thickness Variance (ThVar), Mean Curvature Variance (MCVar), and Edge Length Variance (EdgeVar). As the underlying anatomy of the brain changes over time, a change in these features is expected. However, we expect the change to be small compared to the one introduced by the mesh deformation models.

The key difference between these metrics and typical CSR evaluation metrics is that they do not compare a predicted mesh to a ground truth mesh, but instead compare a set of mesh predictions to each other. Since we are focusing on longitudinal analysis, we apply these metrics in a patient-wise manner, which means that we compare the predictions for a patient internally against each other.

Note that some of these metrics may depend on the scale of the meshes and the total number of vertices per mesh used in the evaluation. This is important to note, as it may limit the comparability of the metrics across different datasets and methods.

Also, note that these metrics have been established before or during the development of the V2C-Long method and before any of the evaluations presented in Chapter 5.

### 3.2.1. Variance-Based Correspondence Metrics

The first set of metrics we propose is based on the variance of the vertex or edge-wise features. This is motivated by the fact that in a realistic setting, morphological changes in the brain are expected to be smooth, resulting in low variance in the vertex-wise features over time.

To compute these metrics, we first compute the feature for each vertex or edge in each mesh. Then, we compute the variance of these features across all meshes for each vertex or edge. To get a single number, we use the median of these variances across all vertices or edges. The median is preferred to the mean because features such as the discrete mean curvature are very sensitive and may lead to outliers that can propagate into the variance and mean of the variance.

So overall, the typical computational steps are:

1. **Feature Computation** (for each mesh and vertex/edge)
2. **Variance Computation** (for all meshes)
3. **Median Computation** (over all vertices/edges)

In the following, we will describe the first two steps in a formal way. Let $N_\mathrm{M}$ be the number of meshes we are comparing and $N_\mathrm{V}$ and $N_\mathrm{E}$ be the number of vertices and edges in the meshes respectively. $v_{i,j}$ is the $j$-th vertex of the $i$-th mesh and $e_{i,j}$ is the $j$-th edge of the $i$-th mesh containing a pair of vertex indices.

#### Edge Length Variance (EdgeVar)

The Edge Length Variance (EdgeVar) metric measures how much the lengths of corresponding edges differ between the meshes. The edge length is here defined as the Euclidean distance between the two vertices of an edge.

A good match of edge lengths makes it likely that the vertices and faces of the meshes are well aligned. However, relative "shifts" across the mesh or mesh regions may not be well captured by this metric, since no absolute position information is directly used.

To compute the metric, the length of each edge in each mesh is computed first:

$$\mathrm{length}_{i,j} = \|v_{i,e_{i,j}(2)} - v_{i,e_{i,j}(1)}\| \in \mathbb{R}^{N_\mathrm{E}} \tag{3.1}$$

The edge lengths are then normalized by the mean edge length in the mesh containing them, to make the metric more robust to different mesh scales and changes in the surface area due to morphological ground truth changes:

$$\mathrm{norm\_length}_{i,j} = \frac{\mathrm{length}_{i,j}}{\frac{1}{N_\mathrm{E}} \sum_{k=1}^{N_\mathrm{E}} \mathrm{length}_{i,k}} \tag{3.2}$$

Finally, the variance of the normalized edge lengths is computed for each edge across all meshes:

$$\text{EdgeVar}_j = \frac{1}{N_\text{M}} \sum_{i=1}^{N_\text{M}} (\text{norm\_length}_{i,j} - \overline{\text{norm\_length}_j})^2 \tag{3.3}$$

with $\overline{\text{norm\_length}_j}$ being the average normalized edge length of the $j$-th edge across all meshes ($\frac{1}{N_\text{M}} \sum_{i=1}^{N_\text{M}} \text{norm\_length}_{i,j}$).

### Cortical Thickness Variance (ThVar)

The variance of cortical thickness values is a metric that measures how much the cortical thickness differs between the mesh predictions. Since the cortical thickness is of general interest for downstream analysis, this metric is particularly important for longitudinal analysis. Low variance values indicate a smoothness in the cortical thickness across the meshes, which is desirable for these downstream applications. Note that the cortical thickness values also depend on the scale of the coordinate system used.

To compute the variance of cortical thickness, we need to distinguish between white matter and pial surfaces. Instead of computing the metrics separately for both meshes, we compute the metric only for the white matter vertices. The thickness at each white matter vertex is computed as the distance to the pial surface (point-to-face):

$$\text{thickness}_{i,j} = d(v_{i,j}, M_i^{\text{pial}}) \tag{3.4}$$

Next, the variances of these thickness values at each vertex across all meshes are calculated:

$$\text{ThVar}_j = \frac{1}{N_\text{M}} \sum_{i=1}^{N_\text{M}} (\text{thickness}_{i,j} - \overline{\text{thickness}_j})^2 \tag{3.5}$$

where $\overline{\text{thickness}_j}$ is the average thickness of the $j$-th vertex across all meshes ($\frac{1}{N_\text{M}} \sum_{i=1}^{N_\text{M}} \text{thickness}_{i,j}$).

### Mean Curvature Variance (MCVar)

The Mean Curvature Variance (MCVar) metric (cf. Subsection 2.1.2) provides a more geometric view of the mesh correspondence than the previous metrics. This metric can be particularly useful for evaluating the correspondence in the densely-folded areas of the cortex containing gyri and sulci, where there are large regional variations in curvature. High vertex correspondence in these regions is important as thickness measurements are very sensitive to small displacements in these regions. Like the cortical thickness, the mean curvature depends on the scale of the coordinate system used.

For the metric, we compute the discrete mean curvature at each vertex of each mesh using a combination of a cotangent Laplacian and the normal vector of each neighboring face of a vertex. Note that no smoothing is applied, but the median should reduce the effect of outliers, however. This also makes the computation more efficient. The exact algorithm used can be found in the source code of PyCortex 1.2.4 [14]. Let $H_{i,j}$ be the discrete mean curvature at the $j$-th vertex of the $i$-th mesh computed in this way. The variance of the discrete mean curvature is then computed as:

$$\mathrm{MCVar}_j = \frac{1}{N_\mathrm{M}} \sum_{i=1}^{N_\mathrm{M}} (H_{i,j} - \overline{H}_j)^2 \tag{3.6}$$

where $\overline{H}_j$ is the average mean curvature of the $j$-th vertex across all meshes ($\frac{1}{N_\mathrm{M}} \sum_{i=1}^{N_\mathrm{M}} H_{i,j}$).

### 3.2.2. Mean Vertex Distance ($\mathrm{V}_{\mathrm{L2}}$)

For the Mean Vertex Distance ($\mathrm{V}_{\mathrm{L2}}$) metric, we compute the Euclidean distance between each vertex and the centroid of all corresponding vertices in all meshes. The vertex distance is a very direct metric for evaluation of vertex correspondence. In particular, it is useful for evaluating reconstructions of the same brain anatomy, as is the case in the test-retest dataset (cf. Section 4.1), where the reconstructed meshes should be as similar as possible due to a short time interval between the scans.

To compute the metric, we first compute the distance between each vertex and the centroid of the corresponding vertices in all meshes:

$$\mathrm{VertDist}_{i,j} = \|v_{i,j} - \overline{v}_j\| \tag{3.7}$$

where $\overline{v}_j$ is the centroid of the $j$-th vertex across all meshes computed as the coordinate-wise average ($\frac{1}{N_\mathrm{M}} \sum_{i=1}^{N_\mathrm{M}} v_{i,j}$). Then, we take the average over all vertices and all meshes to get a single number per patient/set of meshes:

$$\mathrm{V}_{\mathrm{L2}} = \frac{1}{N_\mathrm{M} N_\mathrm{V}} \sum_{i=1}^{N_\mathrm{M}} \sum_{j=1}^{N_\mathrm{V}} \mathrm{VertDist}_{i,j} \tag{3.8}$$

Note that no median or variance is used in this metric.

### 3.2.3. Longitudinal Parcellation Consistency (ParcF1)

The Longitudinal Parcellation Consistency (ParcF1) is a metric that evaluates whether nearby vertices share the same parcellation label. This allows us to see how well the cortical regions overlap between the meshes.

To compute the metric, for each pair of meshes $M_i$ and $M_k$ where $i \neq k$, the parcellation label of each vertex in $M_i$ is compared to the parcellation label of the closest vertex in $M_k$. Based on these labels of the vertices in $M_i$ and their respective closest vertex in $M_k$, a weighted F1-score is computed (weighted by the number of instances of each label in the fsaverage parcellation). If a zero division occurs, e.g., if a label is never present in the closest vertices, that label's weight is set to zero for the averaging. The exact algorithm used to compute the F1-score can be found in the source code of scikit-learn 1.3.2 [84]. Of the $N_\mathrm{M} * (N_\mathrm{M} - 1)$ F1-scores, the median is taken as the final metric value.

Since in our case the meshes are derived from the fsaverage template, we can use the parcellation maps provided by FreeSurfer directly with the meshes (cf. Subsection 2.3.1). For this metric, we decided to use the Destrieux parcellation, because it is more fine-grained than the Desikan-Killiany parcellation (cf. Subsection 2.2.2).

# 4. Experimental Setup

In this chapter, we describe the setup used to evaluate different configurations of the V2C-Long model and to compare it with other methods. Figure A.1 in the appendix gives a detailed overview over the complete training procedure and evaluation pipeline.

## 4.1. Datasets

### 4.1.1. Overview

We use three different datasets to evaluate the V2C-Long model: the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset [36], the test-retest (TRT) dataset [55], and the Open Access Series of Imaging Studies (OASIS) dataset [40]. The ADNI dataset is used for training, validation, and testing, while the TRT and OASIS datasets are used as additional test sets to evaluate the generalization of the methods to unseen datasets. All these datasets are publicly available and widely used in related work. Table 4.1 gives an overview of the datasets used in this work and Figure 4.1 shows the distribution of patients in terms of number of scans per patient.

Due to the large number of scans in the ADNI dataset, we train and evaluate our model only on the pial and white matter surfaces of the right hemisphere. We expect the model to behave similarly on the left hemisphere.

| dataset | #patients | #scans | avg. #scans per pt.±std. | typical TBCS. | age range |
|---|---|---|---|---|---|
| ADNI | 1243 | 5433 | 4.37±1.88 | 6-18 months | 55 – 96 |
| TRT | 3 | 120 | 40.00±0.00 | <24 hours | 42 – 90 |
| OASIS | 100 | 288 | 2.88±1.11 | 1-3 years | 26 – 31 |

Table 4.1.: Overview of the datasets used in this work. TBCS = time between consecutive scans.

#### Alzheimer's Disease Neuroimaging Initiative (ADNI)

The Alzheimer's Disease Neuroimaging Initiative is a large, longitudinal study of elderly patients, including healthy subjects, subjects diagnosed with Mild Cognitive Impairment (MCI), and subjects diagnosed with AD [36]. The ADNI dataset is the most commonly used dataset in the deep CSR methods mentioned in this work [41, 103, 53, 5, 6, 17, 30, 77, 16] and contains longitudinal MRI scans with time information given in multiples of 6 months since the first scan.

After removal of failed FreeSurfer reconstructions (see Subsection 4.1.2), our dataset consists of 1243 patients with a total of 5433 scans. The lowest and highest number of time points per patient are 2 and 13 respectively, with an average of 4.37 time points per patient (standard deviation: 1.88 time points).

### Open Access Series of Imaging Studies-3 (OASIS)

The OASIS-3 dataset is another open access dataset of longitudinal MRI scans and is part of the OASIS project [40, 58, 57]. It contains MRI scans of both cognitively normal subjects and subjects diagnosed with AD. The timing information is less precise than in the ADNI dataset, as only the patient's age in years at the time of the scan is given.

We use the OASIS dataset as an additional way to test the robustness and generalization of the V2C-Long model to unseen datasets. Since we use the OASIS dataset only for external evaluation, we have reduced the dataset size by randomly selecting 100 patients, resulting in a total of 288 scans. The ages of these patients at the time of their scans ranges from 42 to 90 years.

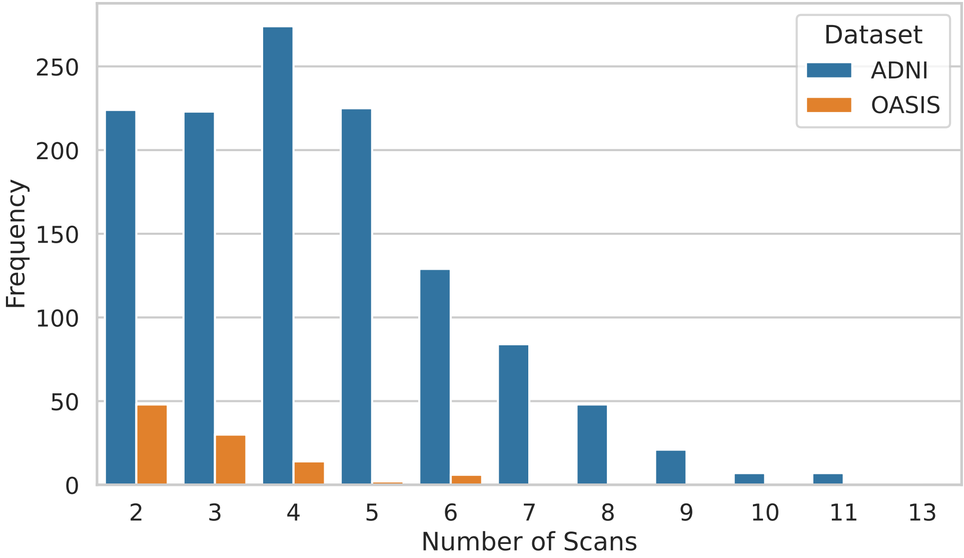


Figure 4.1.: Distribution of total scans per patient in the ADNI and OASIS datasets.

### Test-Retest (TRT)

The TRT dataset by Maclaren et al. [55] consists of three subjects with 40 scans each. The scans were collected by scanning each subject twice within 24 hours for 20 consecutive days.

Given that the morphological changes in the brain are expected to be minimal in this time frame, the TRT dataset provides a stable benchmark for testing the consistency of CSR methods. In the context of mesh correspondence, the TRT dataset allows the evaluation of correspondence metrics (cf. Subsection 4.6.2) without measuring the variance introduced by morphological changes in the brain. As seen in Table 4.1, the subjects are relatively young in comparison to the subjects in the ADNI and OASIS dataset, which could make the reconstruction for these patients harder, since the models are trained on ADNI data.

### 4.1.2. Preprocessing

For each dataset, the following preprocessing procedure is performed:

1. The officially provided raw files are processed using FreeSurfer v7.2 and its `recon-all` pipeline (cf. Subsection 2.3.1).
2. MRI images for which FreeSurfer processing failed are removed.
3. The images (`orig.mgz`), meshes (`rh.pial`, `rh.white`), and segmentation maps (`aseg.mgz`) are registered to MNI152 space (cf. Subsection 2.1.3) with an affine transformation generated by NiftyReg [62] (more precisely `reg_aladin` v1.5.76), which registers the respective MRI image to the MNI152 template image.

For training and evaluation of related methods (cf. Section 4.5), the following additional preprocessing steps are performed:

- All ADNI meshes are resampled to fsaverage as described in [77]. The meshes are transformed to MNI space with the same transformation matrix as the `orig.mgz` file (see above).
- All scans are run through FreeSurfer's longitudinal pipeline (cf. Subsection 2.3.1). The resulting meshes are transformed to MNI space using the same transformation as the `orig.mgz` file from the cross-sectional pipeline (see above).

Finally, to train the Template Generation Model (TGM) and the V2C-Long, the same additional preprocessing steps are performed as described in [5]. Namely, the ground truth meshes were reduced to about 40,000 vertices with quadratic edge collapse decimation [29] using MeshLab [12]. The MRI images were cropped to size (192,224,192) and intensity normalized between 0 and 1 using min-max normalization.

### 4.1.3. Training/Validation/Test Split

The main dataset used for training is the ADNI dataset. We split the ADNI dataset into a training set, a validation set, and a test set using the same 70/10/20 split as in [6] and [77]. Importantly, we have split the longitudinal dataset at the patient level and balance the splits with respect to patient age, sex, and diagnosis.

## 4.2. General Training Setup

All V2C-Flow models trained in this work (including the TGM and V2C-Long models) are trained on the ADNI training set with the following setup:

- **Model**: The model has two components: a 3D U-Net operating in image space and a graph network operating in mesh space. We use the same configuration as Bongratz, Rickmann, and Wachinger [6], i.e. the graph network consists of two graph Neural Ordinary Differential Equation (NODE) blocks with 5 Euler steps each.

- **Ground Truth**: We use the FreeSurfer surfaces and segmentations derived from the cross-sectional `recon-all` pipeline. The ground truths are registered to MNI space as described in Subsection 4.1.2. For training purposes, we further reduce the ground truth meshes with quadratic edge collapse decimation [29] to about 40,000 vertices per mesh to save memory.

- **Loss**: The training loss is a combination of a cross-entropy loss for the U-Net, a modified, curvature-weighted, Chamfer loss for the graph, and an edge and normal consistency regularization loss for the graph (see [6] for details). For the modified Chamfer loss, we sample 100,000 points on the reduced ground truth meshes and resample them if the validation Average Symmetric Surface Distance (ASSD) has not improved for 20 epochs. The gradients are clipped to a maximum norm of 200,000.

- **Optimizer**: The models are optimized using the AdamW [51] optimizer with $\beta_1 = 0.9, \beta_2 = 0.999$, a weight decay of $10^{-4}$, a batch size of 1 and a cyclic learning rate schedule [82]. The base learning rate is $10^{-4}$ for the U-Net weights and $5 \cdot 10^{-5}$ for the graph parameters.

- **Validation**: Unless otherwise specified, the model was evaluated every 5 epochs on the ADNI validation set. When referring to the "best" epoch, we refer to the epoch with the lowest mean ASSD (over all surfaces and patients) on the ADNI validation set (this does not include aggregations in the NXN/NXN_sorted modes).

- **Surfaces**: Due to the large number of scans in the longitudinal ADNI dataset, and due to the linear scaling of GPU and system memory requirements with the number of surfaces, we train and evaluate our model only on the pial and white matter surfaces of the right hemisphere.

## 4.3. Template Generation

For template generation, we pretrain a single model on the ADNI training set for 55 epochs using the smaller, smoothed *fsaverage6* template, which contains around 40,000 vertices per surface. We evaluate the model every 5 epochs for the first 45 epochs and every epoch for the last 10 epochs. The best validation mean ASSD score was obtained after 46 epochs.

The same model is then trained on the larger, smoothed fsaverage template for 15 epochs. The epoch with the best validation score was epoch 7.

Finally, the model is used to predict all surfaces in the full ADNI dataset (training+validation+test), the OASIS dataset, and the TRT dataset. For the MEAN and MEDIAN pairing modes, a patient-specific template is generated by averaging the predicted surfaces of a patient.

## 4.4. Ablation Studies

To find the best configuration of pairing mode and longitudinal template features, we perform two ablation studies on the ADNI validation set. In the first ablation study, we compare the performance of the pairing modes (cf. Subsection 3.1.2) by training and evaluating a model for each. In the second ablation study, we measure the effect of different longitudinal template features (cf. Subsection 3.1.3), by training and evaluating a model for each combination of features.

### 4.4.1. Pairing Mode

For the pairing mode ablation study, we train a separate model for each pairing mode (except the PREV_CHAIN mode) on the ADNI training set. In addition, we train a model for the static, smooth fsaverage template used in the original V2C-Flow paper [5]. We call this model the STATIC mode model. All models use the ADNI templates generated by the trained TGM described in Section 4.3. To save training time, we initialize each model with the weights from the trained TGM. After training, we evaluate each model on the ADNI validation set and select the best model based on the reconstruction and correspondence metrics (cf. Subsection 4.6.2 and Section 3.2).

Since the number of generated pairings, and thus the number of steps per epoch, varies between the different pairing modes (cf. Table 3.1)), we train each model for roughly the same number of steps (about 57,000) instead of epochs. To additionally evaluate the models at similar points in training, we also vary the number of epochs between each evaluation. An overview of the concrete number of steps and epochs used for each model can be found in Table 4.2.

| Pairing Mode | #Steps per Epoch | #Epochs | Total #Steps | Eval. every N epochs |
|---|---:|---:|---:|---:|
| PREV | 2,880 | 20 | 57,600 | 5 |
| NxN | 19,073 | 3 | 57,219 | 1 |
| NxN_SORTED | 11,409 | 5 | 57,045 | 2 |
| All other modes | 3,745 | 15 | 56,175 | 5 |

Table 4.2.: Overview of steps and epochs used for each model in the pairing mode ablation study.

### 4.4.2. Template Features

For the template features ablation study, a separate model is trained for each combination of template features (cf. Subsection 3.1.3) on the ADNI training set. This results in a total of 4 models: One model trained without longitudinal template features, one model trained with additional time difference features, one model trained with graph features from the TGM, and one model trained with both. Since the number of features varies between the different combinations, we cannot reuse existing model weights for initialization, and train each model from scratch instead. We train each model for 45 epochs (about 170,000 steps) and evaluate each model every 5 epochs on the ADNI validation set. In the last five epochs, we evaluate the models after each epoch.

Finally, we select the final configuration of the V2C-Long model based on the reconstruction and correspondence metrics.

## 4.5. Comparison with Related Methods

After selecting the final configuration of the V2C-Long model, we evaluate it on the ADNI test set, the OASIS dataset, and the TRT dataset. We also evaluate the following related methods on the same datasets.

### 4.5.1. FreeSurfer Longitudinal

Since FreeSurfer is the de facto standard method for CSR [5], it is an important baseline for comparison. As described in Subsection 4.1.2, the FreeSurfer longitudinal pipeline (cf. Subsection 2.3.1) is run on all scans in our ADNI test set, the OASIS dataset, and the TRT dataset. We then use the same transformations used the normal FreeSurfer ground truth meshes to transform the FreeSurfer longitudinal meshes (right hemisphere, pial and white) to MNI space. As the ground truth meshes used to evaluate the reconstruction metrics come from FreeSurfer itself, we expect good results in terms of reconstruction accuracy.

### 4.5.2. V2C-Flow

Since the V2C-Long model is based on V2C-Flow [5], we also include it in our evaluation. Note that the general setup of this model is the same as the STATIC mode model used in the pairing mode ablation study (see Subsection 4.4.1), i.e., we use the smoothed fsaverage template. The hyperparameters used for training are also the same as those used in the V2C-Long model, as described in Section 4.2. However, instead of initializing the model with the weights from the trained TGM, we train the model from scratch for 45 epochs on the ADNI training set. In this way, we can fairly compare the performance with the best V2C-Long model from the template features ablation study. The model was evaluated every 5 epochs on the ADNI validation set. For the last 5 epochs, the model was evaluated every epoch. For the final evaluation on the ADNI test set, the OASIS dataset, and the TRT dataset, we selected the epoch with the best mean ASSD score on the validation set.

### 4.5.3. CorticalFlow++

CorticalFlow++ [17, 42] is another explicit mesh deformation method for Cortical Surface Reconstruction. It learns a diffeomorphic deformation field which gives theoretical guarantees on the smoothness and invertibility of the deformation and further implies no self-intersections in the reconstructed surfaces [42]. However, due to discretization and numerical integration errors, self-intersections still occur in practice, although at a much lower rate than in Vox2Cortex or V2CC [77].

For our evaluation, we use the publicly available source code of CorticalFlow++ [83], based on Git commit `550953e6b07`. For the Chamfer loss, we sample 100,000 points on the ground truth mesh. The template mesh used has about 140,000 vertices per surface, providing a similar level of detail as the meshes in our V2CC, V2C-Flow, and V2C-Long models, which have about 160,000 vertices each. The three deformation blocks of the model were trained on the ADNI training set for 70,000 iterations each. This results in a total of 210,000 iterations, which is more than the 170,000 iterations used for the V2C-Long model.

### 4.5.4. V2CC

The final comparison method is V2CC [77]. See Subsection 2.3.3 for a description and visual overview of the method.

The mesh loss used to train V2CC is the one proposed by Rickmann, Bongratz, and Wachinger [77]. With $M_y$ being the predicted mesh and $M_{y'}$ being the resampled ground truth mesh, the mesh loss is defined as follows (adapted from [77]):

$$\mathcal{L}_{\text{Mesh}}(M_y, M'_y) = \mathcal{L}_1(M_y, M'_y) + \lambda \mathcal{L}_{\text{reg}}(M_y) \tag{4.1}$$

where $\mathcal{L}_1$ is the $L^1$ norm of the mean absolute error between the corresponding vertex coordinates of the predicted mesh and the resampled ground truth mesh, and $\mathcal{L}_{\text{reg}}$ is the normal consistency loss used in V2C-Flow and Vox2Cortex [5, 6]. Additionally, the cross entropy-loss for the U-Net segmentation is added 1:1 to the mesh loss for the final loss. For our training, we set the parameter $\lambda = 0.003$ for the white matter surface and $\lambda = 0.007$ for the pial surface.

As described in Subsection 4.1.2, we resampled the ADNI ground truth meshes to fsaverage and transformed them to MNI space, using the same transformation matrix as for the non-resampled ground truth meshes. Note that unlike V2C-Long, we did not reduce the meshes for the Chamfer loss. We then trained V2CC for 45 epochs on the ADNI training set and evaluated it every 5 epochs on the ADNI validation set. The epoch for the evaluation on the test set and the other datasets was chosen based on the best average ASSD score on the validation set.

## 4.6. Evaluation Procedure

### 4.6.1. Overview

**Evaluation Metrics**

We evaluate all methods in two ways: First, we evaluate the reconstructed surfaces (right hemisphere, pial and white matter) against the ground truth surfaces for all predictions using the reconstruction metrics described Subsection 4.6.2. Second, we evaluate the correspondence of the reconstructed surfaces by applying the metrics described in Section 3.2 to all sets of intra-patient predicted meshes.

This results in a set of values for each scan and surface type for the reconstruction metrics and a set of values for each patient and surface type (white and pial) for the correspondence metrics. For the reconstruction metrics, we first mean the values over all time points of a patient to get a single value per patient and surface type. Finally, we compute the mean and standard deviation of the values over all patients for each surface type and metric.

**Vertex Displacement**

In our V2C-Long model, we additionally track the total distance that each vertex is displaced during a prediction, by summing the vertex displacement distance of each deformation block (in our case, there are two blocks). This metric does not directly measure the quality of the vertex correspondence or the reconstruction accuracy, but it can help in understanding the behavior of the model.

**Datasets Used**

For the ablation studies, we evaluate the relevant models on the ADNI validation set. For the final comparison of the best V2C-Long model with other approaches, we evaluate the relevant models on the ADNI test set, the OASIS dataset, and the TRT dataset.

### 4.6.2. Reconstruction Metrics

In addition to evaluating the correspondence of the reconstructed surfaces with the metrics described in Section 3.2, we also evaluate the accuracy and regularity of the reconstructed surfaces.

As in Bongratz, Rickmann, and Wachinger [6], we use the ASSD and the 90th Percentile Hausdorff Distance ($\mathrm{HD}_{90}$) to evaluate the quality of the reconstruction. To evaluate the regularity of the surfaces, we use the average percentage of self-intersecting faces.

#### Average Symmetric Surface Distance (ASSD)

The Average Symmetric Surface Distance (ASSD) is a metric that measures the distance between two sets. In our case, we define it as the average of the distances from each point in one set to the nearest point in the other set, and vice versa:

$$\mathrm{ASSD}(A, B) = \frac{\sum_{a \in A} d(a, B) + \sum_{b \in B} d(b, A)}{|A| + |B|} \tag{4.2}$$

where $A$ and $B$ are the two sets and $d(\cdot, \cdot)$ is the Euclidean distance from a point to the nearest point in the other set.

For our purposes, $A$ and $B$ are both sets of 100,000 uniformly sampled points on the ground truth and reconstructed surface respectively. These points are resampled for each evaluation pair.

#### 90th Percentile Hausdorff Distance ($\mathrm{HD}_{90}$)

Similar to the ASSD, the Hausdorff Distance is a measure of the distance between two sets. The $\mathrm{HD}_{90}$ is a modified version of the Hausdorff Distance that is less sensitive to outliers. While the Hausdorff Distance is based on the maximum distance from a point in one set to the nearest point in the other set, the $\mathrm{HD}_{90}$ considers the 90th percentile of these per-point distances. For symmetry, we use the maximum of the 90th percentile of the distances from $A$ to $B$ and from $B$ to $A$:

$$\mathrm{HD}_{90}(A, B) = \max\big(\mathrm{Q}_{90}(\{d(a, B) \mid a \in A\}), \mathrm{Q}_{90}(\{d(b, A) \mid b \in B\})\big) \tag{4.3}$$

where $\mathrm{Q}_{90}$ is the 90th percentile of one set and $d$ is again the Euclidean distance from a point to the nearest point in the other set.

As in the case of the ASSD, $A$ and $B$ are both sets of 100,000 uniformly sampled points each on the ground truth and the reconstructed surface, respectively. We use the same points to compute both the ASSD and the $\mathrm{HD}_{90}$.

#### Mean Percentage of Self-Intersecting Faces (%SIF)

The Percentage of Self-intersecting Faces (%SIF) is a measure of the regularity of the reconstructed surfaces.

Mesh deformation-based methods such as Vox2Cortex tend to produce self-intersecting surfaces [5], which can cause issues in downstream processing steps. To evaluate the number of self-intersections, we use the %SIF metric, which is defined as the mean number of self-intersections per 100 faces.

To compute the %SIF, we use MeshLab, an open source mesh processing tool [13], with the `pymeshlab` [63] Python bindings.

### 4.6.3. Evaluation of Pairing Modes

The number of meshes generated and the number of time points reconstructed vary between the different pairing modes. This leads to different evaluation procedures for the different pairing modes:

- **STATIC/FIRST/MEAN/MEDIAN**: For these modes, we get exactly one prediction for each patient and each time point in the validation set. We compute the metrics as described in Subsection 4.6.1.

- **PREV/PREV_CHAIN**: The PREV_CHAIN mode is not trained directly, but is another way to predict meshes using the PREV mode model. The PREV_CHAIN prediction mode starts with the template generated by the Template Generation Model (TGM) for a patient's first time point and then uses the PREV model to predict the second time point. For the third time point and onwards, the previous prediction is used as input, unlike the PREV model, which would use a mesh from the TGM. This is repeated until all time points have been predicted. Figure 4.2 illustrates this prediction mode. As each prediction should have a good vertex correspondence to its template, i.e. the previous prediction, we expect a better overall intra-patient mesh correspondence than in the PREV mode, where each prediction uses a different, more independent template.

  Both modes do not generate predictions for the first time point of a patient. The reconstruction metrics are computed only for the actual predicted time points. This means that we average the reconstruction metrics from the second up to the last time point to get a value per patient, which is then averaged over all patients. For the correspondence metrics, however, we substitute the first time point with the mesh generated by the TGM for fair comparison with the other modes, as all model evaluations should use the same "total scans per patient" distribution. The metric is then computed over all time points for each patient.

- **NXN/NXN_sorted**: For these modes, we get more predictions per patient than there are time points. In addition, we get mean and median aggregates of each time point. We first evaluate the reconstruction metrics for each prediction in the patient (without the aggregates) and average them to get a score per patient. For the correspondence metrics, we evaluate the correspondence across *all* predictions for the patient, which means that this type of evaluation uses significantly more meshes than the evaluation of other modes. We denote the these evaluations as NXN/NXN_sorted in the results.

In addition, we evaluate the metrics for the aggregated meshes separately. For the aggregates, there is only one final mesh per patient and time point, so we evaluate the metrics as in the STATIC/FIRST/MEAN/MEDIAN case. These evaluations are denoted in the results as NxN mean, NxN median, NxN_sorted mean, and NxN_sorted median.

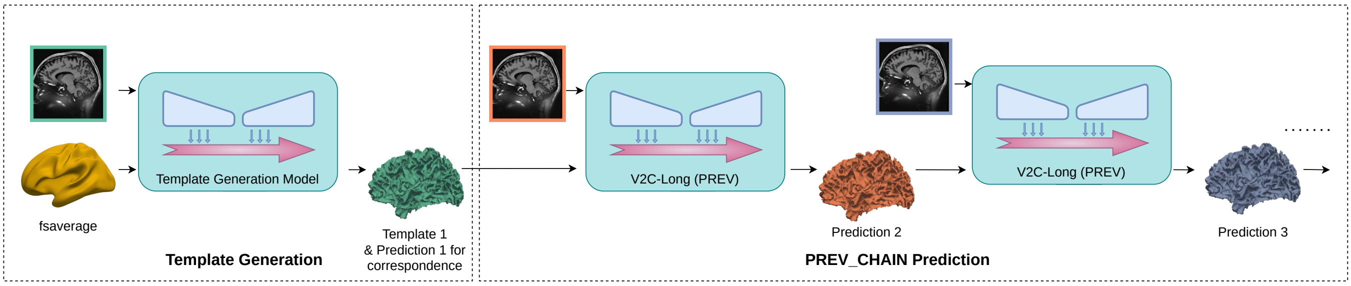


Figure 4.2.: Illustration of the PREV_CHAIN prediction mode. The model uses the previously predicted mesh as input for the next prediction. The first prediction is generated by the TGM.

## 4.7. Implementation Details

All models were trained on an NVIDIA A100 GPU with 40GB of VRAM. We used the PyTorch [67] deep learning framework and the PyTorch3D [70] library along with CUDA 11.3 to implement and run the models. In addition, we used CUDA's automatic mixed precision (AMP) [2] to reduce the memory consumption of the models.

As the training set contains 3745 scans, and the NxN and NxN_sorted pairing modes additionally generate a large number of training items, the total training for a single model could take up to several weeks and require up to 837GB of system memory. The Table A.1 in the appendix gives an overview over the hardware resources used and the training time of the models.

# 5. Results and Discussion

## 5.1. Template Generation

| | | **white surface** | | | **pial surface** | | |
|---|---|---|---|---|---|---|---|
| | **Type** | **ASSD**±std↓ | **HD$_{90}$**±std↓ | **%SIF**±std↓ | **ASSD**±std↓ | **HD$_{90}$**±std↓ | **%SIF**±std↓ |
| ADNI | regular | 0.192±0.120 | 0.424±0.617 | 0.901±0.316 | 0.191±0.110 | 0.433±0.588 | 2.465±0.974 |
| | mean | – | – | 0.997±0.383 | – | – | 2.406±0.875 |
| | median | – | – | 0.454±0.173 | – | – | 1.264±0.627 |
| OASIS | regular | 0.196±0.037 | 0.420±0.079 | 0.965±0.272 | 0.203±0.025 | 0.454±0.067 | 3.011±1.206 |
| | mean | – | – | 1.242±0.419 | – | – | 3.047±0.990 |
| | median | – | – | 0.478±0.175 | – | – | 1.543±0.676 |
| TRT | regular | 0.187±0.014 | 0.418±0.033 | 1.051±0.409 | 0.275±0.035 | 0.617±0.091 | 4.627±0.616 |
| | mean | – | – | 0.712±0.212 | – | – | 2.940±0.599 |
| | median | – | – | 0.329±0.068 | – | – | 1.844±0.473 |

Table 5.1.: Reconstruction results for the white and pial surfaces for the generated templates by the TGM. ASSD and HD$_{90}$ are ommited for mean/median.

The above table shows the reconstruction metrics for the templates generated by the Template Generation Model (TGM), including the patient-wise mean and median templates. Note that for ADNI, the metrics are evaluated over the complete dataset, including the training, validation, and test sets.

Compared to the results of related work [6, 5], the metrics show a very good reconstruction accuracy. On ADNI, the ASSD and HD$_{90}$ metrics are in a similar range for both surfaces. On other datasets, the white surface can be reconstructed better than the pial surface, both in terms of accuracy (ASSD, HD$_{90}$) and regularity (%SIF), than the pial surface. This may be explained by the fact that white surface is typically less complex than the pial surface, which is more folded, making it harder to generalize.

The metrics align with the observations in [6, Table 2], that V2C-Flow is able to generalize well to unseen datasets. In particular on the white surface, the ASSD and HD$_{90}$ scores for both datasets are very close to the scores obtained on ADNI, although the model is trained exclusively on the latter. On the pial surface, the scores are significantly worse for the TRT dataset, which could possibly be caused by the young age of the patients (cf. Table 4.1).

Interestingly, the patient-wise median templates roughly halve the number of self-intersections on both surfaces in ADNI and OASIS and even reduce the number of self-intersections by about two-thirds on the TRT dataset. This could be due to the removal of outliers, which can cause irregularities on the mesh, and the general smoothing property of the median.

## 5.2. Ablation Study: Pairing Mode

The first ablation study compares the different pairing modes (cf. Subsection 3.1.2) on the ADNI validation set. The metrics are evaluated for the best epoch for each model in terms of mean ASSD (over all patients and pial/white surfaces). The best epoch is 15 for the STATIC model, 10 for the FIRST model, 10 for the MEAN model, 15 for the MEDIAN model, 20 for the PREV model, 1 for the NxN model, and 4 for the NxN_SORTED model.

### Reconstruction Accuracy and Self-Intersections

Table 5.2a shows the reconstruction metrics for the different pairing modes. The same trends can be observed as for the template metrics: Pial surface reconstruction is more difficult than white surface reconstruction, and median aggregation of predictions significantly reduces the number of self-intersections. As seen in Figure 5.1 when looking at the PREV_CHAIN mode, the self-intersections from the templates are propagated and amplified in the predictions, which explains why the MEDIAN mode model (trained with the patient-specific MEDIAN templates) has fewer intersections than most other models and the PREV_CHAIN model has the most intersections. Excluding the models related to median operations, V2C-Long models typically have 20-40% more self-intersections than the V2C-Flow model with the fsaverage template (STATIC), due to this amplification.

Both the mean and median post-prediction aggregations for the NxN/NxN_SORTED models perform significantly worse than the other models when it comes to the reconstruction accuracy metrics (ASSD, $HD_{90}$). The other models all reach similar scores for ASSD and $HD_{90}$ for both surfaces, with differences between the values of less than 0.025 mm for both metrics.

### Mesh Correspondence

Tables 5.2b and 5.2c show the mesh correspondence metrics for the different pairing modes. Additionally, the mean total vertex displacement (in mm) is shown for each model.

First, we observe that the correspondence metrics correlate well with each other. A good value in one metric typically indicates a good value in the other metrics, although the relationship is not linear. In addition, the metrics do not differ much between the pial and white surfaces, except for the Mean Curvature Variance (MCVar). This can be explained by the fact that the pial surface is generally more curved than the white surface, resulting in higher median values and thus higher variances.

In most cases, the V2C-Long models outperform V2C-Flow (STATIC) in the correspondence metrics. In pairing modes, where intra-patient predictions share the same template (FIRST, MEAN, MEDIAN), the correspondences are significantly and consistently better than in V2C-Flow. This is expected because the V2C-Long deformations are applied to the same vertices for all time point predictions and the displacements

| | white surface | | | pial surface | | |
|---|---|---|---|---|---|---|
| **Pairing Mode** | **ASSD**±std↓ | **HD$_{90}$**±std↓ | **%SIF**±std↓ | **ASSD**±std↓ | **HD$_{90}$**±std↓ | **%SIF**±std↓ |
| STATIC | 0.1793±0.0305 | 0.4008±0.0745 | 1.042±0.260 | 0.1738±0.0242 | 0.3805±0.0562 | 2.004±0.678 |
| FIRST | 0.1813±0.0274 | 0.3883±0.0610 | 1.149±0.322 | 0.1671±0.0206 | 0.3822±0.0546 | 3.095±1.174 |
| MEAN | 0.1810±0.0303 | 0.4101±0.0752 | 1.286±0.357 | **0.1667**±0.0218 | **0.3727**±0.0528 | 2.842±1.109 |
| MEDIAN | 0.1788±0.0275 | 0.4036±0.0695 | **0.701**±0.225 | **0.1666**±0.0202 | 0.3834±0.0522 | 2.681±1.112 |
| PREV | **0.1783**±0.0281 | 0.3886±0.0653 | 1.148±0.313 | 0.1687±0.0211 | 0.3830±0.0523 | 2.588±0.943 |
| PREV_CHAIN | 0.1800±0.0284 | 0.3924±0.0666 | 1.945±1.090 | 0.1708±0.0214 | 0.3837±0.0520 | 3.327±1.236 |
| NxN | 0.1798±0.0266 | **0.3873**±0.0616 | 1.223±0.313 | 0.1724±0.0207 | 0.3980±0.0544 | 3.099±1.184 |
| NxN_SO. | **0.1768**±0.0267 | **0.3877**±0.0647 | 1.233±0.322 | 0.1689±0.0213 | **0.3804**±0.0526 | 2.753±1.042 |
| NxN mean | 0.2426±0.0437 | 0.5527±0.1129 | 1.179±0.325 | 0.2404±0.0441 | 0.5773±0.1131 | 3.016±1.096 |
| NxN_SO. mean | 0.2442±0.0688 | 0.5673±0.1872 | 1.233±0.335 | 0.2390±0.0677 | 0.5688±0.1831 | 2.768±1.003 |
| NxN median | 0.2450±0.0384 | 0.5789±0.1097 | **0.578**±0.229 | 0.2379±0.0357 | 0.5982±0.1050 | **1.671**±0.787 |
| NxN_SO. median | 0.2199±0.0448 | 0.5182±0.1305 | 0.800±0.333 | 0.2146±0.0432 | 0.5183±0.1233 | **1.896**±0.974 |

(a) Reconstruction results for the pial and white surfaces.

| | white surface | | | | | both |
|---|---|---|---|---|---|---|
| **Pairing Mode** | **EdgeVar**±std↓ | **MCVar**±std↓ | **V$_{L2}$**±std↓ | **ParcF1**±std↑ | **Disp**±std↓ | **ThVar**±std↓ |
| STATIC | 0.0115±0.0051 | 0.0546±0.0233 | 0.9385±0.2221 | 0.9263±0.0139 | 7.8585±0.3472 | 0.0487±0.0198 |
| FIRST | 0.0027±0.0012 | 0.0222±0.0094 | 0.4120±0.1019 | 0.9699±0.0079 | 1.0346±0.2512 | 0.0255±0.0114 |
| MEAN | 0.0028±0.0012 | 0.0268±0.0117 | 0.4047±0.1042 | 0.9710±0.0081 | 0.9124±0.1353 | 0.0255±0.0111 |
| MEDIAN | **0.0021**±0.0010 | **0.0185**±0.0078 | **0.4018**±0.1001 | **0.9719**±0.0079 | **0.8955**±0.1311 | **0.0242**±0.0108 |
| PREV | 0.0115±0.0052 | 0.0475±0.0213 | 0.9806±0.2213 | 0.9221±0.0145 | 1.0895±0.2051 | 0.0460±0.0196 |
| PREV_CHAIN | 0.0062±0.0037 | 0.0341±0.0188 | 0.7164±0.2251 | 0.9401±0.0148 | 1.0895±0.2051 | 0.0361±0.0155 |
| NxN | 0.0155±0.0060 | 0.0673±0.0239 | 0.9383±0.2225 | 0.9318±0.0126 | 1.0852±0.1168 | 0.0463±0.0188 |
| NxN_SO. | 0.0133±0.0054 | 0.0613±0.0236 | 0.8943±0.2214 | 0.9326±0.0125 | **0.9056**±0.1773 | 0.0433±0.0179 |
| NxN mean | 0.0044±0.0022 | 0.0508±0.0287 | 0.4040±0.0993 | 0.9713±0.0076 | – | 0.0281±0.0118 |
| NxN_SO. mean | 0.0072±0.0033 | 0.0626±0.0311 | 0.6165±0.1377 | 0.9509±0.0111 | – | 0.0404±0.0154 |
| NxN median | **0.0021**±0.0008 | **0.0118**±0.0037 | **0.3864**±0.0937 | **0.9738**±0.0069 | – | **0.0227**±0.0095 |
| NxN_SO. median | 0.0042±0.0015 | 0.0221±0.0070 | 0.5344±0.1145 | 0.9582±0.0085 | – | 0.0310±0.0121 |

(b) Correspondence results for the white surface and the results for the thickness variance.

| | pial surface | | | | |
|---|---|---|---|---|---|
| **Pairing Mode** | **EdgeVar**±std↓ | **MCVar**±std↓ | **V$_{L2}$**±std↓ | **ParcF1**±std↑ | **Disp**±std↓ |
| STATIC | 0.0113±0.0051 | 0.0384±0.0166 | 0.9760±0.2301 | 0.9231±0.0145 | 8.6509±0.4410 |
| FIRST | 0.0024±0.0011 | 0.0141±0.0059 | 0.4069±0.1025 | 0.9628±0.0111 | 1.0491±0.2579 |
| MEAN | 0.0023±0.0010 | 0.0168±0.0072 | **0.3943**±0.1040 | **0.9662**±0.0108 | 0.9177±0.1344 |
| MEDIAN | **0.0019**±0.0009 | **0.0116**±0.0049 | 0.3957±0.1001 | 0.9651±0.0107 | **0.8530**±0.1278 |
| PREV | 0.0112±0.0051 | 0.0345±0.0156 | 1.0172±0.2292 | 0.9155±0.0164 | 1.0944±0.2101 |
| PREV_CHAIN | 0.0058±0.0035 | 0.0226±0.0123 | 0.7167±0.2283 | 0.9358±0.0160 | 1.0944±0.2101 |
| NxN | 0.0149±0.0059 | 0.0500±0.0184 | 0.9759±0.2311 | 0.9253±0.0148 | 1.0989±0.1155 |
| NxN_SO. | 0.0127±0.0052 | 0.0440±0.0171 | 0.9299±0.2293 | 0.9279±0.0138 | **0.9122**±0.1803 |
| NxN mean | 0.0039±0.0019 | 0.0368±0.0215 | 0.3994±0.0994 | 0.9646±0.0103 | – |
| NxN_SO. mean | 0.0068±0.0031 | 0.0464±0.0235 | 0.6313±0.1416 | 0.9434±0.0126 | – |
| NxN median | **0.0019**±0.0007 | **0.0075**±0.0023 | **0.3816**±0.0940 | **0.9681**±0.0091 | – |
| NxN_SO. median | 0.0039±0.0014 | 0.0149±0.0047 | 0.5434±0.1164 | 0.9523±0.0100 | – |

(c) Correspondence results the pial surface.

Table 5.2.: Ablation study results for different pairing modes for both reconstruction and correspondence metrics on the ADNI validation set. The two best results for each metric are highlighted. For a description of each metric, see Subsection 4.6.2 and Section 3.2.

are relatively small because the template is already close to the target mesh, which is a natural limit for most correspondence metrics. On the other hand, pairing modes where the template is not shared (PREV, NxN, NxN_SORTED), have correspondence metrics in a similar range to the V2C-Flow model (STATIC). This can be explained by the fact that the templates generated by the TGM (which contains a V2C-Flow architecture) are likely to have similar mesh correspondence as the meshes from the STATIC model, and the displacements in the V2C-Long model are comparatively small and do not significantly affect the mesh correspondence (and would rather add noise/variance to the vertex positions than remove it).

The PREV_CHAIN model performs better than the PREV mode, probably because of the continuous vertex displacement across all time steps. However, when looking at Figure 5.1, it is clear that the number of self-intersections is amplified over time. In addition, the correspondence metrics get worse faster for more scans than for other modes (see Figure 5.2), indicating that each additional prediction amplifies the noise in the predictions for the PREV_CHAIN mode. This also explains why the correspondence metrics are worse than for the FIRST/MEAN/MEDIAN modes, where such a noise amplification does not occur.

The NxN mean model performs similarly to the patient-specific template models (FIRST, MEAN, MEDIAN) in the more position-related metrics ($V_{L2}$, ParcF1, Thickness), but worse in the EdgeVar and MCVar metrics. The NxN median aggregation model performs best in most metrics compared to all other models. The good results in both cases may be caused due to the inclusion of multiple meshes per time point, which may reduce the overall variance in the set of meshes that are evaluated with the correspondence metrics. In both mean/median versions, the NxN_SORTED model performs worse than the NxN model, which may be related to the imbalance in the number of predictions per time point, which leads to biases in the mean/median computation (cf. Figure 3.3).

### 5.2.1. Bias

**Reconstruction Accuracy by Time Point**

Since some pairing modes (PREV, FIRST, NxN_sorted) have a bias towards certain processing directions or a certain time point (compare with [76, 74]), and modes like MEAN/MEDIAN might favor certain time points closer to the middle of the time series, we examine the effect of the time point on the reconstruction metrics. Figure 5.1 shows the mean reconstruction metrics grouped by prediction time point for patients with exactly 5 total scans (18 patients).

The first thing to note is that the results are very similar for the STATIC, FIRST, MEDIAN, MEAN, PREV, NxN, NxN mean models for all time points, both in absolute terms and in terms of the relative differences between the models. For the NxN median/NxN_SORTED, NxN_SORTED mean/medians, the metrics vary significantly between time points. This could be caused by increased sensitivity to outliers, which

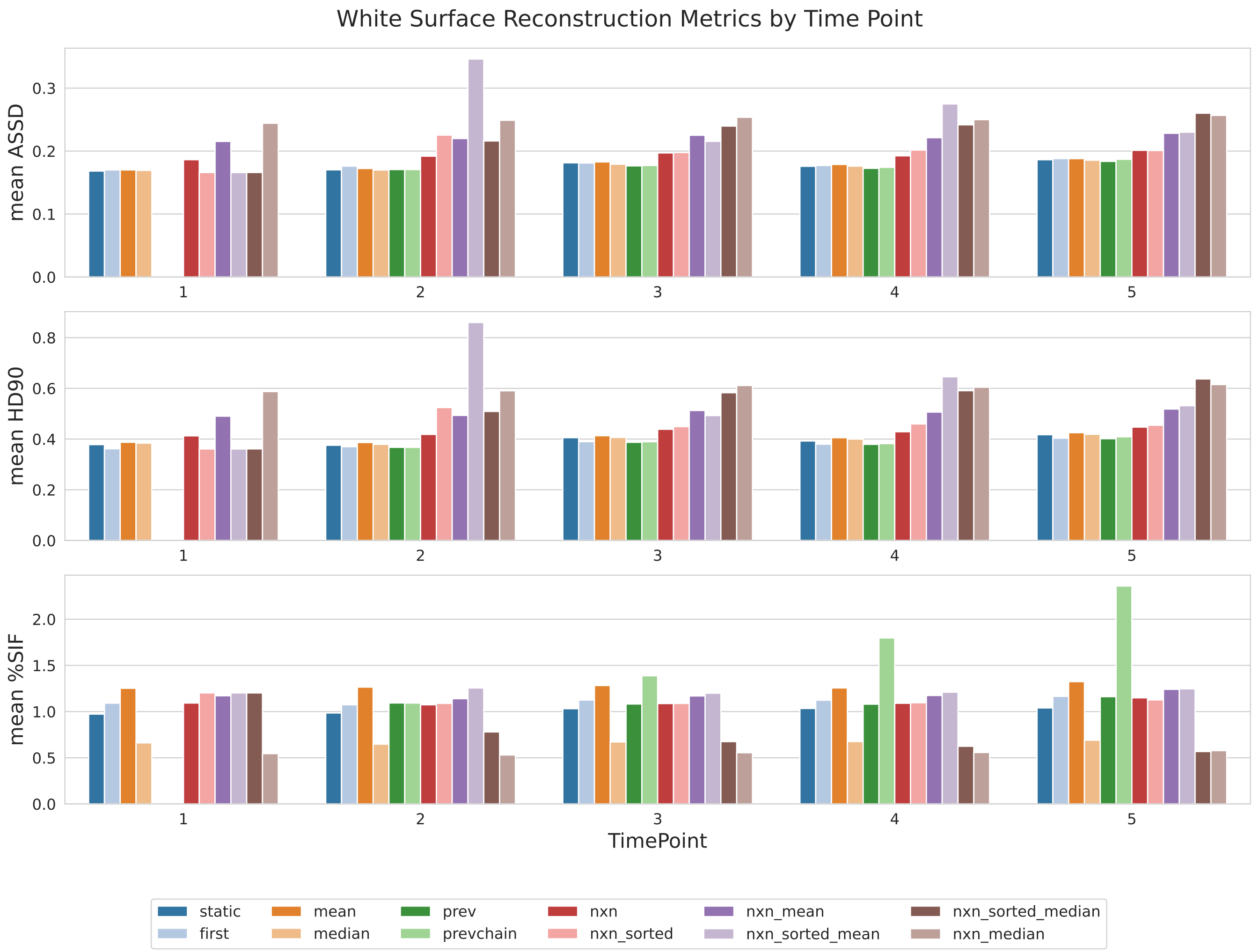


Figure 5.1.: Mean reconstruction metrics of the white surface for the pairing mode ablation study on the ADNI validation set for patients with exactly 5 total scans (n=18). See Figure A.2 in the appendix for the pial surface. The PREV and PREV_CHAIN modes do not yield a direct prediction for the first time point.

may be more extreme for some time points in the dataset.

All models achieve slightly worse results for time points 3, 4, and 5. As this happens even to the STATIC template mode, which predicts all steps independent of the specific patient and time point, we believe that this could be caused by variance in the dataset itself, i.e., the meshes for these time points are slightly harder to predict.

There is a slight relative increase for the PREV_CHAIN mode (light green): For time point 2, its predictions are the same as for the PREV mode, which makes sense since both modes use the same template for this prediction. After that, the PREV_CHAIN mode starts to perform relatively worse for each subsequent time point. This may be related to the increased number of self-intersections for the PREV_CHAIN mode, as described earlier, which may make it more difficult to correctly predict the target surface. However, this effect is quite small.

In summary, there is no direct evidence for an intrinsic pairing mode bias, except for PREV_CHAIN. The pial metrics show a similar trend and are depicted in the appendix (Figure A.2).

### Correspondence by Number of Scans

In addition, we examine bias with respect to the total number of scans per patient. Figure 5.2 shows the correspondence metrics grouped by number of scans per patient and pairing mode for the white surface. Patients with more than 7 scans are excluded as there are fewer than 10 patients in this category in the ADNI validation set. The pial results can be found in the appendix (Figure A.3), and show a similar trend. Clearly, the correspondence metrics get worse the more total scans are evaluated. This is important to note because it means that they are not directly comparable between patients with different numbers of scans and between datasets with different distributions of the number of scans per patient. There may be several reasons for this. First, patients with more scans are likely to have more morphological changes in their anatomy. Second, normalization by $\frac{1}{n}$, as opposed to $\frac{1}{n-1}$ in variance-based metrics, leads to biased estimators that underestimate the true variance [91]. This effect is stronger for smaller sample sizes. However, the relative differences between the pairing modes do not seem to be significantly affected by the number of scans per patient.

### Winning Model

We discard the NxN/NxN_SORTED mean and median aggregation models, as they are not able to reconstruct the mesh sufficiently well (cf. Table 5.2). Of the remaining models, the MEDIAN model performs best in all metrics, except for $V_{L2}$ and ParcF1 on the pial surface, where it is second to the MEAN model. It also significantly reduces the number of self-intersections. In addition, there is no evidence of a model-related bias towards a particular time point or number of scans per patient for the MEDIAN model. Therefore, we select the MEDIAN pairing mode as the "winner" model and use it for the following ablation studies and the final comparison. From now on, when we refer to V2C-Long, we mean the V2C-Long with the MEDIAN pairing mode.

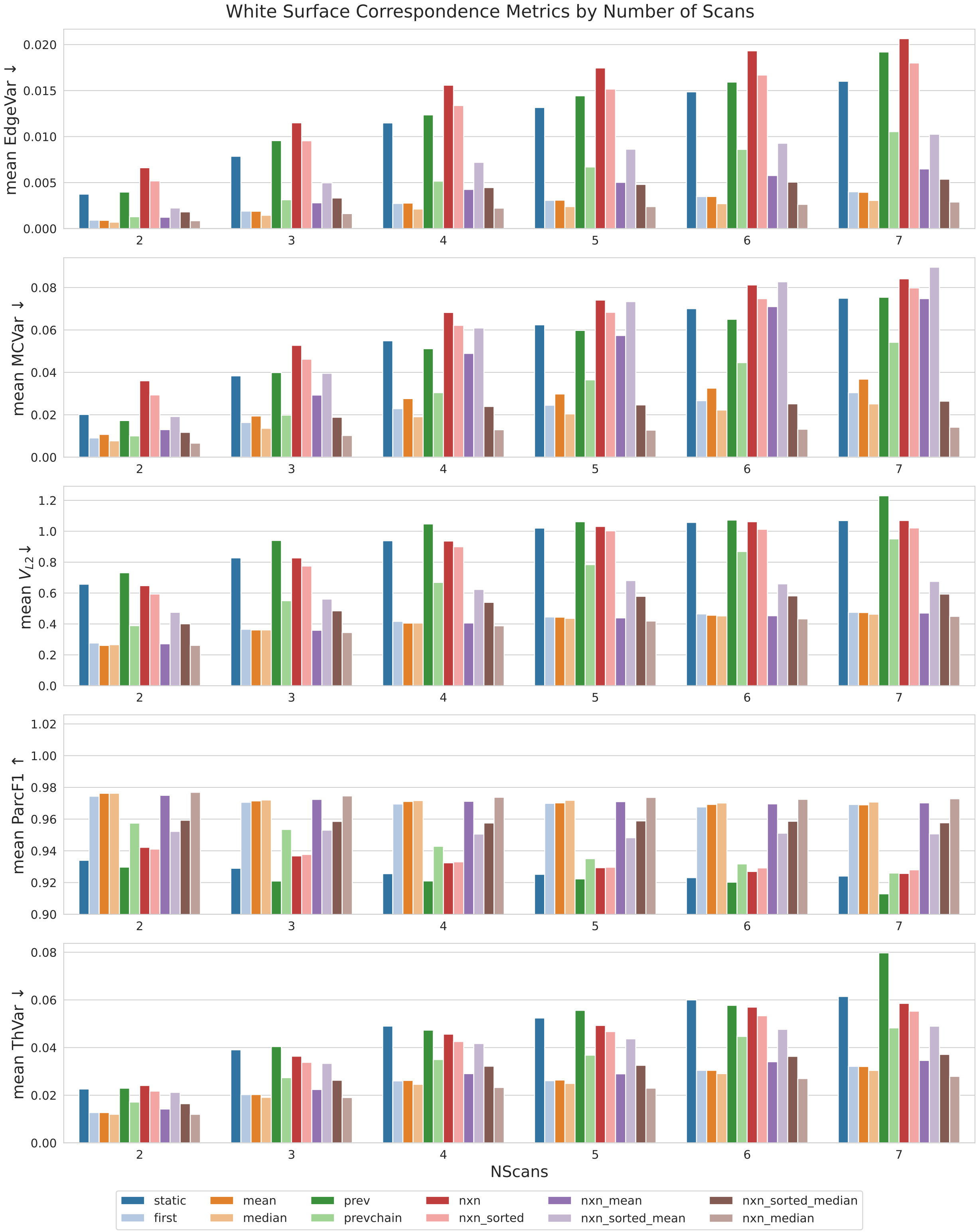


Figure 5.2.: Mean correspondence metrics for the white surface and the thickness on the ADNI validation set, grouped by number of scans per patient and pairing mode. Patients with more than 7 scans are excluded, as there are fewer than 10 patients in that category. See Figure A.3 (appendix) for the pial surface.

### 5.2.2. Vertex Displacement and Mesh Correspondence

In this section, we attempt to explain why the MEDIAN mode outperforms the FIRST and MEAN modes, even though all three operate similarly, by using a patient-specific template.

We believe that the performance difference can be partially explained by the total vertex displacement of the prediction, i.e., how far in total each vertex is moved from the original, patient-specific template position to the predicted position during the two deformation steps. As shown in Tables 5.2b and c, the MEDIAN model has the lowest overall mean displacement of the three models, closely followed by the MEAN model, and with a little more distance by the FIRST model.

Figure 5.3 shows the mean vertex displacement against the correspondence metrics for both the white and pial surfaces. In the scatter plots, each data point represents a patient in the ADNI validation set in one of the three models. There is a clear relationship between the displacement and the correspondence metrics. The higher the displacement, the worse the correspondence metrics tends to be, it is particularly strong for the $V_{L2}$ with a Pearson correlation coefficient for the MEDIAN model of 0.89 for the white surface and 0.91 for the pial surface. The relationship is weaker for the MCVar metric (coefficients of 0.58 and 0.62 respectively), which we believe is due to both the higher sensitivity of the metric and nonlinearities in the metric itself.

For the MEAN model, some outliers can be seen in the displacement, probably caused by the fact that the mean operation is more sensitive to outliers than the median operation, resulting in template vertices that are located farther away from the target surfaces. The displacements for the FIRST model are generally higher than for the other two models because the surface for the first time point (which is used as the template) is typically at a morphological extreme compared to the other time points.

Interestingly, the outliers for the MEAN template do not cause a proportional increase in the correspondence metrics. It also appears that the linear relationships between displacement and correspondence metrics for the models are "shifted" by some offset. This is quite pronounced in the pial $V_{L2}$ plot (top right). This indicates that the lower displacements in the MEDIAN model do not fully explain the better correspondence metrics. The cause of the shift may lie in the way and extent of vertex deformation by the graph NODEs of the mode. For example, if most of the variation in correspondence are introduced by small displacements the second block, potentially larger displacements (e.g., in the FIRST model) in the first block might not affect the correspondence metrics as much. We leave this question open for future work.

For similar plots with more modes included, see Figure A.4 and Figure A.5 in the appendix.

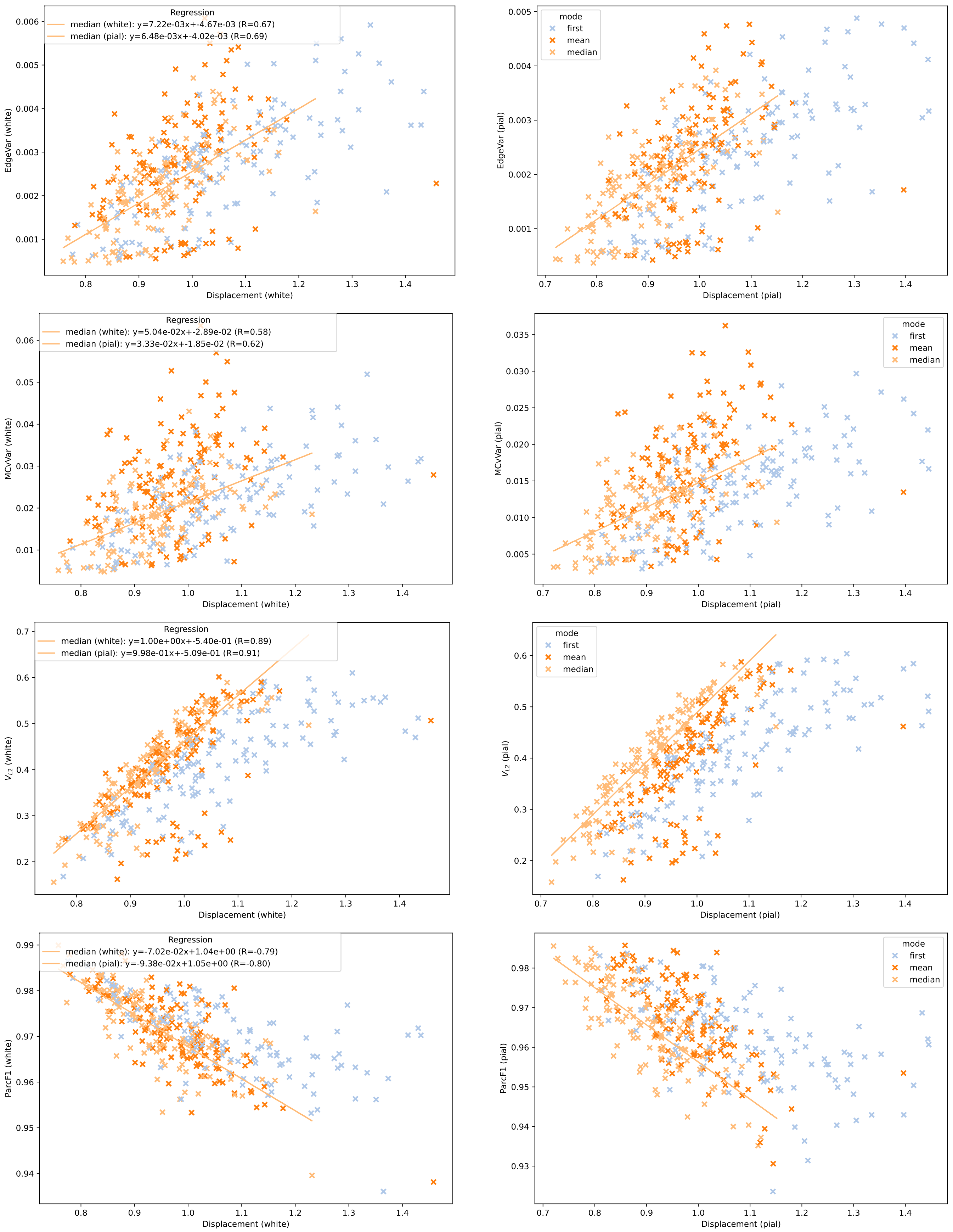


Figure 5.3.: Mean displacement plotted against correspondence metrics for selected modes. Each data point represents a patient in the ADNI validation set. Linear regression results are plotted for the MEDIAN pairing mode. R: Pearson correlation coefficient.

## 5.3. Ablation Study: Template Features

Table 5.3 shows the results of the ablation study on the different template features. The winning pairing mode from the previous ablation study (MEDIAN, from now on the default for V2C-Long) is used for the models in this study. Four models were trained for 45 epochs each: V2C-Long without any additional features (V2C-Long, best epoch: 45), V2C-Long with additional template-target time difference features (V2C-Long+$\Delta$t, best epoch: 42), V2C-Long with latent features from the graph part of the Template Generation Model (TGM) (V2C-Long+glf, best epoch: 42), and a model with both additional features (V2C-Long+$\Delta$t+glf, best epoch: 40). Note that for the time difference features, each input vertex coordinate vector is extended by the time difference between the template and target time point, in months. For the MEDIAN templates, the time difference is set as the relative distance between the target time point and the patient's *mean* time point.

Looking at the metrics, some trends can be observed:

1. **Time Difference Features and Reconstruction Accuracy:** Although one might expect the time difference features to give the model an advantage, as it may learn to predict cortical atrophy over time, the results show that both models using time difference features perform worst in terms of reconstruction accuracy (ASSD and $HD_{90}$). This may be due to the mix of healthy and diseased patients in the dataset, which makes it more difficult to learn a general atrophy pattern. The other two models perform about the same in terms of ASSD and $HD_{90}$. The graph features do not seem to have a significant effect on reconstruction accuracy.

2. **Graph Latent Features and Mesh Structure Correspondence Metrics:** The graph latent features seem to have a positive effect on EdgeVar, ThVar, and MCVar, which are all metrics that are sensitive to the local mesh structure. The V2C-Long+glf model performs best on both surfaces for both metrics, while the V2C-Long+$\Delta$t consistently performs worst.

3. **Time Difference Features and Position-Dependent Correspondence Metrics:** Both metrics that are sensitive to the absolute position of the vertices ($V_{L2}$ and ParcF1) seem to benefit from the time difference features. The V2C-Long+$\Delta$t model performs best in both metrics on both surfaces, followed by V2C-Long+$\Delta$t+glf. Combined with the previous trend, an explanation for the performance of the time difference models could be that the trajectories of corresponding vertices are more similar in these models, potentially leading to less regularity in the mesh structure (bad for EdgeVar, ThVar, and MCVar) and less accurate predictions (ASSD and $HD_{90}$), but to better $V_{L2}$ and ParcF1 scores.

4. **Features and Displacements** Using any of the additional features seems to lower the mean vertex displacement. This may indicate that the model is more confident in its predictions, in the sense that the vertex trajectories more closely follow a straight line from the template to the target position. This explanation is

more plausible in this situation because compared to the previous ablation study, the displacement does not correlate as strongly with the correspondence metrics (cf. Figure 5.3).

In general, the differences between the models are not very large. However, given that the above trends can be observed consistently across metrics and features used, we believe that the additional features do have a significant effect on the model performance and that the differences are not caused by random training fluctuations. More research on the exact vertex-wise effects of the feature additions is needed to fully understand the results.

Although there is no clear winner in terms of metrics, we choose the V2C-Long+glf model as the final model to evaluate on the test set because it consistently performs best in two correspondence metrics while not suffering from a significant decrease in reconstruction accuracy.

| | white surface | | | pial surface | | |
|---|---|---|---|---|---|---|
| **Model** | **ASSD**±std↓ | **HD$_{90}$**±std↓ | **%SIF**±std↓ | **ASSD**±std↓ | **HD$_{90}$**±std↓ | **%SIF**±std↓ |
| V2C-Long | 0.1701±0.0298 | 0.3799±0.0720 | 0.638±0.232 | **0.1640**±0.0221 | **0.3680**±0.0538 | 2.345±0.926 |
| V2C-Long+Δt | 0.1794±0.0316 | 0.4026±0.0760 | 0.743±0.271 | 0.1777±0.0239 | 0.4123±0.0620 | **2.061**±0.990 |
| V2C-Long+glf | **0.1697**±0.0272 | **0.3785**±0.0666 | 0.641±0.243 | 0.1649±0.0216 | 0.3764±0.0543 | 2.168±0.954 |
| V2C-Long+Δt+glf | 0.1723±0.0269 | 0.3815±0.0663 | **0.615**±0.228 | 0.1656±0.0209 | 0.3788±0.0538 | 2.303±1.017 |

(a) Reconstruction results for the pial and white surfaces.

| | white surface | | | | | both |
|---|---|---|---|---|---|---|
| **Model** | **EdgeVar**±std↓ | **MCVar**±std↓ | **V$_{L2}$**±std↓ | **ParcF1**±std↑ | **Disp**±std↓ | **ThVar**±std↓ |
| V2C-Long | 0.00161±0.00080 | 0.01569±0.00679 | 0.3918±0.1093 | 0.9731±0.0091 | 0.7469±0.1584 | 0.0231±0.0103 |
| V2C-Long+Δt | 0.00199±0.00095 | 0.02061±0.00869 | **0.3741**±0.1022 | **0.9760**±0.0084 | 0.6947±0.1379 | 0.0247±0.0106 |
| V2C-Long+glf | **0.00137**±0.00068 | **0.01445**±0.00634 | 0.3810±0.1068 | 0.9746±0.0091 | 0.6894±0.1566 | 0.0212±0.0095 |
| V2C-Long+Δt+glf | 0.00140±0.00068 | 0.01506±0.00654 | 0.3745±0.1033 | 0.9759±0.0085 | **0.6699**±0.1458 | **0.0209**±0.0093 |

(b) Correspondence results for the white surface and the results for the thickness variance.

| | pial surface | | | | |
|---|---|---|---|---|---|
| **Model** | **EdgeVar**±std↓ | **MCVar**±std↓ | **V$_{L2}$**±std↓ | **ParcF1**±std↑ | **Disp**±std↓ |
| V2C-Long | 0.00137±0.00069 | 0.00942±0.00402 | 0.3848±0.1082 | 0.9667±0.0117 | 0.7571±0.1567 |
| V2C-Long+Δt | 0.00156±0.00077 | 0.01305±0.00544 | **0.3606**±0.1015 | **0.9713**±0.0111 | **0.6673**±0.1378 |
| V2C-Long+glf | **0.00119**±0.00058 | **0.00878**±0.00371 | 0.3704±0.1050 | 0.9694±0.0116 | 0.6801±0.1524 |
| V2C-Long+Δt+glf | 0.00124±0.00060 | 0.00941±0.00400 | 0.3656±0.1025 | 0.9696±0.0116 | 0.6695±0.1447 |

(c) Correspondence results the pial surface.

Table 5.3.: Ablation study results for additional template features for both reconstruction and correspondence metrics on the ADNI validation set. The best result for each metric is highlighted. For a description of each metric, see Subsection 4.6.2 and Section 3.2.

## 5.4. Comparison with Related Methods

After the ablation studies, we evaluate the final version of the model, V2C-Long+glf (MEDIAN pairing mode, template with additional graph latent features) on the ADNI test set, the OASIS, and TRT datasets. In addition, we evaluate V2C-Flow [6], CorticalFlow++ [17], V2CC [77], and FreeSurfer longitudinal (c.f. Subsection 2.3.1) on the same datasets. In Table 5.4, the reconstruction metrics for the pial and white surfaces are shown, with the best non-FreeSurfer model highlighted in bold. In Table 5.5, the mesh correspondence metrics are shown, with the best value for each metric highlighted. The ParcF1 metric was only evaluated for V2C-Flow, V2C-Long, and V2CC, as these are the only methods that derive their meshes from FreeSurfer's fsaverage subject, whose parcellation labels are used for the ParcF1 metric.

### 5.4.1. Reconstruction Accuracy

On the ADNI test set, V2C-Long achieves a mean ASSD of 0.1767 mm on the white surface and 0.1736 mm on the pial surface, and a mean $HD_{90}$ of 0.4102 mm and 0.4092 mm respectively. This makes it the best of the non-FreeSurfer methods, beating even V2C-Flow, the method on which it is based, by a significant margin on all datasets (except $HD_{90}$ on the pial surface, where it is second to CorticalFlow++ by a small margin). This is consistent with the results obtained in the first ablation study (see Table 5.2a), where the MEDIAN pairing mode (which V2C-Long uses) outperforms the STATIC mode, which is equivalent to V2C-Flow. A possible reason for this could be that V2C-Long's patient-specific median template provides the model with a good prior for the target surface. Since some work has been put into training the Template Generation Model (TGM), this can be seen as an unfair advantage for V2C-Long. Therefore, the result should be taken with a grain of salt, as it is unclear if this effect will persist as the models are trained for more epochs, or whether the ASSD and $HD_{90}$ validation scores will converge to similar values.

Although V2CC uses a similar architecture to V2C-Flow and V2C-Long, it performs worst in both ASSD and $HD_{90}$ and particularly struggles on the pial surface (with an ASSD score of 0.2432 and an $HD_{90}$ score of 0.5578). The low performance could be related to the resampled ground truth meshes or to the Vox2Cortex deformation blocks used in V2CC.

In terms of self-intersections, V2C-Long beats V2C-Flow by a significant margin (1.005 vs. 0.643 and 2.151 vs. 2.443 %SIF respectively), which is likely due to the low self-intersections in the median templates (see Table 5.1), which are not significantly amplified by the low displacements in the V2C-Long model. However, both models lag behind V2CC and CorticalFlow++, with V2CC having the lowest number of self-intersections on the white surface (0.456 %SIF), and CorticalFlow++ having the lowest number on the pial surface (0.3679 %SIF), excluding FreeSurfer.

FreeSurfer longitudinal has the highest overall reconstruction accuracy, which is not surprising as the ground truth meshes are derived from FreeSurfer itself, albeit from the

| | | white surface | | | pial surface | | |
|---|---|---|---|---|---|---|---|
| | **Model** | **ASSD**±std↓ | **HD$_{90}$**±std↓ | **%SIF**±std↓ | **ASSD**±std↓ | **HD$_{90}$**±std↓ | **%SIF**±std↓ |
| ADNI | V2C-Flow | 0.1857±0.1160 | 0.4264±0.6453 | 1.0049±0.3887 | 0.1798±0.1087 | 0.4185±0.5618 | 2.4425±0.8981 |
| | V2C-L.+glf | **0.1767**±0.1607 | **0.4102**±0.7512 | 0.6425±0.3752 | **0.1736**±0.1602 | **0.4092**±0.7295 | 2.1507±0.9213 |
| | CF++ | 0.2140±0.1393 | 0.4923±0.8063 | 0.1728±0.2524 | 0.1907±0.1296 | 0.4456±0.8064 | **0.3679**±0.4290 |
| | V2CC | 0.2197±0.0363 | 0.4923±0.0888 | **0.0456**±0.0741 | 0.2432±0.0401 | 0.5578±0.1044 | 1.6935±0.8550 |
| | FS long. | 0.1513±0.0890 | 0.3172±0.1864 | 0.0012±0.0023 | 0.1446±0.0779 | 0.2995±0.2210 | 0.0042±0.0047 |
| OASIS | V2C-Flow | 0.1844±0.0238 | 0.4187±0.0551 | 1.0746±0.2714 | 0.1963±0.0243 | 0.4525±0.0618 | 2.8280±1.1284 |
| | V2C-L.+glf | **0.1762**±0.0227 | **0.4028**±0.0525 | 0.7670±0.2632 | **0.1855**±0.0254 | **0.4296**±0.0649 | 2.5683±1.0866 |
| | CF++ | 0.2261±0.0323 | 0.5217±0.0785 | 0.2076±0.1337 | 0.2104±0.0410 | 0.4735±0.0780 | **0.4672**±0.2238 |
| | V2CC | 0.2215±0.0356 | 0.5060±0.0818 | **0.0408**±0.0562 | 0.2816±0.0436 | 0.6539±0.1156 | 1.9513±0.9430 |
| | FS long. | 0.1259±0.0244 | 0.2640±0.0535 | 0.0013±0.0016 | 0.1321±0.0250 | 0.2740±0.0549 | 0.0042±0.0033 |
| TRT | V2C-Flow | 0.1989±0.0152 | 0.4792±0.0470 | 1.3392±0.4193 | 0.2807±0.0395 | 0.6331±0.0925 | 4.4295±0.7092 |
| | V2C-L.+glf | **0.1913**±0.0130 | **0.4587**±0.0376 | 0.6622±0.2196 | **0.2656**±0.0361 | 0.6092±0.0929 | 3.9161±0.6100 |
| | CF++ | 0.2590±0.0146 | 0.6687±0.0554 | 0.1765±0.1283 | 0.2879±0.0316 | **0.6057**±0.0576 | **0.4416**±0.2462 |
| | V2CC | 0.2275±0.0288 | 0.5440±0.0793 | **0.0908**±0.0872 | 0.4339±0.0596 | 1.0760±0.2042 | 2.9964±0.4240 |
| | FS long. | 0.1483±0.0079 | 0.3102±0.0123 | 0.0012±0.0014 | 0.1700±0.0180 | 0.3540±0.0406 | 0.0054±0.0032 |

Table 5.4.: Reconstruction metrics by surface, method, and dataset. The best non-FreeSurfer method for each metric is highlighted in bold.

cross-sectional pipeline. Nevertheless, there is a difference of around 0.15 mm in ASSD and 0.3 mm in HD$_{90}$ for both surfaces. This may be caused by variance in FreeSurfer's reconstructions. Similar reconstruction inconsistencies (in the cross-sectional pipeline) have been observed in the literature [5, Table 3]. FreeSurfer longitudinal meshes contain almost no self-intersections (they occur on less than 0.01% of faces).

The reconstruction accuracy of V2C-Long and V2CC generalizes well to other datasets, with V2C-Long even achieving better scores on the white OASIS surface than on the ADNI dataset. V2CC struggles with the pial surface on the TRT dataset, where the ASSD and HD$_{90}$ scores are almost double that of ADNI (0.2342 vs. 0.4339 and 0.5578 vs. 1.0760 respectively). As explained in Section 4.1, the TRT dataset contains significantly younger patients than the other datasets, which may explain the poorer performance of the methods.

### 5.4.2. Mesh Correspondence

As shown in Table 5.5, in terms of correspondence metrics, our method (V2C-Long) consistently performs best and even beats FreeSurfer longitudinal. FreeSurfer longitudinal only narrowly wins in the EdgeVar metric on the OASIS and TRT datasets. This shows that our method successfully predicts highly aligned intra-patient meshes. The overall trend in terms of mesh correspondence is (from best to worst): V2C-Long > FreeSurfer longitudinal ≈ V2CC > CorticalFlow++ ≈ V2C-Flow, with large gaps between V2CC/FreeSurfer longitudinal and CorticalFlow++/V2C-Flow. V2CC beats FreeSurfer in the ThVar and MCVar metrics, while FreeSurfer beats V2CC in the EdgeVar and $V_{L2}$ metrics. It is unclear why.

The good performance of V2CC is probably due to the L1-loss used in training with resampled ground truth meshes, which leads to more consistent vertex positions in the predictions. The vertex correspondence in V2CC is designed to be cross-sectional, i.e., between intra- and inter-subject meshes [77], which may be a limiting factor compared to V2C-Long and FreeSurfer, which both focus on intra-subject mesh correspondence.

Given that both V2C-Flow and CorticalFlow++ are not designed with mesh correspondence in mind, it is not surprising that they do not perform as well as the other, more dedicated methods. However, the regularity losses employed by both methods can help with the performance of certain metrics. For example, the edge loss, which penalizes different edge lengths *within a single mesh* [5, 6, 92] may push the model to produce uniform edge lengths, which should lead to similar corresponding edge lengths *between meshes*, if other mesh properties (e.g., the total surface area) are similar. This in turn leads to better mesh correspondence and a better EdgeVar metric. Similar effects should exist with the normal consistency loss of V2C-Flow [6, 5] and the MCVar metric.

**Regional Differences**

To see which regions of the brain tend to have a worse mesh correspondence, we plot the correspondence metrics on cortical surfaces in Figure 5.4. The metrics were evaluated for each vertex and for each patient in the ADNI test set. Instead of taking the mean per patient over all vertices, we take a vertex-wise median over all patients to get a single value for each vertex. For the EdgeVar, the vertex values are computed as the mean EdgeVar value of the connected edges. The plot shows that V2C-Long consistently outperforms other methods, on all parts of the surface. Consistent with the mean results in Table 5.5, the second best method is V2CC and both V2C-Flow and CF++ perform worst.

For the EdgeVar metric, both V2C-Flow and CorticalFlow++ struggle with corresponding edge length consistency around the visual cortex/occipital lobe at the back of the brain, and the temporal lobe at bottom on the side of the brain. For the MCVar metric, all methods struggle at the inner part of the temporal lobe, close to the entorhinal cortex. The FreeSurfer ground truth meshes contain sharp "ridges" in this area, explaining the high variance in mean curvature. Similar unrealistic ridges in the FreeSurfer ground truth meshes are found in the inner part of the corpus callosum, where the methods also struggle with the MCVar metric. Around these two regions, some outlier values exceeding 1200 are reached. These, however, cannot be identified in the plot, because the color scale is limited to 0.1.

The plot also shows that the $V_{L2}$ metric does not contain clear outlier regions, but is rather smoothly distributed over the entire surface, while both MCVar and EdgeVar have clear outlier regions, which motivates the use of the median over the vertices for these methods in the final results presented in the tables.

| | | **white surface** | | | | **both** |
|---|---|---|---|---|---|---|
| | **Model** | **EdgeVar**±std↓ | **MCVar**±std↓ | $\mathbf{V_{L2}}$±std↓ | **ParcF1**±std↑ | **ThVar**±std↓ |
| ADNI | V2C-Flow | 0.01038±0.00486 | 0.04463±0.01927 | 0.9611±0.3428 | 0.9235±0.0301 | 0.0503±0.0327 |
| | V2C-L.+glf | **0.00141**±0.00123 | **0.01474**±0.01133 | **0.4067**±0.1291 | **0.9710**±0.0154 | **0.0217**±0.0149 |
| | CF++ | 0.00876±0.00484 | 0.03398±0.01557 | 0.9291±0.5240 | – | 0.0435±0.0288 |
| | V2CC | 0.00269±0.00122 | 0.02351±0.00959 | 0.6306±0.2049 | 0.9585±0.0155 | 0.0232±0.0100 |
| | FS long. | 0.00168±0.00099 | 0.02414±0.01212 | 0.5877±1.1955 | – | 0.0354±0.0198 |
| OASIS | V2C-Flow | 0.00932±0.00475 | 0.04095±0.01986 | 1.0101±0.2448 | 0.9113±0.0200 | 0.0501±0.0203 |
| | V2C-L.+glf | 0.00129±0.00074 | **0.01327**±0.00644 | **0.4321**±0.1312 | **0.9630**±0.0160 | **0.0221**±0.0097 |
| | CF++ | 0.00877±0.00474 | 0.03221±0.01663 | 1.0775±0.3276 | – | 0.0504±0.0246 |
| | V2CC | 0.00259±0.00129 | 0.02210±0.01002 | 0.7131±0.2007 | 0.9470±0.0170 | 0.0263±0.0116 |
| | FS long. | **0.00127**±0.00081 | 0.02077±0.01102 | 0.5361±0.1762 | – | 0.0332±0.0149 |
| TRT | V2C-Flow | 0.01850±0.00206 | 0.07944±0.01905 | 1.0472±0.0775 | 0.9331±0.0044 | 0.0726±0.0059 |
| | V2C-L.+glf | 0.00222±0.00018 | **0.01976**±0.00235 | **0.4004**±0.0150 | **0.9819**±0.0010 | **0.0301**±0.0005 |
| | CF++ | 0.01638±0.00299 | 0.06391±0.01585 | 1.0217±0.0703 | – | 0.0676±0.0038 |
| | V2CC | 0.00485±0.00093 | 0.03861±0.00629 | 0.6475±0.0630 | 0.9664±0.0036 | 0.0362±0.0030 |
| | FS long. | **0.00218**±0.00026 | 0.03972±0.00402 | 0.4830±0.0122 | – | 0.0437±0.0003 |

(a) Correspondence results for the white surface and thickness variance.

| | | **pial surface** | | | |
|---|---|---|---|---|---|
| | **Model** | **EdgeVar**±std↓ | **MCVar**±std↓ | $\mathbf{V_{L2}}$±std↓ | **ParcF1**±std↑ |
| ADNI | V2C-Flow | 0.01029±0.00491 | 0.03478±0.01506 | 1.0003±0.3590 | 0.9200±0.0284 |
| | V2C-L.+glf | **0.00122**±0.00110 | **0.00906**±0.00804 | **0.3955**±0.1298 | **0.9659**±0.0167 |
| | CF++ | 0.01412±0.00716 | 0.03428±0.01590 | 1.0004±0.5469 | – |
| | V2CC | 0.00290±0.00143 | 0.02120±0.00876 | 0.6628±0.2171 | 0.9453±0.0180 |
| | FS long. | 0.00158±0.00105 | 0.02294±0.01339 | 0.5870±1.2109 | – |
| OASIS | V2C-Flow | 0.00936±0.00488 | 0.03192±0.01614 | 1.0626±0.2602 | 0.9044±0.0219 |
| | V2C-L.+glf | **0.00116**±0.00069 | **0.00839**±0.00435 | **0.4247**±0.1309 | **0.9533**±0.0189 |
| | CF++ | 0.01425±0.00771 | 0.03301±0.01752 | 1.1788±0.3511 | – |
| | V2CC | 0.00290±0.00154 | 0.02157±0.01000 | 0.7599±0.2096 | 0.9289±0.0213 |
| | FS long. | 0.00165±0.00111 | 0.02562±0.01630 | 0.5513±0.1803 | – |
| TRT | V2C-Flow | 0.01929±0.00226 | 0.07731±0.01241 | 1.1429±0.0751 | 0.9176±0.0036 |
| | V2C-L.+glf | **0.00220**±0.00010 | **0.01625**±0.00087 | **0.4079**±0.0085 | **0.9663**±0.0031 |
| | CF++ | 0.02996±0.00510 | 0.07791±0.01857 | 1.1580±0.0734 | – |
| | V2CC | 0.00558±0.00124 | 0.04186±0.00785 | 0.7130±0.0694 | 0.9474±0.0019 |
| | FS long. | 0.00624±0.00102 | 0.12010±0.02224 | 0.5112±0.0107 | – |

(b) Correspondence results for the pial surface.

Table 5.5.: Correspondence metrics by surface, dataset, and method. The best result for each metric is highlighted in bold. For a description of each metric, see Subsection 4.6.2 and Section 3.2

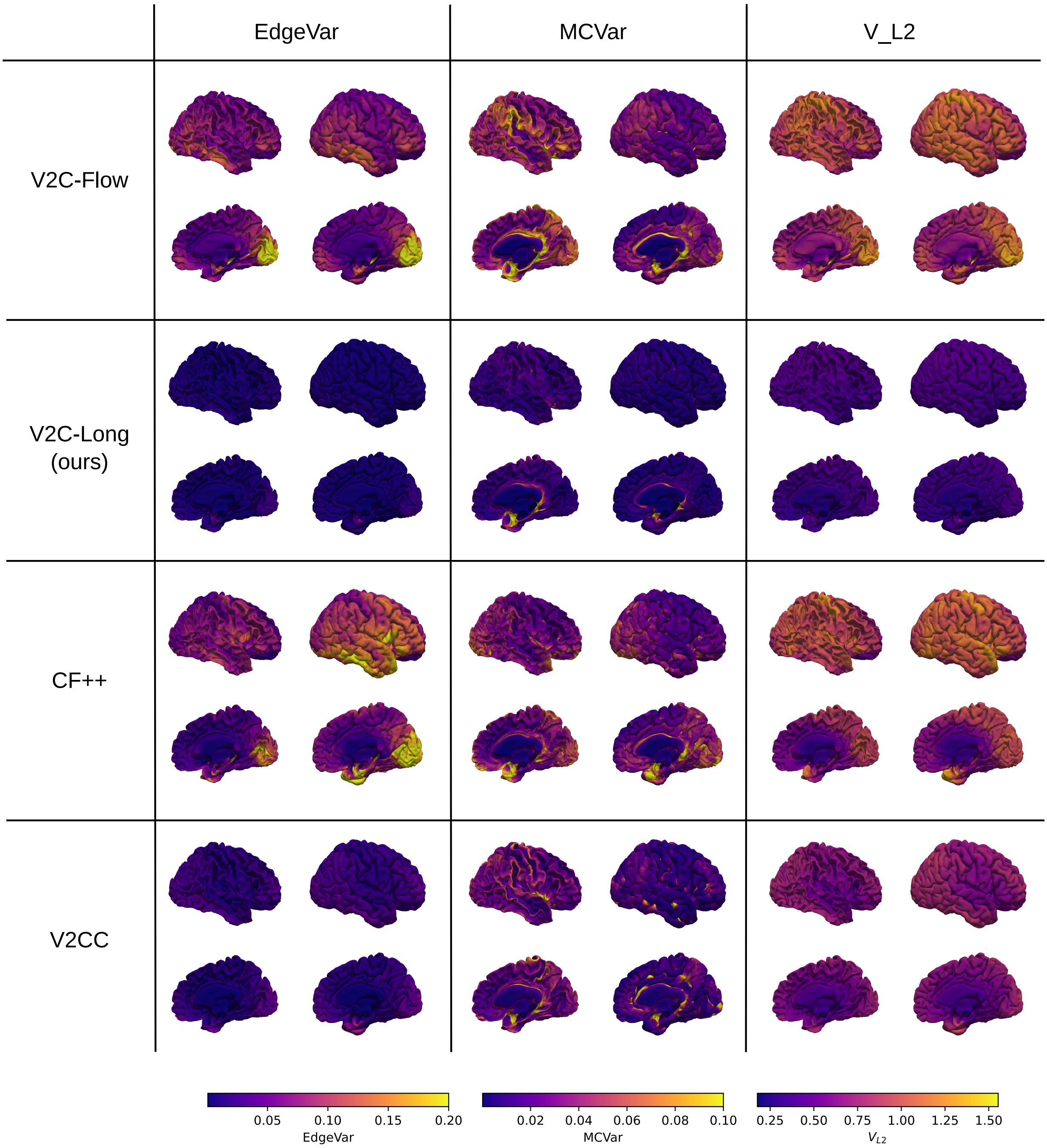


Figure 5.4.: Vertex-wise correspondence metrics (median over all patients) on the ADNI test set for different methods plotted on a randomly chosen ADNI brain. FreeSurfer longitudinal is excluded, as the inter-patient meshes vary in the number of vertices. The EdgeVar and MCVar values are max-limited to 0.2 and 0.1, respectively, to avoid outliers from dominating the color scale.

### Parcellation Consistency

As seen in Table 5.5, the ParcF1 metric differs considerably between the V2C-Flow, V2C-Long, and V2CC methods. For all datasets and surfaces, V2C-Long performs best, followed by V2CC, and then V2C-Flow. To get a clearer picture of the differences between the methods, we plot the number of mismatches between parcellation labels of nearest-neighbors across time points in Figure 5.5. The number of mismatches is computed as described in Subsection 3.2.3, before the F1-score is applied, and then summed over the vertex ids for a patient. This is done for each of the three participants in the test-retest (TRT) dataset and for each of the three methods. Since there should be little to no change in the morphological ground truth for these subjects, the number of mismatches should reflect well the mesh consistency of the methods. Vertices that are never misclassified are shown in gray, while misclassified regions are highlighted in red/yellow.

The misclassifications appear around the parcellation region boundaries defined by the fsaverage Destrieux parcellation labels and are caused by relative shifts/offsets of corresponding vertices in the intra-patient mesh predictions. The wider the colored areas around a parcellation boundary, the stronger the relative shifts produced by each method. V2C-Long produces fewer mismatches than the other methods and the meshes have less severe misalignments, as indicated by narrower highlighted regions around the parcellation boundaries. This indicates an overall better correspondence of the intra-patient meshes produced by V2C-Long, which is consistent with the results in Table 5.5, including the other correspondence metrics.

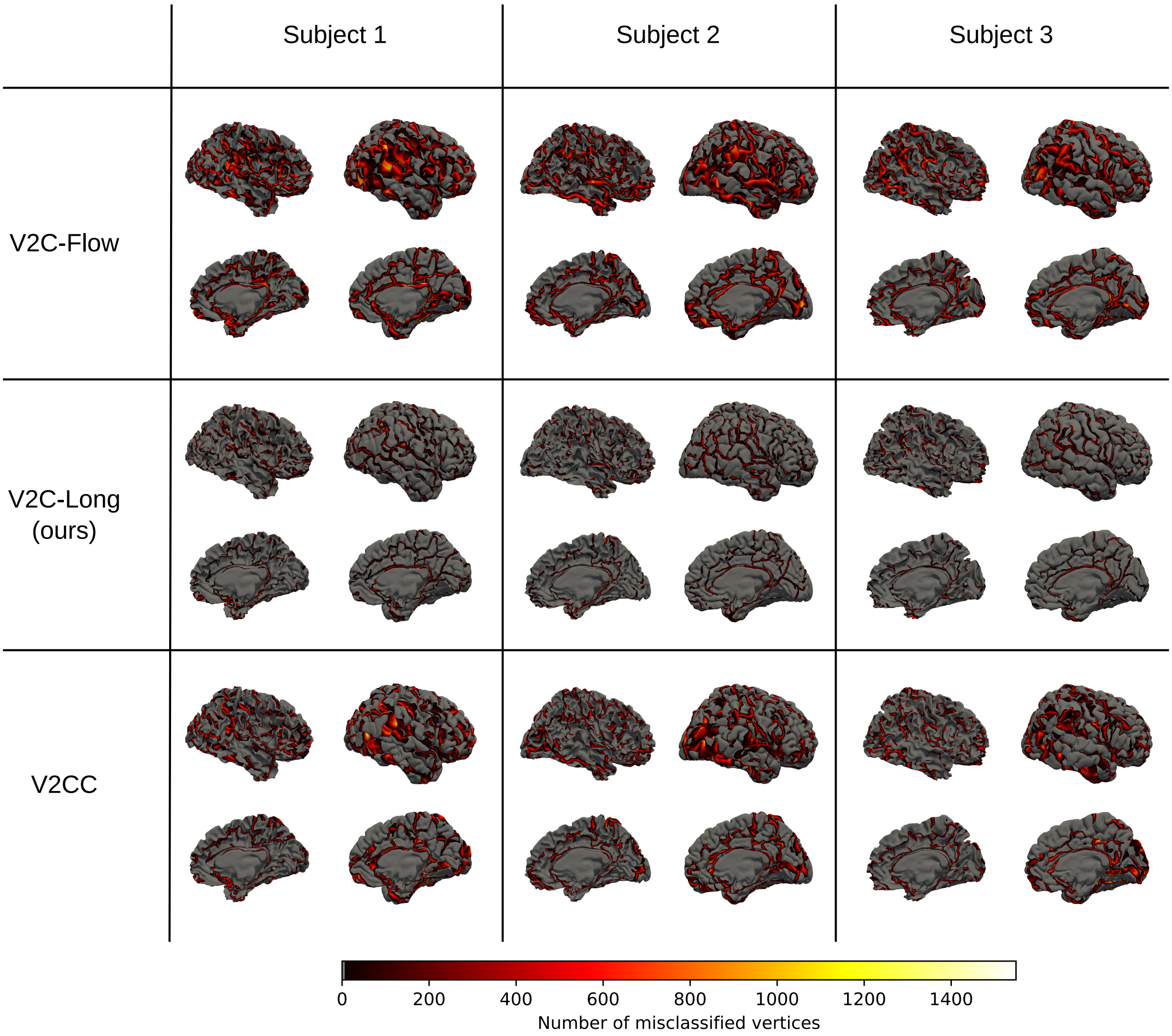


Figure 5.5.: Number of mismatches in the Destrieux parcellation labels between nearest-neighbors across time points per vertex (see Subsection 3.2.3) for the three TRT subjects and for V2C-Flow, V2C-Long and V2CC. Vertices that are never misclassified are shown in gray. The missclassifications take place around the parcellation region boundaries. Bigger red regions suggest that the methods produces longitudinal meshes where the extent of parcellation regions overlap is larger (i.e. corresponding regions of the meshes are shifted by a larger amount). V2C-Long produces fewer misclassifications than V2C-Flow and V2CC and has significantly narrower regions of misclassification.

# 6. Conclusion

## 6.1. Summary

We presented a new method, called V2C-Long, which produces longitudinal, intra-subject aligned meshes in about twice the training and inference time compared to V2C-Flow. By conducting two ablation studies, we found a good configuration of V2C-Long by combining a patient-wise median template with additional vertex input features extracted from the template generation process. Our method, applicable to any number of scans, produces meshes that are state of the art in terms of vertex correspondence and reconstruction accuracy, and have significantly fewer self-intersecting faces than V2C-Flow, the method on which it is based on.

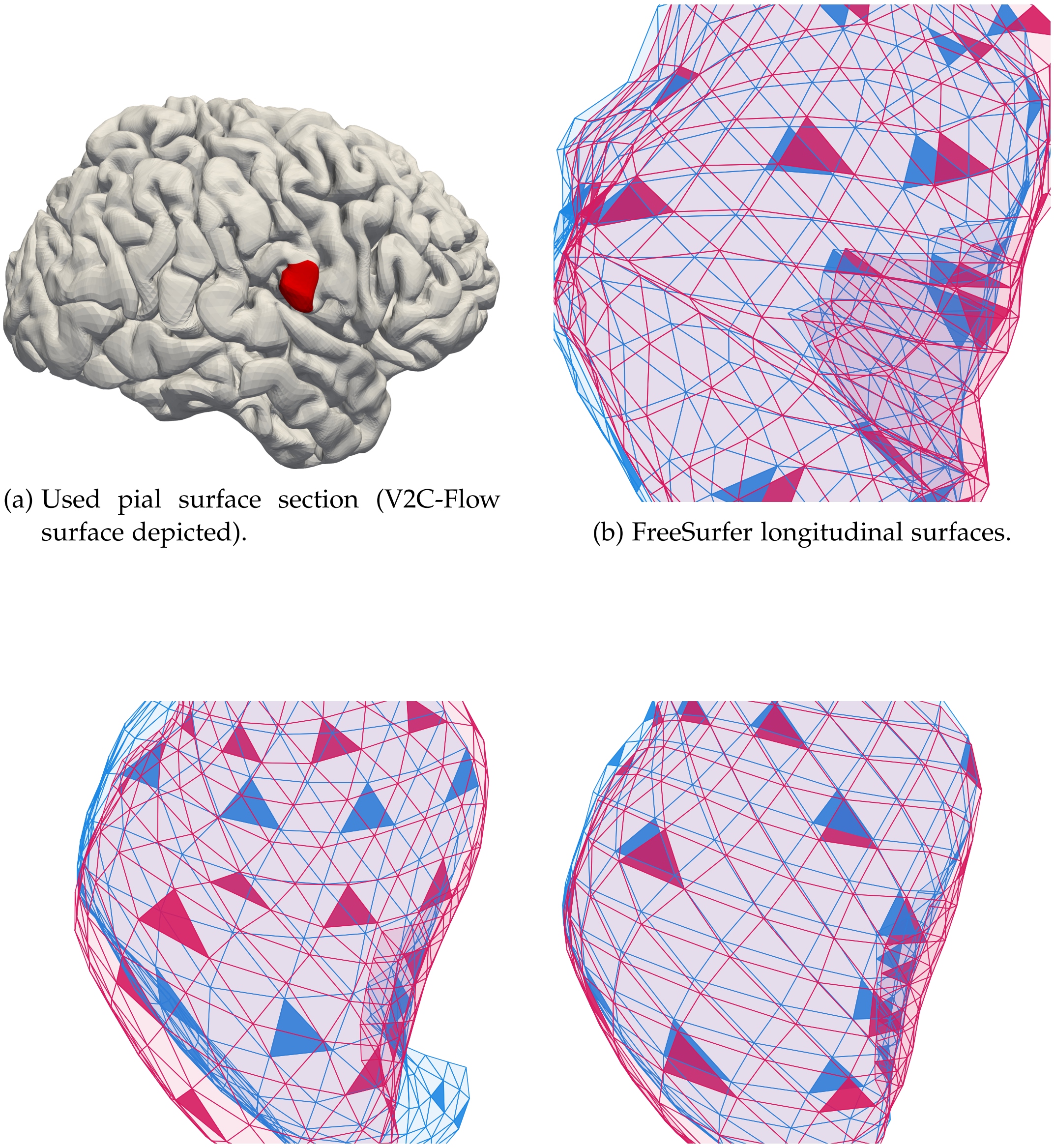
Finally, our general approach is not limited to the V2C-Flow method, but can be applied to any explicit mesh deformation method, where the produced meshes have the same number of vertices and the same vertex connectivity.

To illustrate the improved mesh alignment of V2C-Long, Figure 6.1 shows the same pial surface region and pair of reconstructed surfaces as Figure 1.1 in the Introduction. This time, the meshes produced by our final V2C-Long model are also shown. V2C-Long combines the positive aspects of V2C-Flow and FreeSurfer longitudinal by aligning meshes as well as FreeSurfer longitudinal, but also by producing meshes with similar smoothness and regularity as V2C-Flow.

In addition to the new method, we introduced a set of new, well-motivated metrics to measure mesh correspondence. The metrics show similar trends, indicating that they measure the same underlying property. The qualitative results, as seen for example in Figure 6.1, suggesting that good values for these metrics are a good indicator of high intra-subject correspondence of meshes.

## 6.2. Limitations and Outlook

Since this work marks a first step in a new and promising direction – deep learning for longitudinal cortical surface reconstruction – there is room for further research and improvement. We highlight some potential research directions below.



(a) Used pial surface section (V2C-Flow surface depicted).

(b) FreeSurfer longitudinal surfaces.

(c) V2C-Flow surfaces.

(d) V2C-Long surfaces.

Figure 6.1.: Mesh correspondence visualized on two pial surfaces of two intra-subject scans (TRT s1_01, s1_02). V2C-Long meshes have similar smoothness and regularity as V2C-Flow predictions and at the same time have a mesh correspondence that is at least as good as FreeSurfer longitudinal meshes.

### Correspondence Metrics

The correspondence metrics we introduced in Section 3.2 are the first of their kind, and we have found them useful for evaluating mesh correspondence. However, they have some limitations. First, our correspondence metrics are not scale independent (e.g., the L2 distance between two vertices is not scale invariant). In addition, we have found that the metrics depend on the number of meshes that are being compared (cf. Figure 5.2). Both of these properties make it difficult to compare the metrics across methods and datasets. However, a simple way to reduce the impact of the total number of meshes is to normalize the variance-based metrics (MCVar, EdgeVar, and ThVar) with $\frac{1}{n-1}$ instead of $\frac{1}{n}$. For now, however, we suggest applying these metrics only to meshes in the same coordinate system and not comparing them across different datasets.

An interesting direction for future work would be to use concepts from differential geometry and shape analysis, such as shape descriptors [100, 15, 66], to define correspondence metrics that can capture whether vertices are located at the same anatomical location in a way that is robust to larger morphological changes. However, in our experience, shape analysis methods often fail on cortical surfaces due to the large number of vertices and the high degree of folding.

Another interesting direction for further metrics would be to measure the consistency of changes across neighboring vertices and time points.

### Bias in Dataset

As explained in Section 4.1, in both the ADNI and OASIS datasets, some patients have more scans than others. This introduces sampling bias into the dataset, since some patients are overrepresented potentially making the model learn to reconstruct these patients better. In addition, the timing of the scans is not consistent across patients, which can also introduce a bias. This is important to keep in mind when comparing the performance of different methods on these datasets.

### Learnable Mesh Correspondence

In this work, we create a new pipeline for longitudinal mesh reconstruction by taking an existing CSR method (V2C-Flow) and modifying the inputs used for the method. However, the core architecture of V2C-Flow is not changed, except for the possibility of additional vertex features. Therefore, the mesh correspondence is not learned, but is a result of combining and modifying the inputs and outputs of the method.

Future work could investigate whether the mesh correspondence can be learned explicitly and unsupervised using tailored loss functions, architectures, or training methods, such as Siamese networks [99] or contrastive learning [35, 33, 11]. This may be a difficult task however, as one must ensure that the reconstructed meshes of a patient are not over-regularized and still reconstruct the ground truth meshes well without a bias towards a “mean” mesh.

### Downstream Usage of V2C-Long Meshes

We speculate that the generated meshes by V2C-Long are suitable for downstream tasks such as thickness analysis because they have similar or better alignment than FreeSurfer longitudinal, which is currently the standard way to reconstruct meshes for longitudinal analysis [64, 56]. However, we leave this question open for future work.

# A. Appendix

| Model Type | Peak RAM | Peak VRAM | Epoch Duration | Eval. Duration |
|---|---|---|---|---|
| STATIC/V2C-Flow | ~423 GB | ~38 GB | ~5h | ~3h40min |
| FIRST | ~197 GB | ~38 GB | ~5h | ~3h10min |
| PREV | ~193 GB | ~38 GB | ~3h30min | ~2h20min |
| NxN | ~330 GB | ~38 GB | ~40h | ~31h |
| NxN_SORTED | ~351 GB | ~38 GB | ~16h | ~17h30min |
| MEAN/MEDIAN | ~182 GB | ~38 GB | ~5h | ~3h10min |
| MEDIAN+glf | ~359 GB | ~38 GB | ~5h30min | ~3h25min |
| V2CC | ~837 GB | ~27 GB | ~2h15min | ~2h50min |

Table A.1.: Hardware resources used for different models trained on a NVIDIA A100 GPU with 40GB of VRAM. The eval duration refers to the evaluation of the ADNI validation set. Peak RAM is equal to the MaxRSS reported by SLURM [98] and the higher RAM usage for V2CC is caused by loading the full resampled ground truths into memory. The training for V2C-Long models required less RAM as the training for V2C-Flow, because of additional optimizations. The addition of time difference features did not change the hardware resources used significantly. The V2C-Flow based models require more VRAM than the V2CC model due to the different architecture used for the graph deformation blocks. glf: additional graph features.

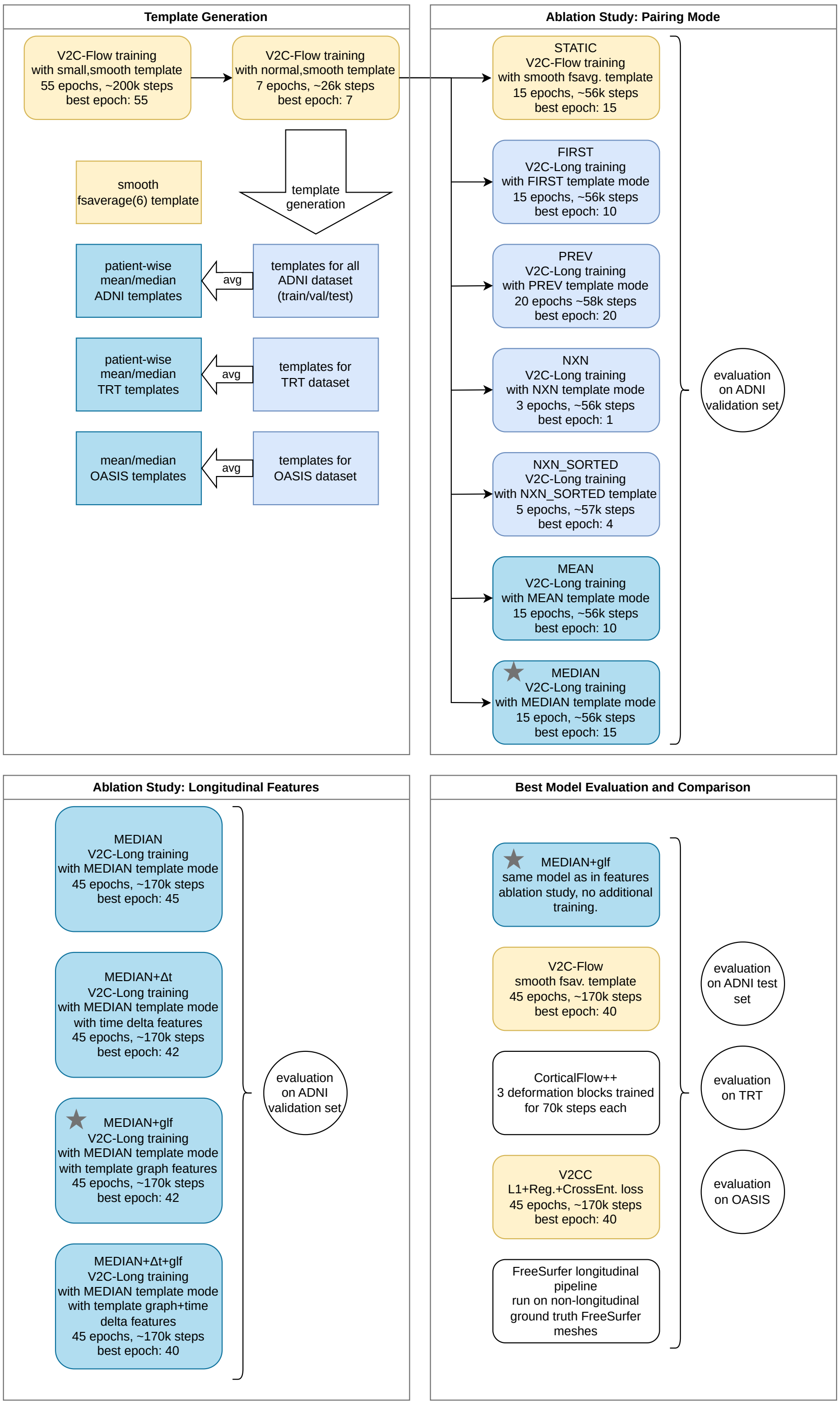

Template Generation
V2C-Flow training with small,smooth template 55 epochs, ~200k steps best epoch: 55
V2C-Flow training with normal,smooth template 7 epochs, ~26k steps best epoch: 7
smooth fsaverage(6) template
template generation
patient-wise mean/median ADNI templates
avg
templates for all ADNI dataset (train/val/test)
patient-wise mean/median TRT templates
avg
templates for TRT dataset
mean/median OASIS templates
avg
templates for OASIS dataset
Ablation Study: Pairing Mode
STATIC V2C-Flow training with smooth fsavg. template 15 epochs, ~56k steps best epoch: 15
FIRST V2C-Long training with FIRST template mode 15 epochs, ~56k steps best epoch: 10
PREV V2C-Long training with PREV template mode 20 epochs ~58k steps best epoch: 20
NXN V2C-Long training with NXN template mode 3 epochs, ~56k steps best epoch: 1
NXN_SORTED V2C-Long training with NXN_SORTED template 5 epochs, ~57k steps best epoch: 4
MEAN V2C-Long training with MEAN template mode 15 epochs, ~56k steps best epoch: 10
★ MEDIAN V2C-Long training with MEDIAN template mode 15 epoch, ~56k steps best epoch: 15
evaluation on ADNI validation set
Ablation Study: Longitudinal Features
MEDIAN V2C-Long training with MEDIAN template mode 45 epochs, ~170k steps best epoch: 45
MEDIAN+Δt V2C-Long training with MEDIAN template mode with time delta features 45 epochs, ~170k steps best epoch: 42
★ MEDIAN+glf V2C-Long training with MEDIAN template mode with template graph features 45 epochs, ~170k steps best epoch: 42
MEDIAN+Δt+glf V2C-Long training with MEDIAN template mode with template graph+time delta features 45 epochs, ~170k steps best epoch: 40
evaluation on ADNI validation set
Best Model Evaluation and Comparison
★ MEDIAN+glf same model as in features ablation study, no additional training.
V2C-Flow smooth fsav. template 45 epochs, ~170k steps best epoch: 40
CorticalFlow++ 3 deformation blocks trained for 70k steps each
V2CC L1+Reg.+CrossEnt. loss 45 epochs, ~170k steps best epoch: 40
FreeSurfer longitudinal pipeline run on non-longitudinal ground truth FreeSurfer meshes
evaluation on ADNI test set
evaluation on TRT
evaluation on OASIS

Figure A.1.: Training and evaluation procedure for template generation, ablation studies, and the final comparison. The colors of the boxes indicate which type of templates were used for training and evaluation. A thin arrow indicates initialization of model weights with the ones from a previous model. An asterisk (★) marks the "winner" model in the respective context.

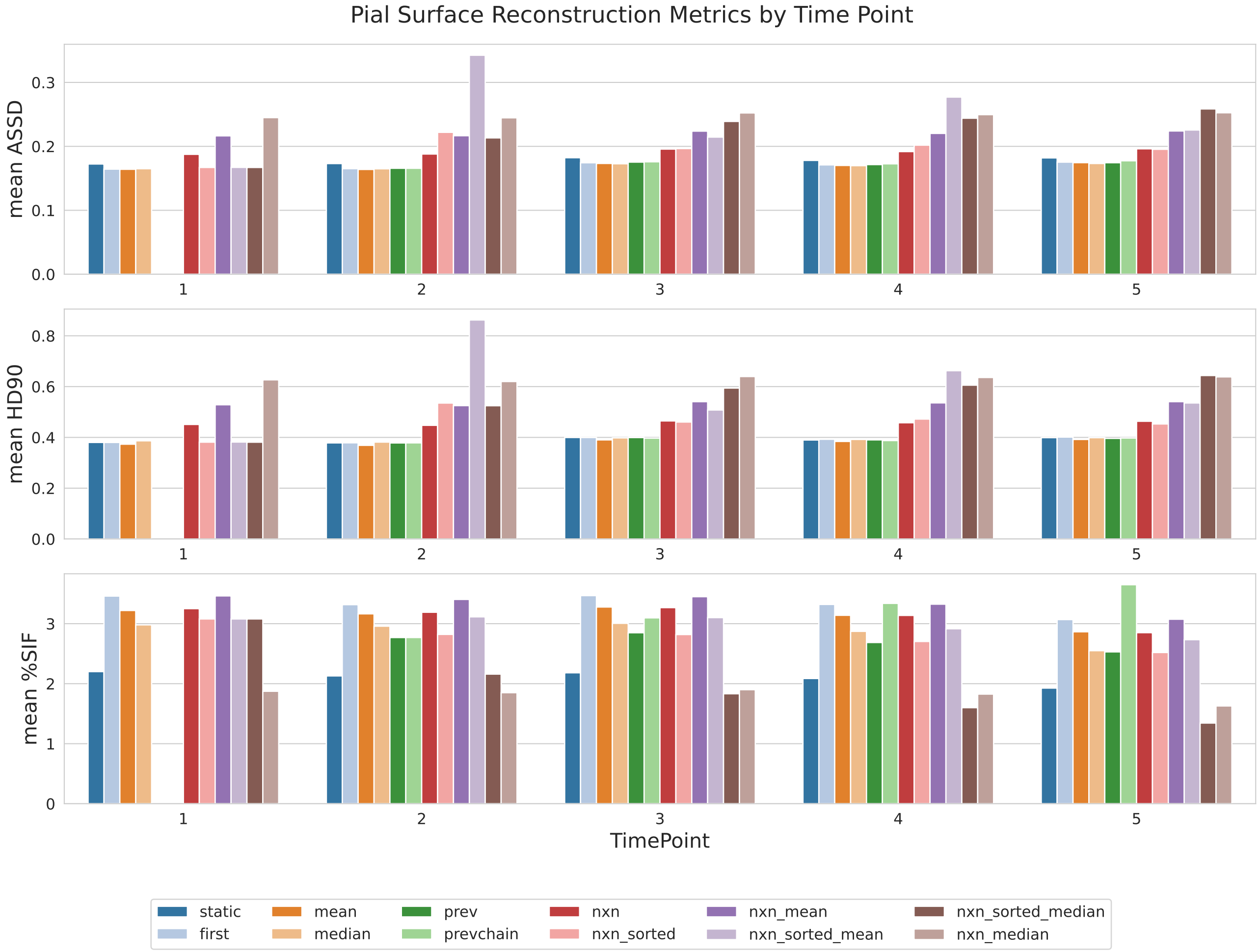


Figure A.2.: Ablation Study 1: Mean reconstruction metrics of the pial surface on the ADNI validation set for all patients with exactly 5 total scans. The PREV and PREV_CHAIN modes do not yield a direct prediction for the first time point.

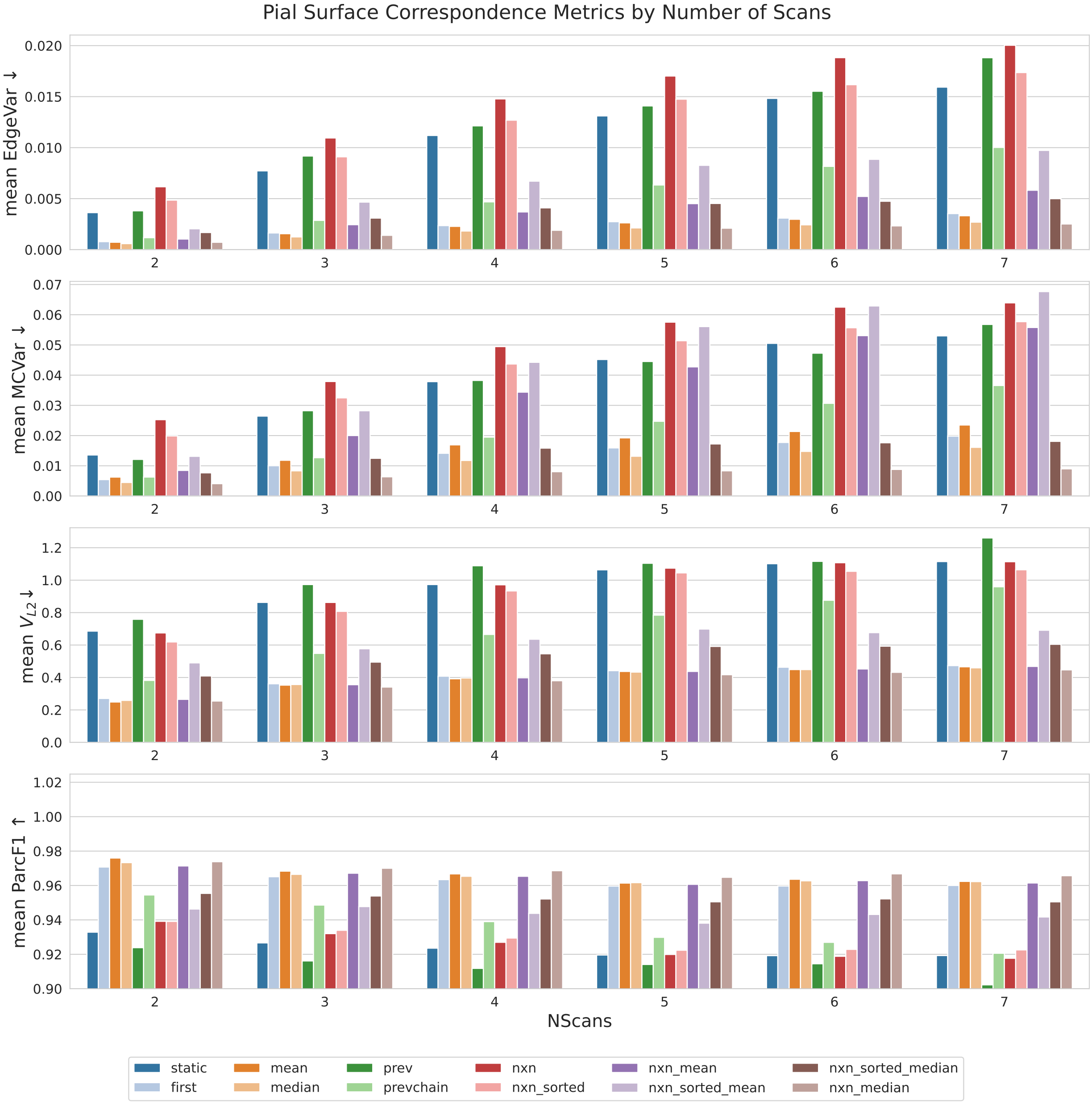


Figure A.3.: Ablation Study 1: Mean correspondence metrics for the pial surface on the ADNI validation set, grouped by number of scans per patient and pairing mode. Patients with more than 7 scans are excluded, as there are less than 10 patients in that category.

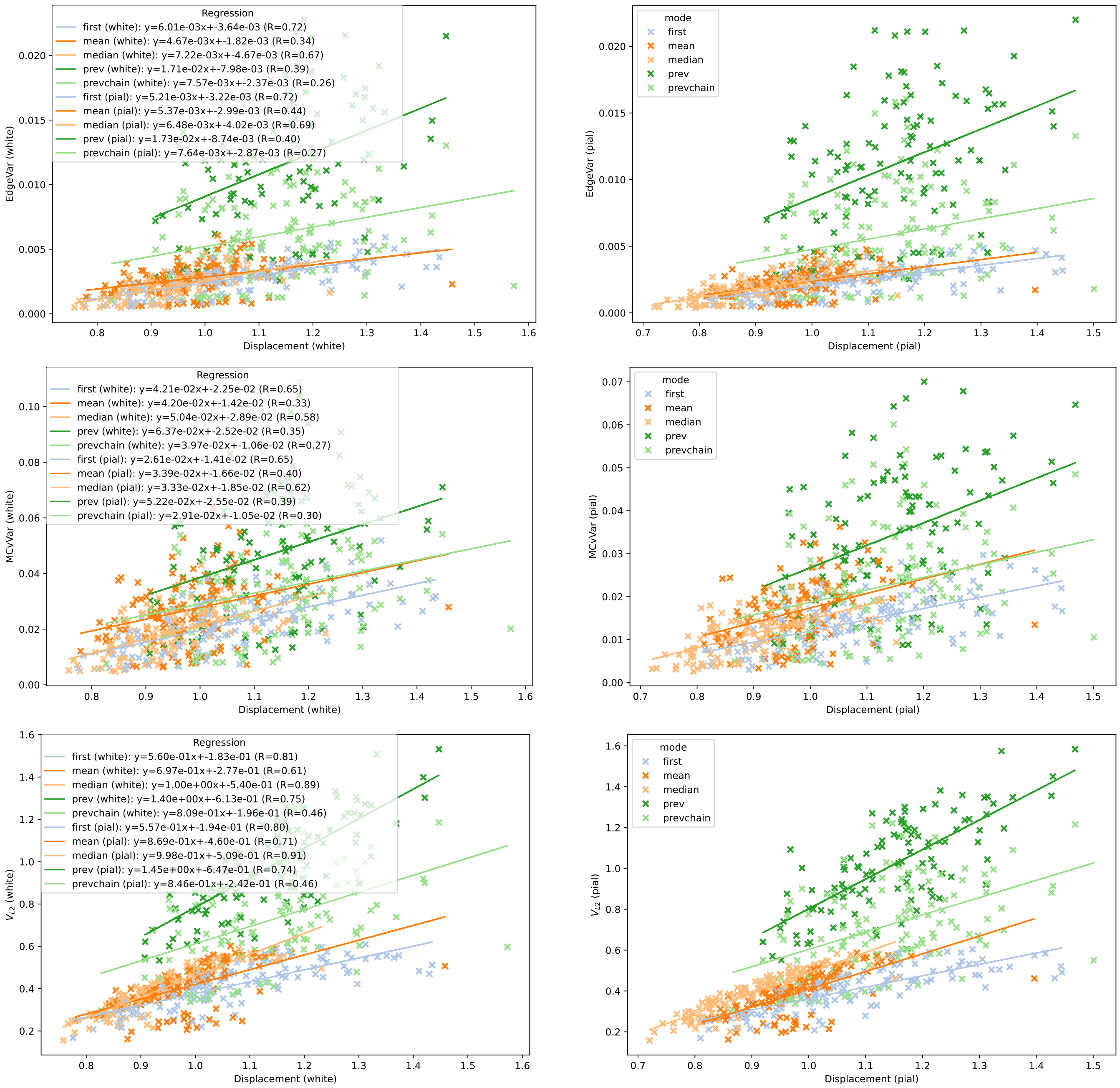


Figure A.4.: Ablation Study 1: Mean displacement plotted against EdgeVar, MeanCurv, $V_{L2}$ metrics for selected modes with linear regression results. Each data point represents a patient in the ADNI validation set. R = Pearson correlation coefficient.

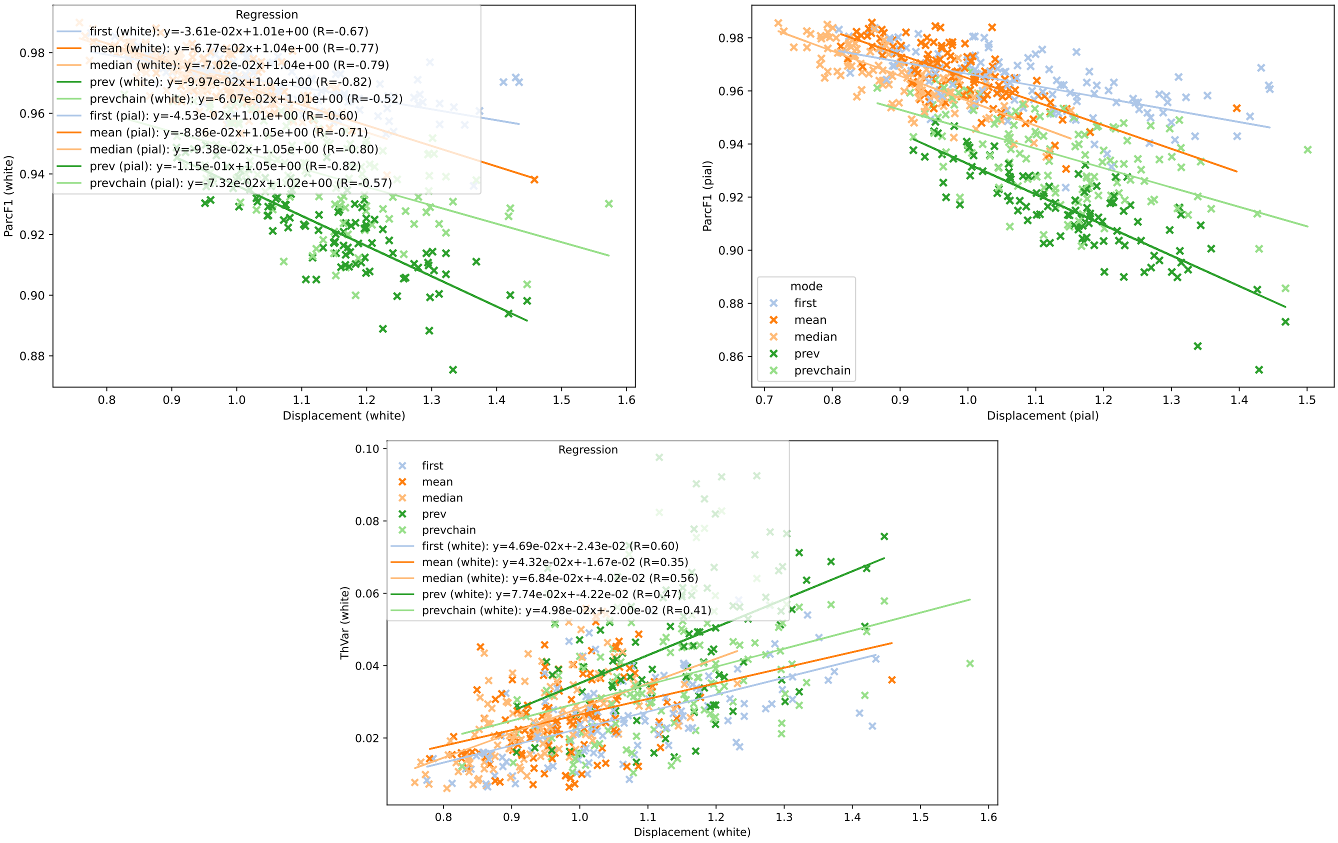


Figure A.5.: Ablation Study 1: Mean displacement plotted against ParcF1 and ThVar metrics for selected modes with linear regression results. Each data point represents a patient in the ADNI validation set. R = Pearson correlation coefficient.

# Abbreviations

**%SIF** Percentage of Self-intersecting Faces

**ADNI** Alzheimer's Disease Neuroimaging Initiative

**AD** Alzheimer's Disease

**ASSD** Average Symmetric Surface Distance

**CSF** Cerebrospinal Fluid

**CSR** Cortical Surface Reconstruction

**CTE** Cortical Thickness Estimation

**EdgeVar** Edge Length Variance

**HD**$_{90}$ 90th Percentile Hausdorff Distance

**MCI** Mild Cognitive Impairment

**MCVar** Mean Curvature Variance

**MRI** Magnetic Resonance Imaging

**NODE** Neural Ordinary Differential Equation

**OASIS** Open Access Series of Imaging Studies

**ParcF1** Longitudinal Parcellation Consistency

**SDF** Signed Distance Function

**TGM** Template Generation Model

**ThVar** Cortical Thickness Variance

**TRT** test-retest

**$\mathbf{V_{L2}}$** Mean Vertex Distance

# List of Figures

# List of Tables

# Bibliography

[1] Alzheimers Disease Neuroimaging Initiative, Y.-L. Wang, W. Chen, W.-J. Cai, H. Hu, W. Xu, Z.-T. Wang, X.-P. Cao, L. Tan, and J.-T. Yu. "Associations of White Matter Hyperintensities with Cognitive Decline: A Longitudinal Study." In: *Journal of Alzheimer's Disease* 73.2 (Jan. 21, 2020). Ed. by Y. Liu, pp. 759–768. ISSN: 13872877, 18758908. DOI: `10.3233/JAD-191005`.

[2] *Automatic Mixed Precision package – torch.amp PyTorch 2.1 documentation*. Accessed on 2023-12-06, Archived at: `https://web.archive.org/web/20231206151310/https://pytorch.org/docs/stable/amp.html`.

[3] B. B. Avants, N. Tustison, G. Song, et al. "Advanced normalization tools (ANTS)." In: *Insight j* 2.365 (2009), pp. 1–35.

[4] J. L. Bernal-Rusiel, D. N. Greve, M. Reuter, B. Fischl, and M. R. Sabuncu. "Statistical Analysis of Longitudinal Neuroimage Data with Linear Mixed Effects Models." In: *NeuroImage* 66 (Feb. 2013), pp. 249–260. ISSN: 10538119. DOI: `10.1016/j.neuroimage.2012.10.065`.

[5] F. Bongratz, A.-M. Rickmann, S. Pölsterl, and C. Wachinger. *Vox2Cortex: Fast Explicit Reconstruction of Cortical Surfaces from 3D MRI Scans with Geometric Deep Neural Networks*. Mar. 18, 2022. arXiv: `2203.09446 [cs]`. URL: `http://arxiv.org/abs/2203.09446` (visited on 05/02/2023). preprint.

[6] F. Bongratz, A.-M. Rickmann, and C. Wachinger. "Neural Deformation Fields for Template-Based Reconstruction of Cortical Surfaces from MRI." In: *Medical Image Analysis (under review)* (2023).

[7] M. Bota, O. Sporns, and L. W. Swanson. "Architecture of the Cerebral Cortical Association Connectome Underlying Cognition." In: *Proceedings of the National Academy of Sciences* 112.16 (Apr. 21, 2015). ISSN: 0027-8424, 1091-6490. DOI: `10.1073/pnas.1504394112`.

[8] S. R. Buss. *3D computer graphics: a mathematical introduction with OpenGL*. Cambridge University Press, 2003.

[9] F. Cardinale, G. Chinnici, M. Bramerio, R. Mai, I. Sartori, M. Cossu, G. Lo Russo, L. Castana, N. Colombo, C. Caborni, E. De Momi, and G. Ferrigno. "Validation of FreeSurfer-Estimated Brain Cortical Thickness: Comparison with Histologic Measurements." In: *Neuroinformatics* 12.4 (Oct. 2014), pp. 535–542. ISSN: 1539-2791, 1559-0089. DOI: `10.1007/s12021-014-9229-2`.

[10] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud. *Neural Ordinary Differential Equations*. Dec. 13, 2019. arXiv: `1806.07366 [cs, stat]`. URL: `http://arxiv.org/abs/1806.07366` (visited on 06/20/2023). preprint.

[11] C.-Y. Chuang, J. Robinson, L. Yen-Chen, A. Torralba, and S. Jegelka. "Debiased Contrastive Learning." In: ().

[12] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia. "MeshLab: an Open-Source Mesh Processing Tool." In: *Eurographics Italian Chapter Conference*. Ed. by V. Scarano, R. D. Chiara, and U. Erra. The Eurographics Association, 2008. ISBN: 978-3-905673-68-5. DOI: `10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136`.

[13] P. Cignoni, A. Muntoni, G. Ranzuglia, and M. Callieri. *MeshLab*. DOI: `10.5281/zenodo.5114037`.

[14] *cortex.polyutils.surface – pycortex documentation*. Accessed on 2023-12-04, Archived at: `https://web.archive.org/web/20231204213644/https://gallantlab.org/pycortex/_modules/cortex/polyutils/surface.html`.

[15] K. Crane, C. Weischedel, and M. Wardetzky. "The Heat Method for Distance Computation." In: *Communications of the ACM* 60.11 (Oct. 24, 2017), pp. 90–99. ISSN: 0001-0782, 1557-7317. DOI: `10.1145/3131280`.

[16] R. S. Cruz, L. Lebrat, P. Bourgeat, C. Fookes, J. Fripp, and O. Salvado. *DeepCSR: A 3D Deep Learning Approach for Cortical Surface Reconstruction*. Oct. 21, 2020. arXiv: `2010.11423 [cs, eess]`. URL: `http://arxiv.org/abs/2010.11423` (visited on 05/05/2023). preprint.

[17] R. S. Cruz, L. Lebrat, D. Fu, P. Bourgeat, J. Fripp, C. Fookes, and O. Salvado. *CorticalFlow$\hat{\{}++\}$: Boosting Cortical Surface Reconstruction Accuracy, Regularity, and Interoperability*. June 14, 2022. arXiv: `2206.06598 [cs, eess]`. URL: `http://arxiv.org/abs/2206.06598` (visited on 05/31/2023). preprint.

[18] A. M. Dale, B. Fischl, and M. I. Sereno. "Cortical Surface-Based Analysis." In: ().

[19] R. S. Desikan, F. Ségonne, B. Fischl, B. T. Quinn, B. C. Dickerson, D. Blacker, R. L. Buckner, A. M. Dale, R. P. Maguire, B. T. Hyman, M. S. Albert, and R. J. Killiany. "An Automated Labeling System for Subdividing the Human Cerebral Cortex on MRI Scans into Gyral Based Regions of Interest." In: *NeuroImage* 31.3 (July 2006), pp. 968–980. ISSN: 10538119. DOI: `10.1016/j.neuroimage.2006.01.021`.

[20] C. Destrieux, B. Fischl, A. Dale, and E. Halgren. "Automatic Parcellation of Human Cortical Gyri and Sulci Using Standard Anatomical Nomenclature." In: *NeuroImage* 53.1 (Oct. 2010), pp. 1–15. ISSN: 10538119. DOI: `10.1016/j.neuroimage.2010.06.010`.

[21] A.-T. Du, N. Schuff, J. H. Kramer, H. J. Rosen, M. L. Gorno-Tempini, K. Rankin, B. L. Miller, and M. W. Weiner. "Different Regional Patterns of Cortical Thinning in Alzheimer's Disease and Frontotemporal Dementia." In: *Brain* 130.4 (Nov. 21, 2006), pp. 1159–1166. ISSN: 0006-8950, 1460-2156. DOI: `10.1093/brain/awm016`.

[22] H. Edelsbrunner and J. L. Harer. *Computational topology: an introduction*. American Mathematical Society, 2022.

[23] A. Eshaghi, F. Prados, W. J. Brownlee, D. R. Altmann, C. Tur, M. J. Cardoso, F. De Angelis, S. H. Van De Pavert, N. Cawley, N. De Stefano, M. L. Stromillo, M. Battaglini, S. Ruggieri, C. Gasperini, M. Filippi, M. A. Rocca, A. Rovira, J. SastreGarriga, H. Vrenken, C. E. Leurs, J. Killestein, L. Pirpamer, C. Enzinger, S. Ourselin, C. A. G. WheelerKingshott, D. Chard, A. J. Thompson, D. C. Alexander, F. Barkhof, O. Ciccarelli, and on behalf of the MAGNIMS study group. "Deep Gray Matter Volume Loss Drives Disability Worsening in Multiple Sclerosis." In: *Annals of Neurology* 83.2 (Feb. 2018), pp. 210–222. ISSN: 0364-5134, 1531-8249. DOI: 10.1002/ana.25145.

[24] A. C. Evans, A. L. Janke, D. L. Collins, and S. Baillet. "Brain Templates and Atlases." In: *NeuroImage* 62.2 (Aug. 2012), pp. 911–922. ISSN: 10538119. DOI: 10.1016/j.neuroimage.2012.01.024.

[25] B. Fischl. "FreeSurfer." In: *NeuroImage* 62.2 (Aug. 2012), pp. 774–781. ISSN: 10538119. DOI: 10.1016/j.neuroimage.2012.01.021.

[26] B. Fischl and A. M. Dale. "Measuring the Thickness of the Human Cerebral Cortex from Magnetic Resonance Images." In: *Proceedings of the National Academy of Sciences* 97.20 (Sept. 26, 2000), pp. 11050–11055. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.200033797.

[27] B. Fischl, M. I. Sereno, R. B. H. Tootell, and A. M. Dale. "Highresolution Intersubject Averaging and a Coordinate System for the Cortical Surface." In: ().

[28] *FreeSurfer Version 7 Release Page*. Accessed on 2023-12-08, Available at https://surfer.nmr.mgh.harvard.edu/fswiki/rel7downloads.

[29] M. Garland and P. S. Heckbert. "Surface Simplification Using Quadric Error Metrics." In: ().

[30] K. Gopinath, C. Desrosiers, and H. Lombaert. "SegRecon: Learning Joint Brain Surface Reconstruction and Segmentation from Images." In: *Medical Image Computing and Computer Assisted Intervention MICCAI 2021*. Ed. by M. De Bruijne, P. C. Cattin, S. Cotin, N. Padoy, S. Speidel, Y. Zheng, and C. Essert. Vol. 12907. Cham: Springer International Publishing, 2021, pp. 650–659. ISBN: 978-3-030-87233-5 978-3-030-87234-2. DOI: 10.1007/978-3-030-87234-2_61.

[31] B. A. Gordon, T. M. Blazey, Y. Su, A. Hari-Raj, A. Dincer, S. Flores, J. Christensen, E. McDade, G. Wang, C. Xiong, N. J. Cairns, J. Hassenstab, D. S. Marcus, A. M. Fagan, C. R. Jack, R. C. Hornbeck, K. L. Paumier, B. M. Ances, S. B. Berman, A. M. Brickman, D. M. Cash, J. P. Chhatwal, S. Correia, S. Förster, N. C. Fox, N. R. Graff-Radford, C. La Fougère, J. Levin, C. L. Masters, M. N. Rossor, S. Salloway, A. J. Saykin, P. R. Schofield, P. M. Thompson, M. M. Weiner, D. M. Holtzman, M. E. Raichle, J. C. Morris, R. J. Bateman, and T. L. S. Benzinger. "Spatial Patterns of Neuroimaging Biomarker Change in Individuals from Families with Au-

tosomal Dominant Alzheimer's Disease: A Longitudinal Study." In: *The Lancet Neurology* 17.3 (Mar. 2018), pp. 241–250. ISSN: 14744422. DOI: 10.1016/S1474-4422(18)30028-0.

[32] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen, and M. S. Hämäläinen. "MEG and EEG Data Analysis with MNE-Python." In: *Frontiers in Neuroscience* 7.267 (2013), pp. 1–13. DOI: 10.3389/fnins.2013.00267.

[33] O. J. Hénaff, A. Srinivas, J. D. Fauw, A. Razavi, C. Doersch, and S. M. A. Eslami. "Data-Efficient Image Recognition with Contrastive Predictive Coding." In: ().

[34] A. Hoopes, J. E. Iglesias, B. Fischl, D. Greve, and A. V. Dalca. "TopoFit: Rapid Reconstruction of Topologically-Correct Cortical Surfaces." In: ().

[35] T. Huynh, S. Kornblith, M. R. Walter, M. Maire, and M. Khademi. *Boosting Contrastive Self-Supervised Learning with False Negative Cancellation*. Jan. 2, 2022. arXiv: 2011.11765 [cs]. URL: http://arxiv.org/abs/2011.11765 (visited on 06/20/2023). preprint.

[36] C. R. Jack, M. A. Bernstein, N. C. Fox, P. Thompson, G. Alexander, D. Harvey, B. Borowski, P. J. Britson, J. L. Whitwell, C. Ward, A. M. Dale, J. P. Felmlee, J. L. Gunter, D. L. Hill, R. Killiany, N. Schuff, S. FoxBosetti, C. Lin, C. Studholme, C. S. DeCarli, Gunnar Krueger, H. A. Ward, G. J. Metzger, K. T. Scott, R. Mallozzi, D. Blezek, J. Levy, J. P. Debbins, A. S. Fleisher, M. Albert, R. Green, G. Bartzokis, G. Glover, J. Mugler, and M. W. Weiner. "The Alzheimer's Disease Neuroimaging Initiative (ADNI): MRI Methods." In: *Journal of Magnetic Resonance Imaging* 27.4 (Apr. 2008), pp. 685–691. ISSN: 1053-1807, 1522-2586. DOI: 10.1002/jmri.21049.

[37] A. Klein and J. Tourville. "101 Labeled Brain Images and a Consistent Human Cortical Labeling Protocol." In: *Frontiers in Neuroscience* 6 (2012). ISSN: 1662-4548. DOI: 10.3389/fnins.2012.00171.

[38] H. Laga. *3D Shape Analysis: Fundamentals, Theory, and Applications*. Hoboken, NJ, USA: Wiley, 2019. 345 pp. ISBN: 978-1-119-40510-8.

[39] S. Lamballais and R. L. Muetzel. "QDECR: A Flexible, Extensible Vertex-Wise Analysis Framework in R." In: *Frontiers in Neuroinformatics* 15 (Apr. 22, 2021), p. 561689. ISSN: 1662-5196. DOI: 10.3389/fninf.2021.561689.

[40] P. J. LaMontagne, T. L. Benzinger, J. C. Morris, S. Keefe, R. Hornbeck, C. Xiong, E. Grant, J. Hassenstab, K. Moulder, A. G. Vlassenko, M. E. Raichle, C. Cruchaga, and D. Marcus. *OASIS-3: Longitudinal Neuroimaging, Clinical, and Cognitive Dataset for Normal Aging and Alzheimer Disease*. preprint. Radiology and Imaging, Dec. 15, 2019. DOI: 10.1101/2019.12.13.19014902.

[41] T. Le, K. Nguyen, S. Sun, K. Han, N. Ho, and X. Xie. *Diffeomorphic Deformation via Sliced Wasserstein Distance Optimization for Cortical Surface Reconstruction*. May 27, 2023. arXiv: 2305.17555 [cs]. URL: http://arxiv.org/abs/2305.17555 (visited on 05/31/2023). preprint.

[42] L. Lebrat, R. S. Cruz, F. de Gournay, and D. Fu. “CorticalFlow: A Diffeomorphic Mesh Deformation Module for Cortical Surface Reconstruction.” In: ().

[43] Y. LeCun, Y. Bengio, and G. Hinton. “Deep Learning.” In: *Nature* 521.7553 (May 28, 2015), pp. 436–444. ISSN: 0028-0836, 1476-4687. DOI: `10.1038/nature14539`.

[44] J. P. Lerch, J. C. Pruessner, A. Zijdenbos, H. Hampel, S. J. Teipel, and A. C. Evans. “Focal Decline of Cortical Thickness in Alzheimer’s Disease Identified by Computational Neuroanatomy.” In: *Cerebral Cortex* 15.7 (July 1, 2005), pp. 995–1001. ISSN: 1460-2199, 1047-3211. DOI: `10.1093/cercor/bhh200`.

[45] T. Lewiner, H. Lopes, A. W. Vieira, and G. Tavares. “Efficient Implementation of Marching Cubes’ Cases with Topological Guarantees.” In: *Journal of Graphics Tools* 8.2 (Jan. 2003), pp. 1–15. ISSN: 1086-7651. DOI: `10.1080/10867651.2003.10487582`.

[46] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez. “A Survey on Deep Learning in Medical Image Analysis.” In: *Medical Image Analysis* 42 (Dec. 2017), pp. 60–88. ISSN: 13618415. DOI: `10.1016/j.media.2017.07.005`.

[47] P. Liu, Z. Wu, G. Li, P.-T. Yap, and D. Shen. “Deep Modeling of Growth Trajectories for Longitudinal Prediction of Missing Infant Cortical Surfaces.” In: *Information Processing in Medical Imaging*. Ed. by A. C. S. Chung, J. C. Gee, P. A. Yushkevich, and S. Bao. Vol. 11492. Cham: Springer International Publishing, 2019, pp. 277–288. ISBN: 978-3-030-20350-4 978-3-030-20351-1. DOI: `10.1007/978-3-030-20351-1_21`.

[48] *LongitudinalProcessing - Free Surfer Wiki*. Accessed on 2023-11-29, Archived at: `https://web.archive.org/web/20231129150215/https://surfer.nmr.mgh.harvard.edu/fswiki/LongitudinalProcessing`. 2021.

[49] *LongitudinalStatistics - Free Surfer Wiki*. Accessed on 2023-11-29, Archived at: `https://web.archive.org/web/20231129163836/https://surfer.nmr.mgh.harvard.edu/fswiki/LongitudinalStatistics`.

[50] W. E. Lorensen and H. E. Cline. “Marching Cubes: A High Resolution 3D Surface Construction Algorithm.” In: ().

[51] I. Loshchilov and F. Hutter. *Decoupled Weight Decay Regularization*. Jan. 4, 2019. arXiv: `1711.05101 [cs, math]`. URL: `http://arxiv.org/abs/1711.05101` (visited on 12/05/2023). preprint.

[52] F. Luesebrink, H. Mattern, R. Yakupov, J. Acosta-Cabronero, M. Ashtarayeh, S. Oeltze-Jafra, and O. Speck. *"Data from: Comprehensive ultrahigh resolution whole brain in vivo MRI dataset as a human phantom"*. OpenNeuro, 2021. DOI: `10.18112/openneuro.ds003563.v1.0.1`.

[53] Q. Ma, L. Li, E. C. Robinson, B. Kainz, D. Rueckert, and A. Alansary. *CortexODE: Learning Cortical Surface Reconstruction by Neural ODEs*. Sept. 10, 2022. arXiv: 2202.08329 [cs, eess]. URL: http://arxiv.org/abs/2202.08329 (visited on 05/02/2023). preprint.

[54] Q. Ma, E. C. Robinson, B. Kainz, D. Rueckert, and A. Alansary. *PialNN: A Fast Deep Learning Framework for Cortical Pial Surface Reconstruction*. Sept. 6, 2021. arXiv: 2109.03693 [eess]. URL: http://arxiv.org/abs/2109.03693 (visited on 07/13/2023). preprint.

[55] J. Maclaren, Z. Han, S. B. Vos, N. Fischbein, and R. Bammer. "Reliability of Brain Volume Measurements: A Test-Retest Dataset." In: *Scientific Data* 1.1 (Oct. 14, 2014), p. 140037. ISSN: 2052-4463. DOI: 10.1038/sdata.2014.37.

[56] S. Magon, L. Gaetano, M. M. Chakravarty, J. P. Lerch, Y. Naegelin, C. Stippich, L. Kappos, E.-W. Radue, and T. Sprenger. "White Matter Lesion Filling Improves the Accuracy of Cortical Thickness Measurements in Multiple Sclerosis Patients: A Longitudinal Study." In: *BMC Neuroscience* 15.1 (Dec. 2014), p. 106. ISSN: 1471-2202. DOI: 10.1186/1471-2202-15-106.

[57] D. S. Marcus, A. F. Fotenos, J. G. Csernansky, J. C. Morris, and R. L. Buckner. "Open Access Series of Imaging Studies: Longitudinal MRI Data in Nondemented and Demented Older Adults." In: *Journal of Cognitive Neuroscience* 22.12 (Dec. 1, 2010), pp. 2677–2684. ISSN: 0898-929X, 1530-8898. DOI: 10.1162/jocn.2009.21407.

[58] D. S. Marcus, T. H. Wang, J. Parker, J. G. Csernansky, J. C. Morris, and R. L. Buckner. "Open Access Series of Imaging Studies (OASIS): Cross-sectional MRI Data in Young, Middle Aged, Nondemented, and Demented Older Adults." In: *Journal of Cognitive Neuroscience* 19.9 (Sept. 1, 2007), pp. 1498–1507. ISSN: 0898-929X, 1530-8898. DOI: 10.1162/jocn.2007.19.9.1498.

[59] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger. *Occupancy Networks: Learning 3D Reconstruction in Function Space*. Apr. 30, 2019. arXiv: 1812.03828 [cs]. URL: http://arxiv.org/abs/1812.03828 (visited on 12/03/2023). preprint.

[60] *MeshFix Source Code Repository – Github*. Accessed on 2023-12-08, Available at https://github.com/MarcoAttene/MeshFix-V2.1.

[61] M. Modat. "Global image registration using a symmetric block-matching approach." In: *Journal of Medical Imaging* 1.2 (2014), pp. 024003–024003. DOI: 10.1117/1.JMI.1.2.024003.

[62] M. Modat, D. M. Cash, P. Daga, G. P. Winston, J. S. Duncan, and S. Ourselin. "Global Image Registration Using a Symmetric Block-Matching Approach." In: *Journal of Medical Imaging* 1.2 (Sept. 19, 2014), p. 024003. ISSN: 2329-4302. DOI: 10.1117/1.JMI.1.2.024003.

[63] A. Muntoni and P. Cignoni. *PyMeshLab*. Jan. 2021. DOI: 10.5281/zenodo.4438750.

[64] E. A. Nelson, N. V. Kraguljac, D. M. White, R. D. Jindal, A. L. Shin, and A. C. Lahti. "A Prospective Longitudinal Investigation of Cortical Thickness and Gyrification in Schizophrenia." In: *The Canadian Journal of Psychiatry* 65.6 (June 2020), pp. 381–391. ISSN: 0706-7437, 1497-0015. DOI: 10.1177/0706743720904598.

[65] J. Ouyang, Q. Zhao, E. Adeli, E. V. Sullivan, A. Pfefferbaum, G. Zaharchuk, and K. M. Pohl. *Self-Supervised Longitudinal Neighbourhood Embedding*. June 17, 2021. arXiv: 2103.03840 [cs]. URL: http://arxiv.org/abs/2103.03840 (visited on 06/20/2023). preprint.

[66] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas. "Functional Maps: A Flexible Representation of Maps Between Shapes." In: ().

[67] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. Dec. 3, 2019. arXiv: 1912.01703 [cs, stat]. URL: http://arxiv.org/abs/1912.01703 (visited on 12/06/2023). preprint.

[68] G. Peyré and M. Cuturi. "Computational Optimal Transport: With Applications to Data Science." In: *Foundations and Trendső in Machine Learning* 11.5-6 (2019), pp. 355–607. ISSN: 1935-8237, 1935-8245. DOI: 10.1561/2200000073.

[69] O. Querbes, F. Aubry, J. Pariente, J.-A. Lotterie, J.-F. Démonet, V. Duret, M. Puel, I. Berry, J.-C. Fort, P. Celsis, and The Alzheimer's Disease Neuroimaging Initiative. "Early Diagnosis of Alzheimer's Disease Using Cortical Thickness: Impact of Cognitive Reserve." In: *Brain* 132.8 (Aug. 1, 2009), pp. 2036–2047. ISSN: 1460-2156, 0006-8950. DOI: 10.1093/brain/awp105.

[70] N. Ravi, J. Reizenstein, D. Novotny, T. Gordon, W.-Y. Lo, J. Johnson, and G. Gkioxari. *Accelerating 3D Deep Learning with PyTorch3D*. July 16, 2020. arXiv: 2007.08501 [cs]. URL: http://arxiv.org/abs/2007.08501 (visited on 12/06/2023). preprint.

[71] *recon-all - Free Surfer Wiki*. Accessed on 2023-11-29, Archived at: https://web.archive.org/web/20231129150343/https://surfer.nmr.mgh.harvard.edu/fswiki/recon-all. 2021.

[72] *ReconAllRunTimes - Free Surfer Wiki*. Accessed on 2023-11-24, Archived at: https://web.archive.org/web/20231124115915/https://surfer.nmr.mgh.harvard.edu/fswiki/ReconAllRunTimes. 2008.

[73] J. Ren, Q. Hu, W. Wang, W. Zhang, C. Hubbard, P. Zhang, N. An, Y. Zhou, L. Dahmani, D. Wang, X. Fu, Z. Sun, Y. Wang, R. Wang, L. Li, and H. Liu. "Fast Cortical Surface Reconstruction from MRI Using Deep Learning." In: *Brain Informatics* 9 (Dec. 1, 2022). DOI: 10.1186/s40708-022-00155-7.

[74] M. Reuter and B. Fischl. "Avoiding Asymmetry-Induced Bias in Longitudinal Image Processing." In: *NeuroImage* 57.1 (July 2011), pp. 19–21. ISSN: 10538119. DOI: 10.1016/j.neuroimage.2011.02.076.

[75] M. Reuter, H. D. Rosas, and B. Fischl. "Highly Accurate Inverse Consistent Registration: A Robust Approach." In: *NeuroImage* 53.4 (Dec. 2010), pp. 1181–1196. ISSN: 10538119. DOI: 10.1016/j.neuroimage.2010.07.020.

[76] M. Reuter, N. J. Schmansky, H. D. Rosas, and B. Fischl. "Within-Subject Template Estimation for Unbiased Longitudinal Image Analysis." In: *NeuroImage* 61.4 (July 2012), pp. 1402–1418. ISSN: 10538119. DOI: 10.1016/j.neuroimage.2012.02.084.

[77] A.-M. Rickmann, F. Bongratz, and C. Wachinger. "Vertex Correspondence in Cortical Surface Reconstruction." In: ().

[78] E. C. Robinson, K. Garcia, M. F. Glasser, Z. Chen, T. S. Coalson, A. Makropoulos, J. Bozek, R. Wright, A. Schuh, M. Webster, J. Hutter, A. Price, L. Cordero Grande, E. Hughes, N. Tusor, P. V. Bayly, D. C. Van Essen, S. M. Smith, A. D. Edwards, J. Hajnal, M. Jenkinson, B. Glocker, and D. Rueckert. "Multimodal Surface Matching with Higher-Order Smoothness Constraints." In: *NeuroImage* 167 (Feb. 2018), pp. 453–465. ISSN: 10538119. DOI: 10.1016/j.neuroimage.2017.10.037.

[79] O. Ronneberger, P. Fischer, and T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation." In: *Medical Image Computing and Computer-Assisted Intervention MICCAI 2015*. Ed. by N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi. Vol. 9351. Cham: Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24573-7 978-3-319-24574-4. DOI: 10.1007/978-3-319-24574-4_28.

[80] R. I. Scahill, C. Frost, R. Jenkins, J. L. Whitwell, M. N. Rossor, and N. C. Fox. "A Longitudinal Study of Brain Volume Changes in Normal Aging Using Serial Registered Magnetic Resonance Imaging." In: *Archives of Neurology* 60.7 (July 1, 2003), p. 989. ISSN: 0003-9942. DOI: 10.1001/archneur.60.7.989.

[81] M. E. Shaw, P. S. Sachdev, K. J. Anstey, and N. Cherbuin. "Age-Related Cortical Thinning in Cognitively Healthy Individuals in Their 60s: The PATH Through Life Study." In: *Neurobiology of Aging* 39 (Mar. 2016), pp. 202–209. ISSN: 01974580. DOI: 10.1016/j.neurobiolaging.2015.12.009.

[82] L. N. Smith. "Cyclical Learning Rates for Training Neural Networks." In: *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). Santa Rosa, CA, USA: IEEE, Mar. 2017, pp. 464–472. ISBN: 978-1-5090-4822-9. DOI: 10.1109/WACV.2017.58.

[83] *source code of CorticalFlow++ – Bitbucket page of the Commonwealth Scientific and Industrial Research Organisation*. Accessed 2023-12-05, Available at https://bitbucket.csiro.au/projects/CRCPMAX/repos/corticalflow/browse.

[84] *source code of `sklearn.metrics.f1_score` - Github*. Accessed on 2023-12-04, Archived at: `https://web.archive.org/web/20231204213825/https://github.com/scikit-learn/scikit-learn/blob/3f89022fa/sklearn/metrics/_classification.py`.

[85] R. A. Stelzmann, H. Norman Schnitzlein, and F. Reed Murtagh. "An English Translation of Alzheimer's 1907 Paper, Über Eine Eigenartige Erkankung Der Hirnrinde." In: *Clinical Anatomy* 8.6 (Jan. 1995), pp. 429–431. ISSN: 0897-3806, 1098-2353. DOI: `10.1002/ca.980080612`.

[86] *`subjects/fsaverage/scripts/make_average_surface.log` file within FreeSurfer distribution*. File accessed with FreeSurfer 7.2, which can be downloaded at `https://surfer.nmr.mgh.harvard.edu/fswiki/rel7downloads`.

[87] M. A. Suliman, L. Z. J. Williams, A. Fawaz, and E. C. Robinson. "A Deep-Discrete Learning Framework for Spherical Surface Registration." In: *Medical Image Computing and Computer Assisted Intervention  MICCAI 2022*. Ed. by L. Wang, Q. Dou, P. T. Fletcher, S. Speidel, and S. Li. Vol. 13436. Cham: Springer Nature Switzerland, 2022, pp. 119–129. ISBN: 978-3-031-16445-3 978-3-031-16446-0. DOI: `10.1007/978-3-031-16446-0_12`.

[88] S. Sun, K. Han, D. Kong, H. Tang, X. Yan, and X. Xie. *Topology-Preserving Shape Reconstruction and Registration via Neural Diffeomorphic Flow*. Mar. 21, 2022. arXiv: `2203.08652 [cs]`. URL: `http://arxiv.org/abs/2203.08652` (visited on 05/02/2023). preprint.

[89] M. Thambisetty, J. Wan, A. Carass, Y. An, J. L. Prince, and S. M. Resnick. "Longitudinal Changes in Cortical Thickness Associated with Normal Aging." In: *NeuroImage* 52.4 (Oct. 2010), pp. 1215–1223. ISSN: 10538119. DOI: `10.1016/j.neuroimage.2010.04.258`.

[90] *TopoFit, readme page*. Accessed on 2023-12-04, Archived at: `https://web.archive.org/web/20231203235105/https://github.com/ahoopes/topofit/blob/main/readme.md`.

[91] G. Upton and I. Cook. *A dictionary of statistics 3e*. Oxford quick reference, 2014.

[92] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang. "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images." In: *Computer Vision  ECCV 2018*. Ed. by V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss. Vol. 11215. Cham: Springer International Publishing, 2018, pp. 55–71. ISBN: 978-3-030-01251-9 978-3-030-01252-6. DOI: `10.1007/978-3-030-01252-6_4`.

[93] E. W. Weisstein. *Polyhedral Formula*. MathWorld–A Wolfram Web Resource. Accessed on 2023-12-12, Archived at: `https://web.archive.org/web/20231212160243/https://mathworld.wolfram.com/SchlaefliSymbol.html`.

[94] E. W. Weisstein. *Schläfli Symbol*. MathWorld–A Wolfram Web Resource. Accessed on 2023-12-12, Archived at: `https://web.archive.org/web/20231212160243/https://mathworld.wolfram.com/SchlaefliSymbol.html`.

[95] U. Wickramasinghe, E. Remelli, G. Knott, and P. Fua. "Voxel2Mesh: 3D Mesh Model Generation from Volumetric Data." In: *Medical Image Computing and Computer Assisted Intervention MICCAI 2020*. Ed. by A. L. Martel, P. Abolmaesumi, D. Stoyanov, D. Mateus, M. A. Zuluaga, S. K. Zhou, D. Racoceanu, and L. Joskowicz. Vol. 12264. Cham: Springer International Publishing, 2020, pp. 299–308. ISBN: 978-3-030-59718-4 978-3-030-59719-1. DOI: `10.1007/978-3-030-59719-1_30`.

[96] J. Wu, G. H. Ngo, D. Greve, J. Li, T. He, B. Fischl, S. B. Eickhoff, and B. T. Yeo. "Accurate Nonlinear Mapping between MNI Volumetric and FreeSurfer Surface Coordinate Systems." In: *Human Brain Mapping* 39.9 (Sept. 2018), pp. 3793–3808. ISSN: 1065-9471, 1097-0193. DOI: `10.1002/hbm.24213`.

[97] Q. Xu, W. Wang, D. Ceylan, R. Mech, and U. Neumann. *DISN: Deep Implicit Surface Network for High-quality Single-view 3D Reconstruction*. Dec. 8, 2021. arXiv: `1905.10711 [cs]`. URL: `http://arxiv.org/abs/1905.10711` (visited on 12/03/2023). preprint.

[98] A. B. Yoo, M. A. Jette, and M. Grondona. "SLURM: Simple Linux Utility for Resource Management." In: *Job Scheduling Strategies for Parallel Processing*. Ed. by D. Feitelson, L. Rudolph, and U. Schwiegelshohn. Red. by G. Goos, J. Hartmanis, and J. Van Leeuwen. Vol. 2862. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 44–60. ISBN: 978-3-540-20405-3 978-3-540-39727-4. DOI: `10.1007/10968987_3`.

[99] S. Zagoruyko and N. Komodakis. "Learning to Compare Image Patches via Convolutional Neural Networks." In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, MA, USA: IEEE, June 2015, pp. 4353–4361. ISBN: 978-1-4673-6964-0. DOI: `10.1109/CVPR.2015.7299064`.

[100] D. Zhang, Z. Wu, X. Wang, C. Lv, and M. Zhou. "A Harmonic Wave Kernel Signature for Three-Dimensional Skull Similarity Measurements." In: *2019 International Conference on Cyberworlds (CW)*. 2019 International Conference on Cyberworlds (CW). Kyoto, Japan: IEEE, Oct. 2019, pp. 77–84. ISBN: 978-1-72812-297-7. DOI: `10.1109/CW.2019.00021`.

[101] F. Zhao, Z. Wu, and G. Li. "Deep Learning in Cortical Surface-Based Neuroimage Analysis: A Systematic Review." In: *Intelligent Medicine* 3.1 (Feb. 2023), pp. 46–58. ISSN: 26671026. DOI: `10.1016/j.imed.2022.06.002`.

[102] F. Zhao, Z. Wu, F. Wang, W. Lin, S. Xia, D. Shen, L. Wang, and G. Li. "S3Reg: Superfast Spherical Surface Registration Based on Deep Learning." In: *IEEE Transactions on Medical Imaging* 40.8 (Aug. 2021), pp. 1964–1976. ISSN: 0278-0062, 1558-254X. DOI: `10.1109/TMI.2021.3069645`.

[103] H. Zheng, H. Li, and Y. Fan. *SurfNN: Joint Reconstruction of Multiple Cortical Surfaces from Magnetic Resonance Images*. Mar. 6, 2023. arXiv: `2303.02922 [cs,`

eess]. URL: http://arxiv.org/abs/2303.02922 (visited on 05/02/2023). preprint.