CorticalFlow: A Diffeomorphic Mesh Transformer for Cortical Surface Reconstruction

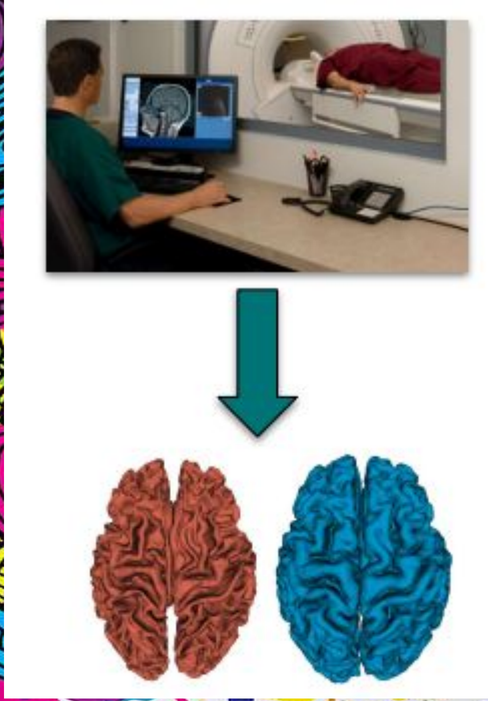
Authors: Leo Lebrat, Rodrigo Santa Cruz, Frederic de Gournay, Darren Fu, Pierrick Bourgeat, Jurgen Fripp, Clinton Fookes, and Olivier Salvado

Presentation by: William Ashbee

Cortical Surface Reconstruction From MRI (CSR)

Given: Structural Image (mri)

Find: Surface model (mesh, spline, etc.)



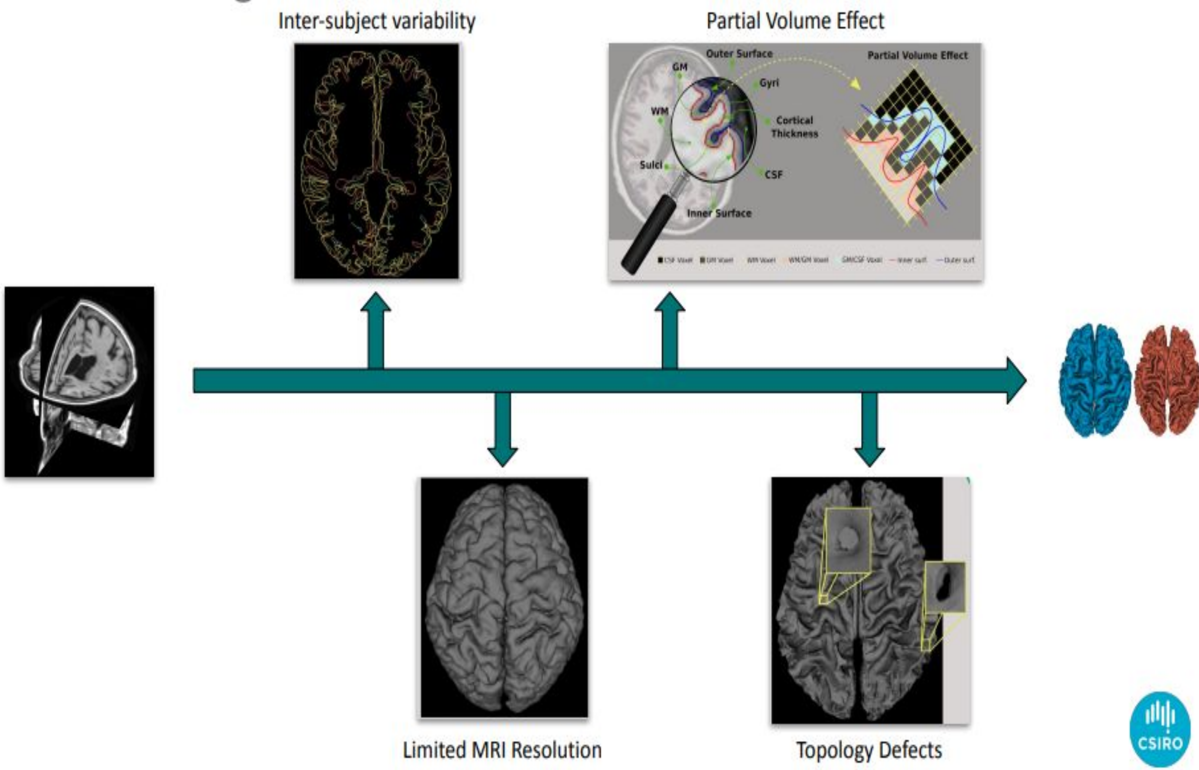
Cortical Surface Reconstruction Challenges

Inter Subject Variability: brain folds as unique as a **fingerprint**

Partial Volume effects: **tight corners** can easily be missed by surface algorithms

Resolution: MRIs are never high **enough resolution** or free of **artifacts**

Topological defects: e.g. self intersections



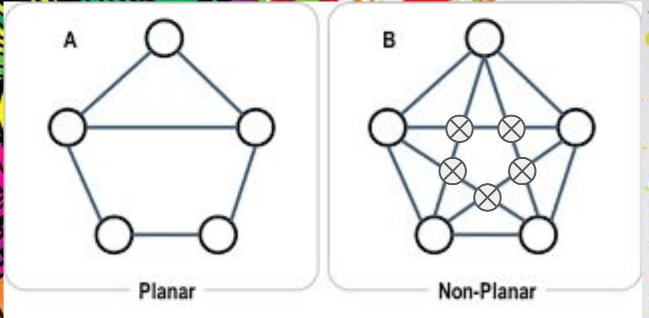
Challenges

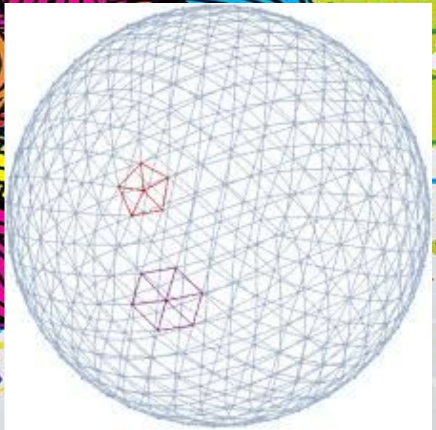
Self intersections:

Meshes can’t be said to be **planar graphs**, but they do strive to avoid **self intersections**

Topology correction algorithms in post processing are **expensive**

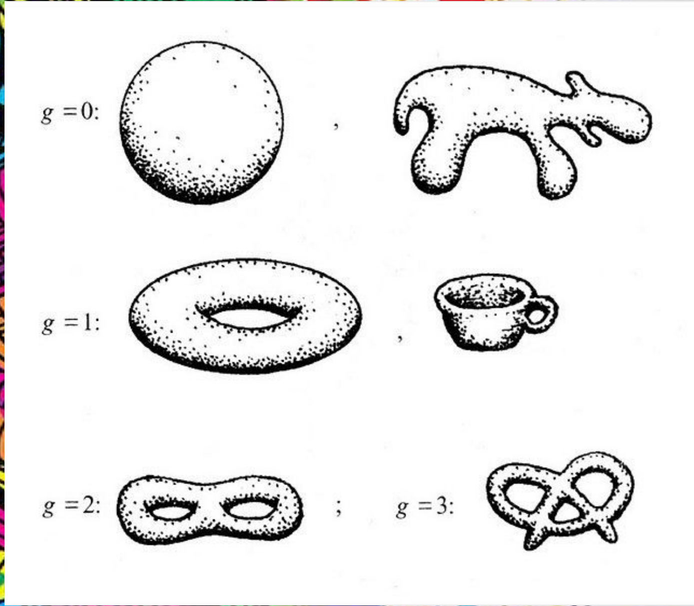
Cheaper to have **theoretical guarantees** during building of mesh





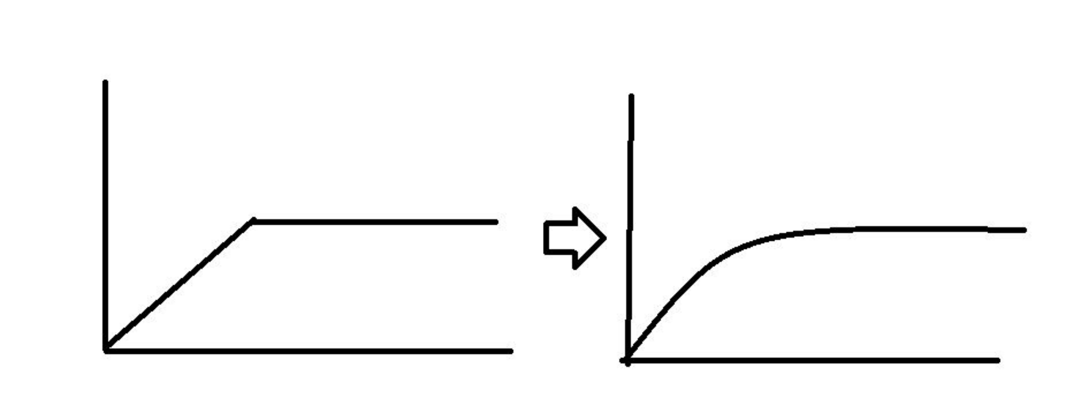
Related work

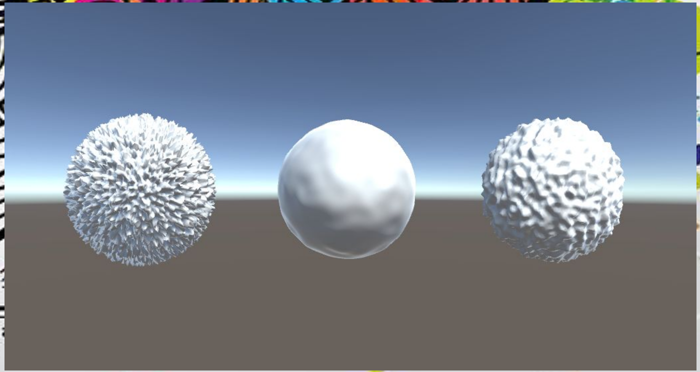
DeepCSR first predicts **implicit surface functions** and then employs an **iso-surface extraction** method along with a topology correction algorithm to obtain **genus zero** surfaces without handles or holes



Related work

Voxel2Mesh extends the vertex-wise template deformation approach of Wang et al. by optimizing several **mesh-smoothing penalty functions**





Related work

NMF builds an invertible mapping that enforces topology conservation upon the resolution of an Ordinary Differential Equation (ODE) through a sequence of residual blocks called **Neural Ordinary Differential Equation** (NODE)

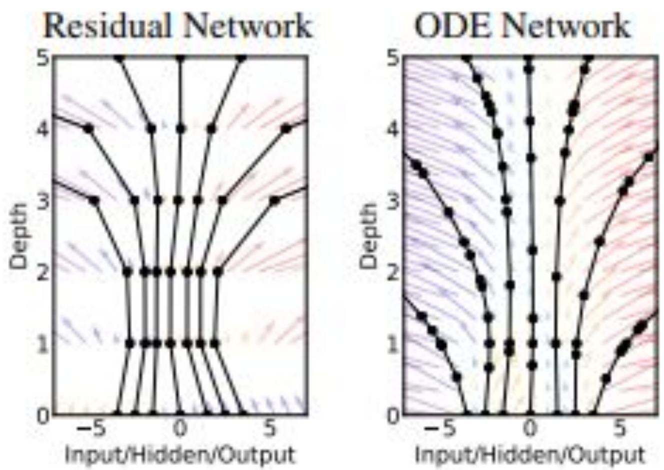


Figure 1: *Left:* A Residual network defines a discrete sequence of finite transformations. *Right:* A ODE network defines a vector field, which continuously transforms the state. *Both:* Circles represent evaluation locations.

Related work (Limitations)

The topology correction algorithm employed by DeepCSR is computationally **expensive** and is **blind to anatomy**.

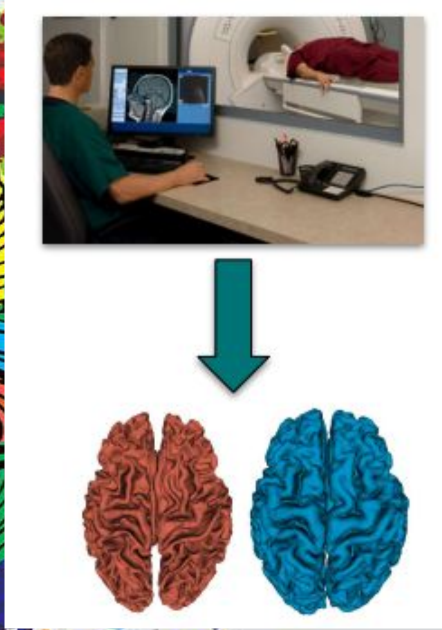
Voxel2Mesh and NMF rely on time-demanding and vertex dependent building blocks such as graph convolution that **do not scale up well** as the number of **vertices** in the template mesh increases to accommodate complex shapes

Method

CorticalFlow (CF) is a multi-level deep learning architecture composed of several **Diffeomorphic Mesh Deformation** (DMD) modules.

It takes as input a 3-dimensional **Magnetic Resonance Image** (MRI) of a patient brain and a template.

CorticalFlow outputs the surface representation of an anatomical substructure by **composing stackable diffeomorphic deformations** generated by DMD modules

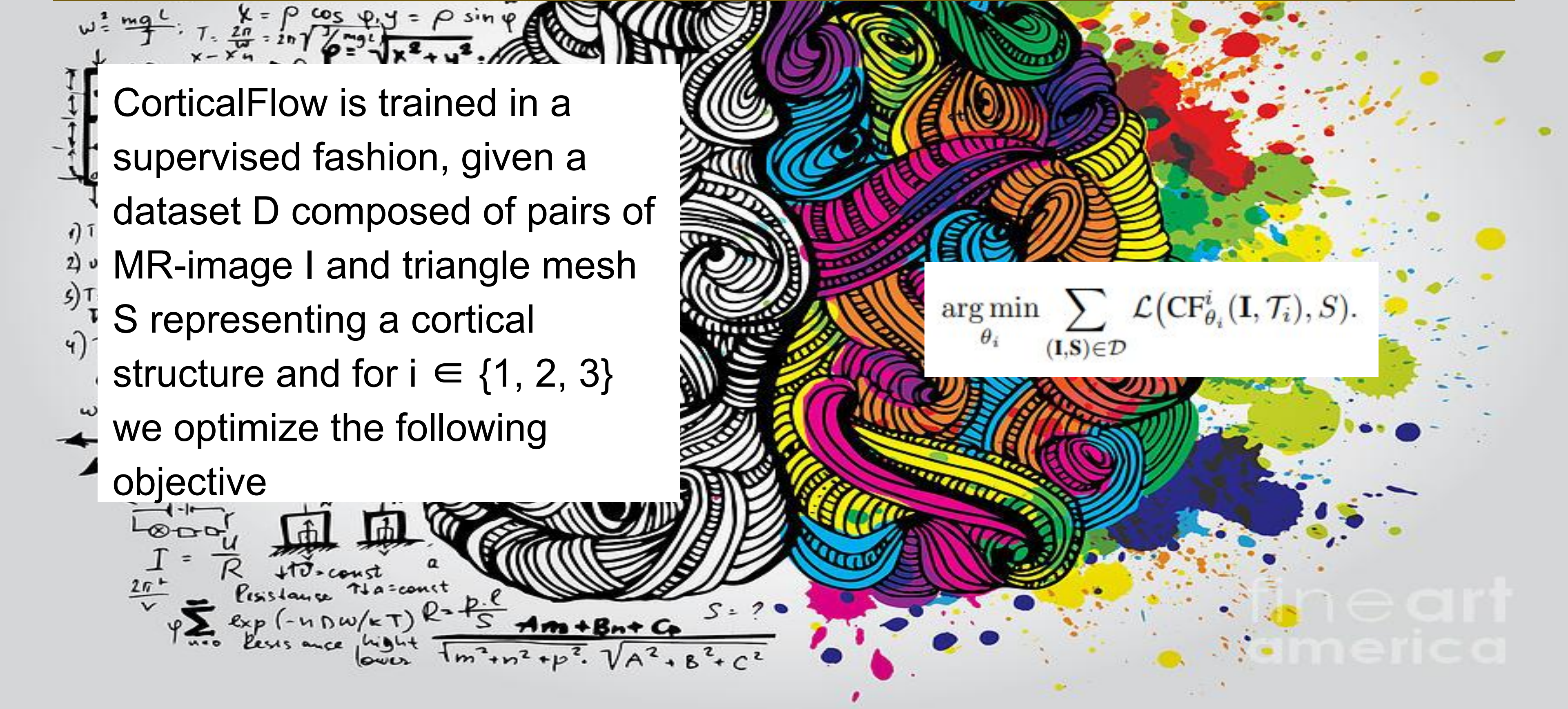


$$\begin{aligned} \mathrm{CF}^{1}_{\theta_1}(\mathbf{I}, \mathcal{T}_1) &= \mathrm{DMD}(\mathrm{UNet}^{1}_{\theta_1}(\mathbf{I}), \mathcal{T}_1)) \\ \mathrm{CF}^{i+1}_{\theta_{i+1}}(\mathbf{I}, \mathcal{T}_{i+1}) &= \mathrm{DMD}(\mathrm{UNet}^{i+1}_{\theta_{i+1}}(\mathbf{U}_1^\frown \cdots \mathbf{U}_i^\frown \mathbf{I}), \mathrm{CF}_i(\mathbf{I}, \mathcal{T}_{i+1})) \quad \text{for } i \geq 1, \end{aligned} \tag{2}$$

Method (objective)

CorticalFlow is trained in a supervised fashion, given a dataset D composed of pairs of MR-image I and triangle mesh S representing a cortical structure and for i ∈ {1, 2, 3} we optimize the following objective

$$\arg\min_{\theta_i} \sum_{(\mathbf{I},\mathbf{S})\in\mathcal{D}} \mathcal{L}(\mathrm{CF}^i_{\theta_i}(\mathbf{I}, \mathcal{T}_i), S).$$



Method (Diffeomorphic deformation module--DMD)

Theorems guarantee surface created will not have self intersections.

The DMD solves a flow ordinary differential equation for each vertex Vi and assigns the result to Vi

$$\frac{d\Phi(s;\mathbf{x})}{ds} = v\left(\Phi(s;\mathbf{x})\right),\ \textit{with}\ \Phi(0;\mathbf{x}) = \mathbf{x}.$$

Algorithm 1 DMD module pseudo-code

Input: $\mathrm{U} \in \mathbb{R}^{H \times W \times D \times 3}$ ▷ Discrete flow
Input: $\mathcal{T}$ with vertices $(V_i)_{i \in 1..m}$ ▷ Triangle mesh
Input: $n \in \mathbb{N}^*$ ▷ Number of integration steps
Output: Updated $(V_i)_{i \in 1..m}$
$h \leftarrow \frac{1}{n}$
Ensure: $h < \frac{1}{L}$
  for $i \in [\![1, m]\!]$ **do**
    for $j \in [\![1, n]\!]$ **do**
      $V_i \leftarrow \Psi(h, V_i)$
    end for
  end for

Diffeomorphic Mesh Deformation (DMD)

Tractable framework for computing a diffeomorphic mapping $\mathbf{\Phi}$ for each surface mesh vertex by solving the **flow ODE**,

$$\frac{d\Phi(s;\mathbf{x})}{ds} = v\left(\Phi(s;\mathbf{x})\right), \text{with } \Phi(0;\mathbf{x}) = x$$

using the iterative approximation method,

$$V_{k+1}^{i} = V_{k}^{i} + hv(V_{k}^{i}), \text{with } h = \frac{1}{N}$$

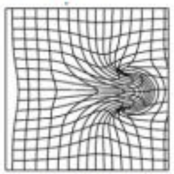
provided by the forward Euler method.

- Retains the initial mesh topology without producing self-intersections.
- We also provide sufficient and comprehensible conditions for meeting the diffeomorphic properties of these transformations.

Related work:

→ Scaling & Squaring in DL Registration [1]:

$$\Phi^{(1/8)} = x + v/8$$
$$\Phi^{(1/4)} = \Phi^{(1/8)} \circ \Phi^{(1/8)}$$
$$\Phi^{(1/2)} = \Phi^{(1/4)} \circ \Phi^{(1/4)}$$
$$\Phi^{(1)} = \Phi^{(1/2)} \circ \Phi^{(1/2)}$$



Voxel-wise integration
$|I| \gg |V|$

→ Neural ODEs [2]:

$$\Phi(x) = x + \int_0^1 f_\theta(x, I)dt$$

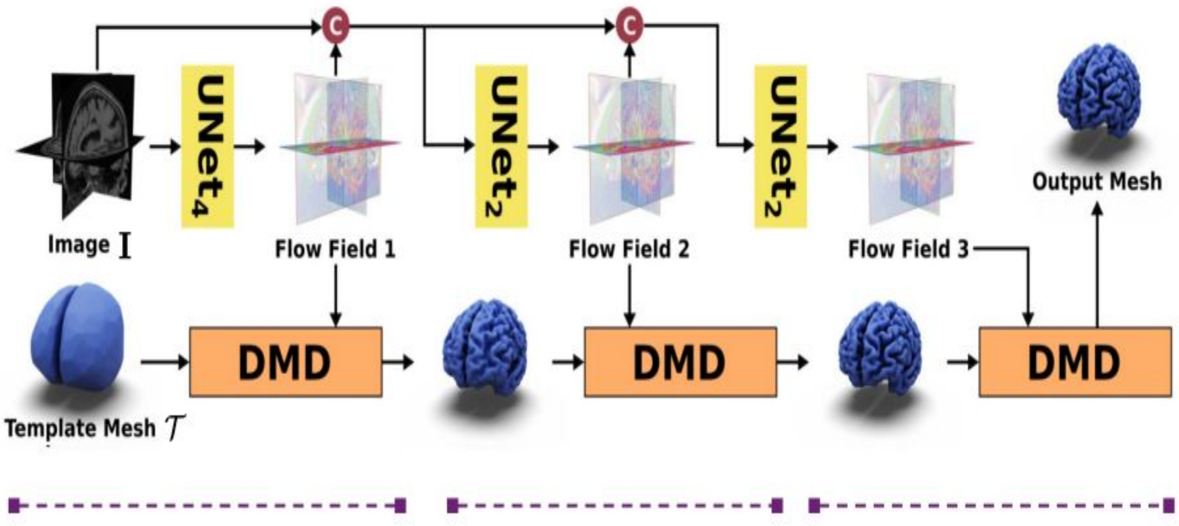
Neural Network with per vertex image feature extractor

1] - Dalca, Adrian V., et al. "Unsupervised learning of probabilistic diffeomorphic registration for images and surfaces." *Medical image analysis* 57 (2019): 226-236.
2] - Gupta and Chandraker. "Neural mesh flow: 3d manifold mesh generation via diffeomorphic flows." *In Advances in Neural Information Processing Systems*, 2020.

Cortical Flow architecture

Theorems guarantee surface created will not have self intersections.

The DMD solves a flow ordinary differential equation for each vertex Vi and assigns the result to Vi



$$\mathrm{CF}^{1}_{\theta_1}(\mathbf{I}, \mathcal{T}_1) = \mathrm{DMD}\left(\mathrm{UNet}^{1}_{\theta_1}(\mathbf{I}), \mathcal{T}_1\right)$$

$$\mathrm{CF}^{i+1}_{\theta_{i+1}}(\mathbf{I}, \mathcal{T}_{i+1}) = \mathrm{DMD}\left(\mathrm{UNet}^{i+1}_{\theta_{i+1}}(\mathbf{U}_1^{\frown} \cdots \mathbf{U}_i^{\frown}\mathbf{I}), \mathrm{CF}^{i}_{\theta_i}(\mathbf{I}, \mathcal{T}_{i+1})\right)$$

Cortical Flow architecture

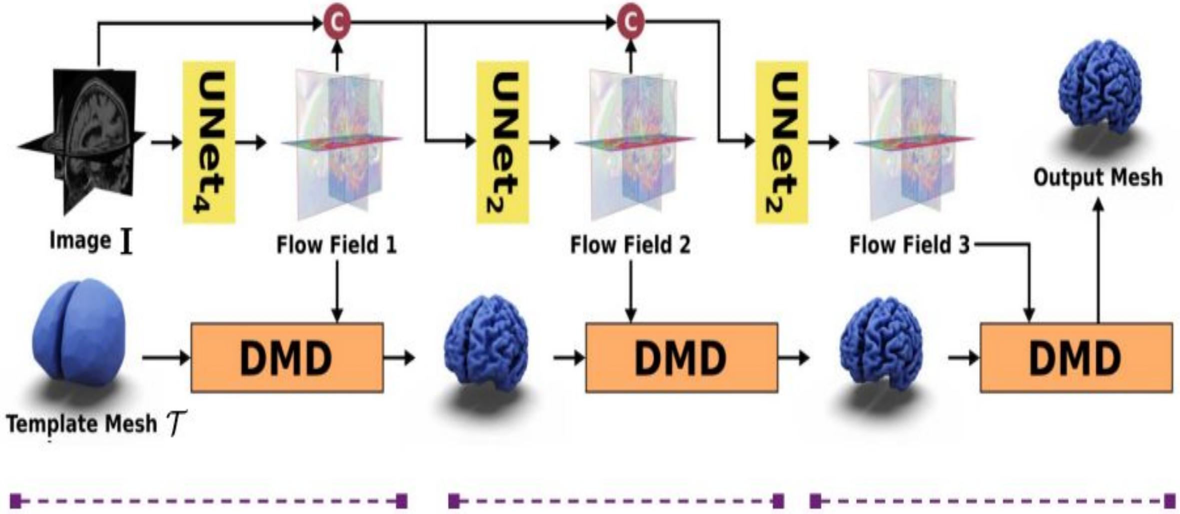
CorticalFlow consists of a chain of three deformations

The first deformation module receives as input a volumetric image and outputs a flow vector field with the same dimensions using UNet-3D

This discrete flow vector field is integrated by the DMD module to compute smooth deformations

The subsequent UNet-3D receives as input the image and the flow vector fields predicted by the previous deformation modules

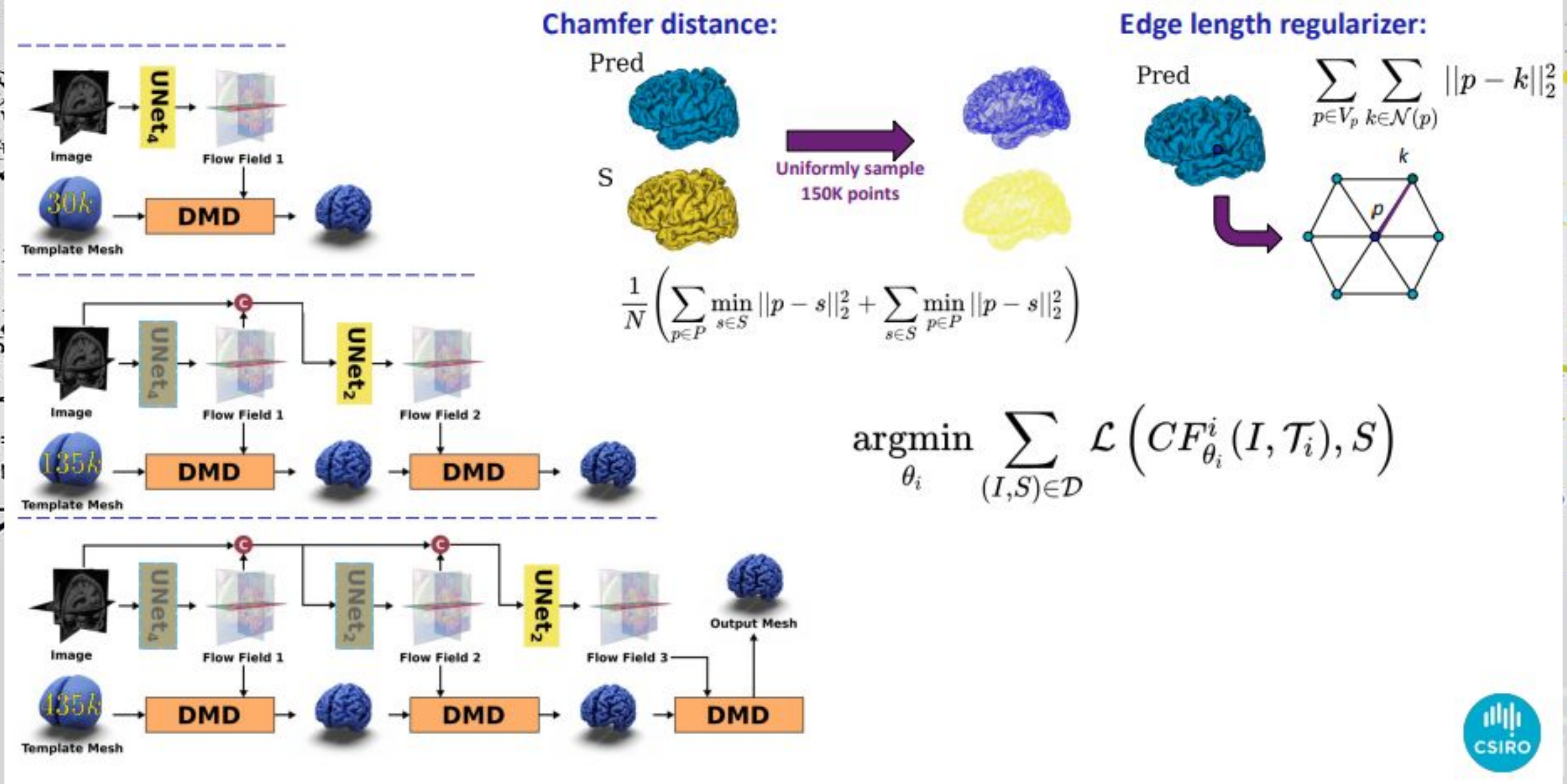
The set of resulting mappings are composed to produce the final mesh



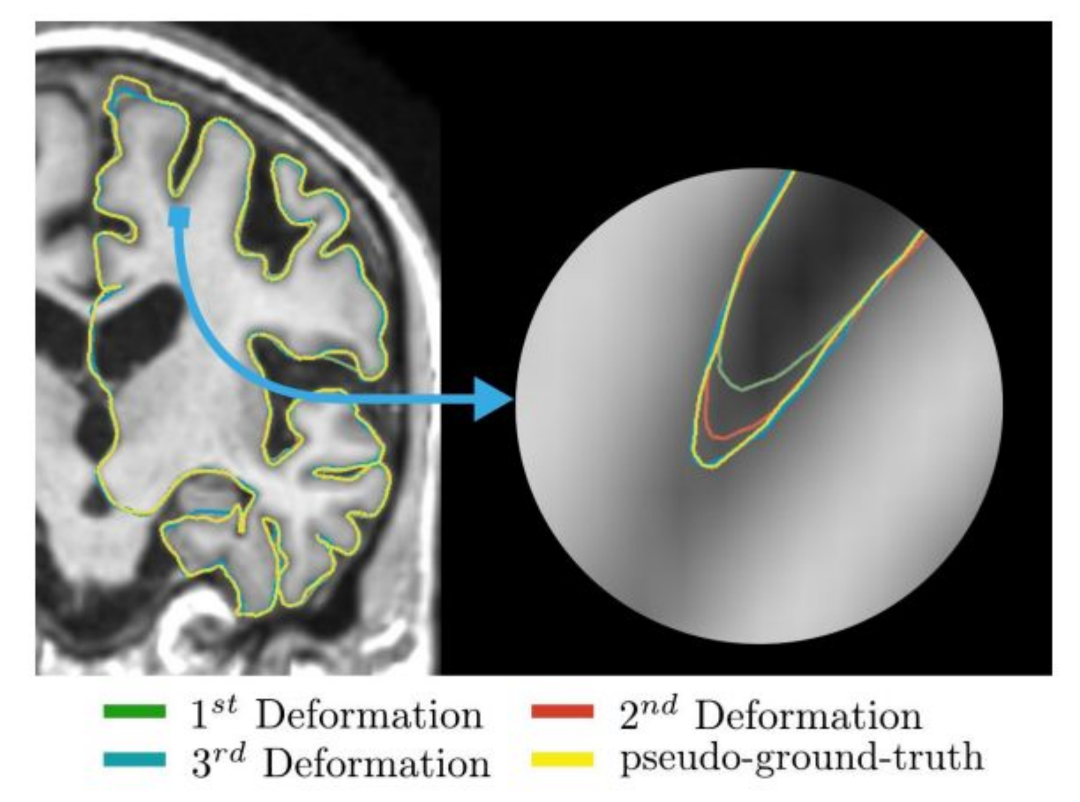
$$\mathrm{CF}^{1}_{\theta_1}(\mathbf{I}, \mathcal{T}_1) = \mathrm{DMD}\left(\mathrm{UNet}^{1}_{\theta_1}(\mathbf{I}), \mathcal{T}_1\right)$$

$$\mathrm{CF}^{i+1}_{\theta_{i+1}}(\mathbf{I}, \mathcal{T}_{i+1}) = \mathrm{DMD}\left(\mathrm{UNet}^{i+1}_{\theta_{i+1}}(\mathbf{U}_1^\frown \cdots \mathbf{U}_i^\frown \mathbf{I}), \mathrm{CF}^{i}_{\theta_i}(\mathbf{I}, \mathcal{T}_{i+1})\right)$$

Multiscale Training/Loss



Example results

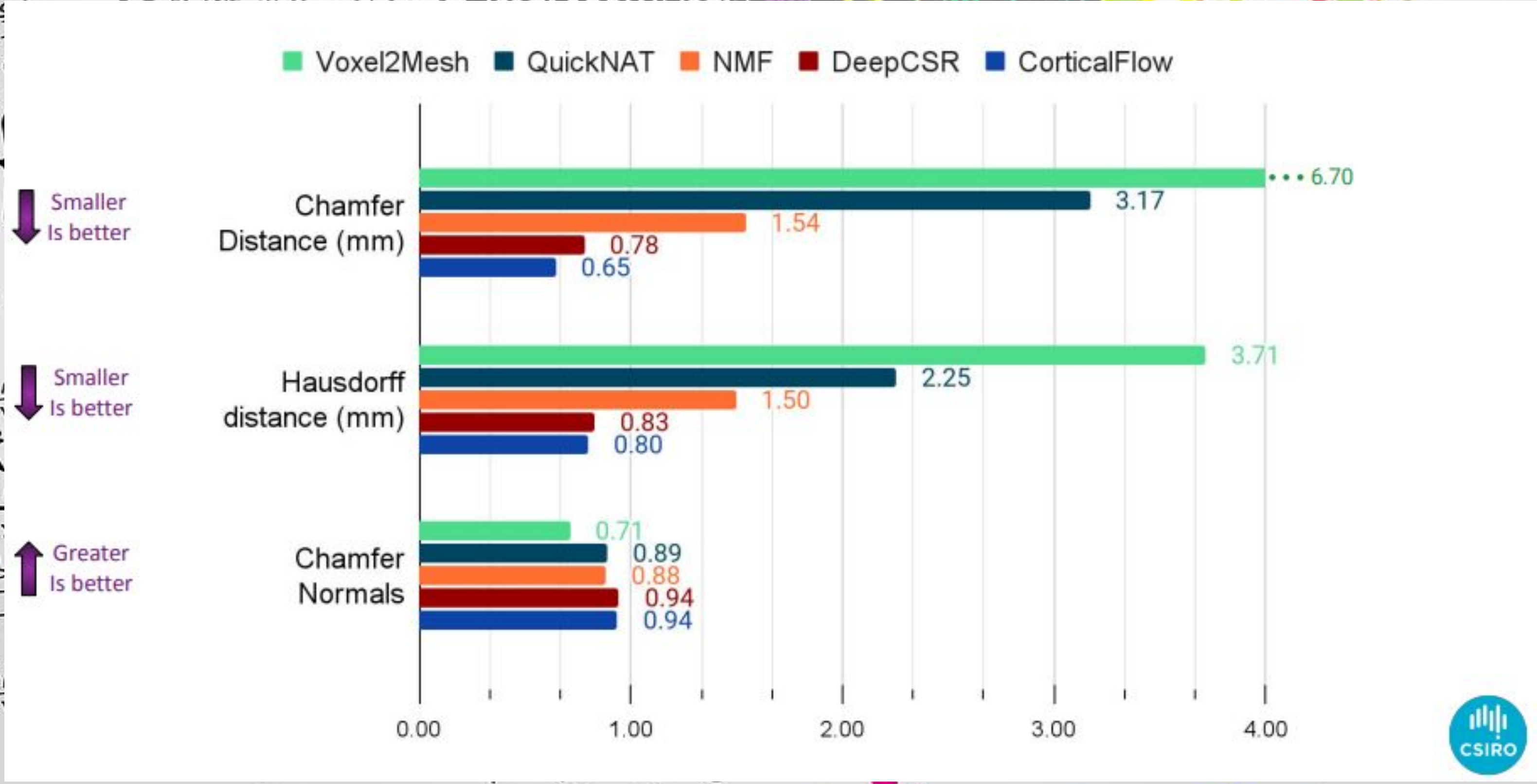


Experiments

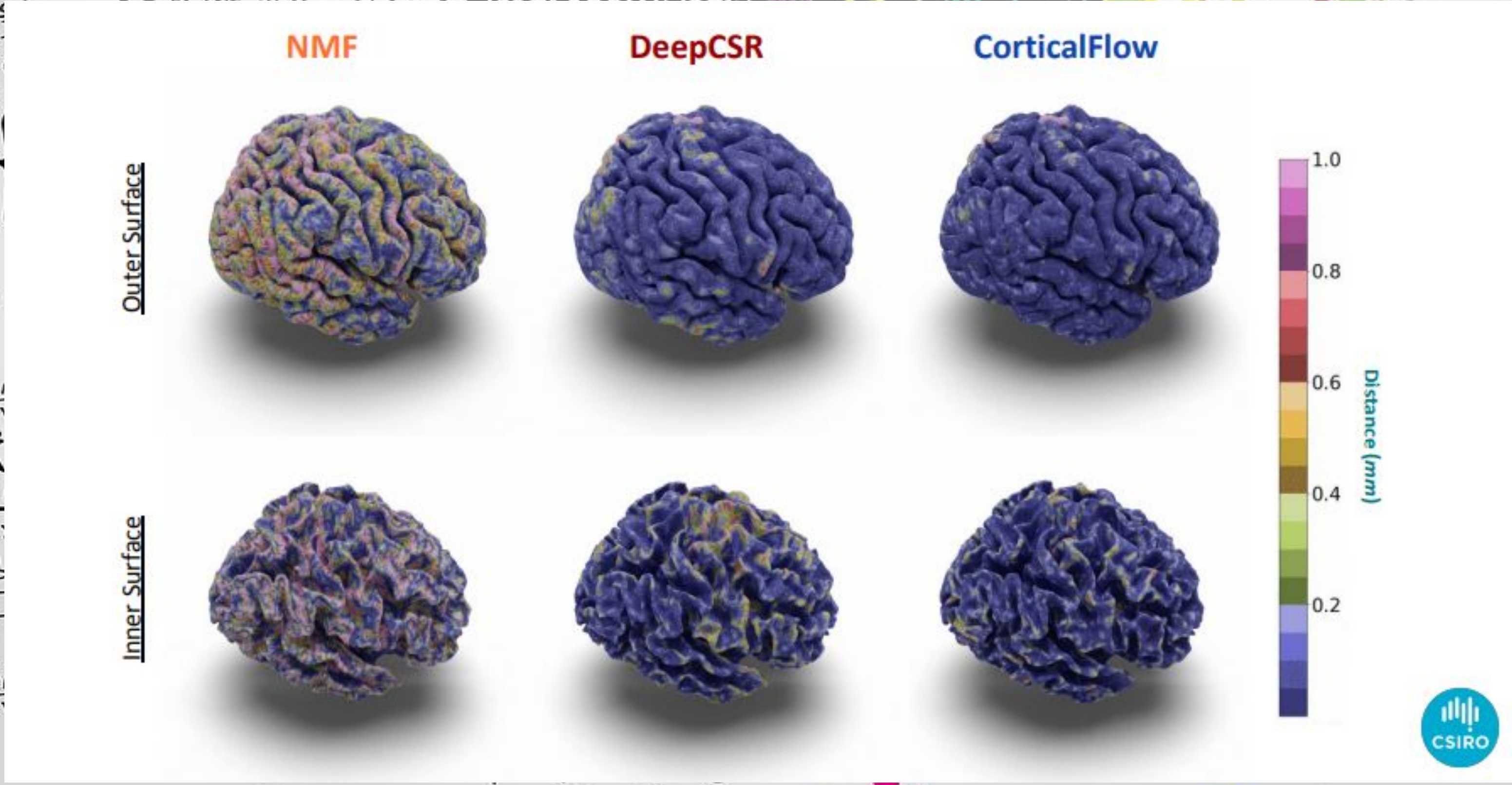
- **Dataset:**
  - MRIs, Pseudo ground truth surfaces, and data splits proposed in [1].
  - 3876 MRI images from **ADNI study**
  - Pseudo ground truth surfaces generated with the **FreeSurfer V6.0 cross-sectional** pipeline.
- **Baselines:**
  - QuickNAT [2]: Voxel-wise segmentation + surface extraction
  - Voxel2Mesh [3]: Deformable model with regularity penalties
  - NMF* [4]: Deformable model with diffeomorphic transformations
  - DeepCSR [1]: Implicit surface prediction + surface extraction + Topology Correction
- **Metrics:**
  - Geometric accuracy: Chamfer distance, Hausdorff distance, and Chamfer normals.
  - Surface regularity: Percentage of self-intersecting faces using **PyMeshLab**.
  - Time and space complexity: Average inference time (in seconds) and inference GPU memory footprint (in GB) to reconstruct the **four cortical surfaces**.

[1] - Santa Cruz et al. DeepCSR: A 3d deep learning approach for cortical surface reconstruction. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021.
[2] - Roy et al. Quicknat: A fully convolutional network for quick and accurate segmentation of neuroanatomy. NeuroImage, 186:713–727, 2019.
[3] - Wickramasinghe et al. Voxel2mesh: 3d mesh model generation from volumetric data. In International Conference on Medical Image Computing and Computer-Assisted Intervention, 2020.
[4] - Gupta and Chandraker. Neural mesh flow: 3d manifold mesh generation via diffeomorphic flows. In Advances in Neural Information Processing Systems, 2020.

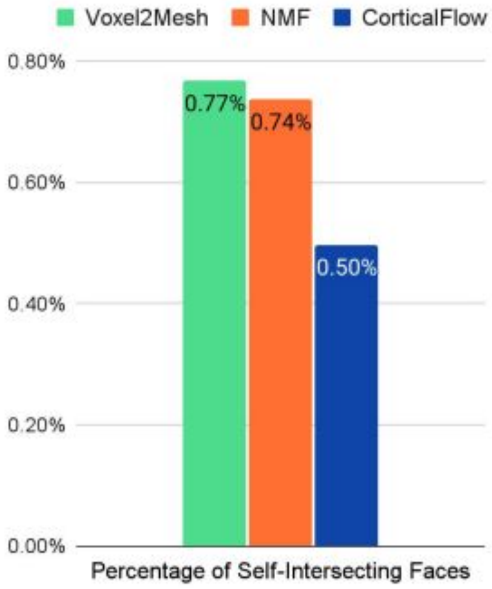
Average Geometric Accuracy (losses used)



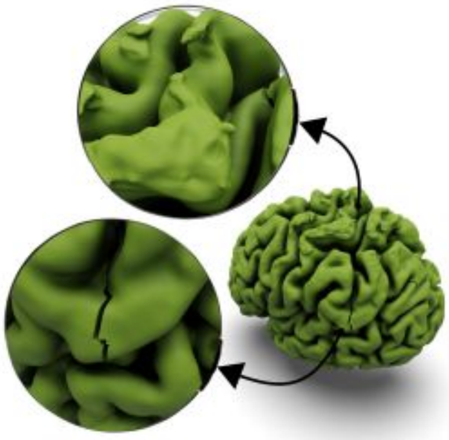
Error color coded surfaces (loss)



Surface Regularity



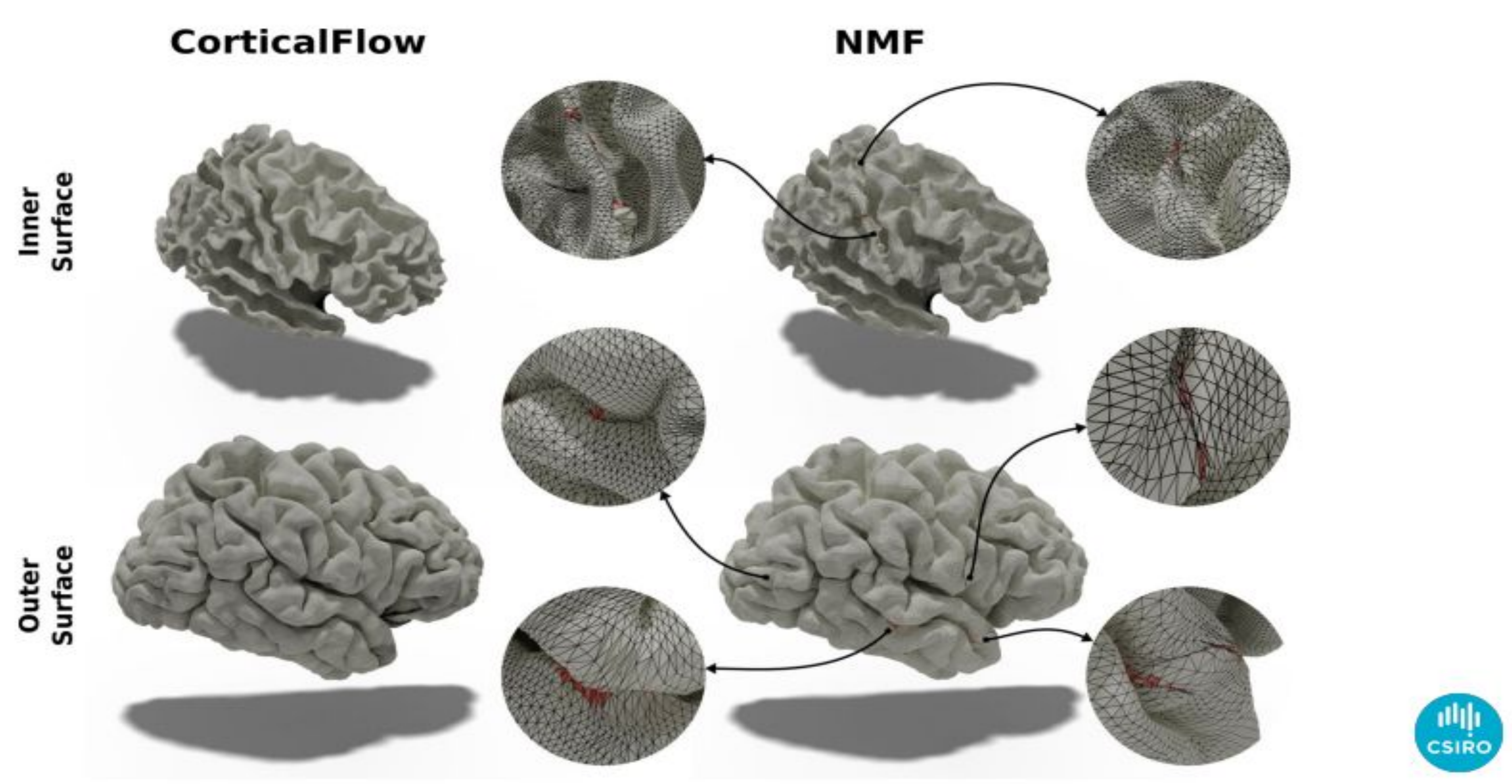
QuickNAT
Multiple connected components, handles and holes.



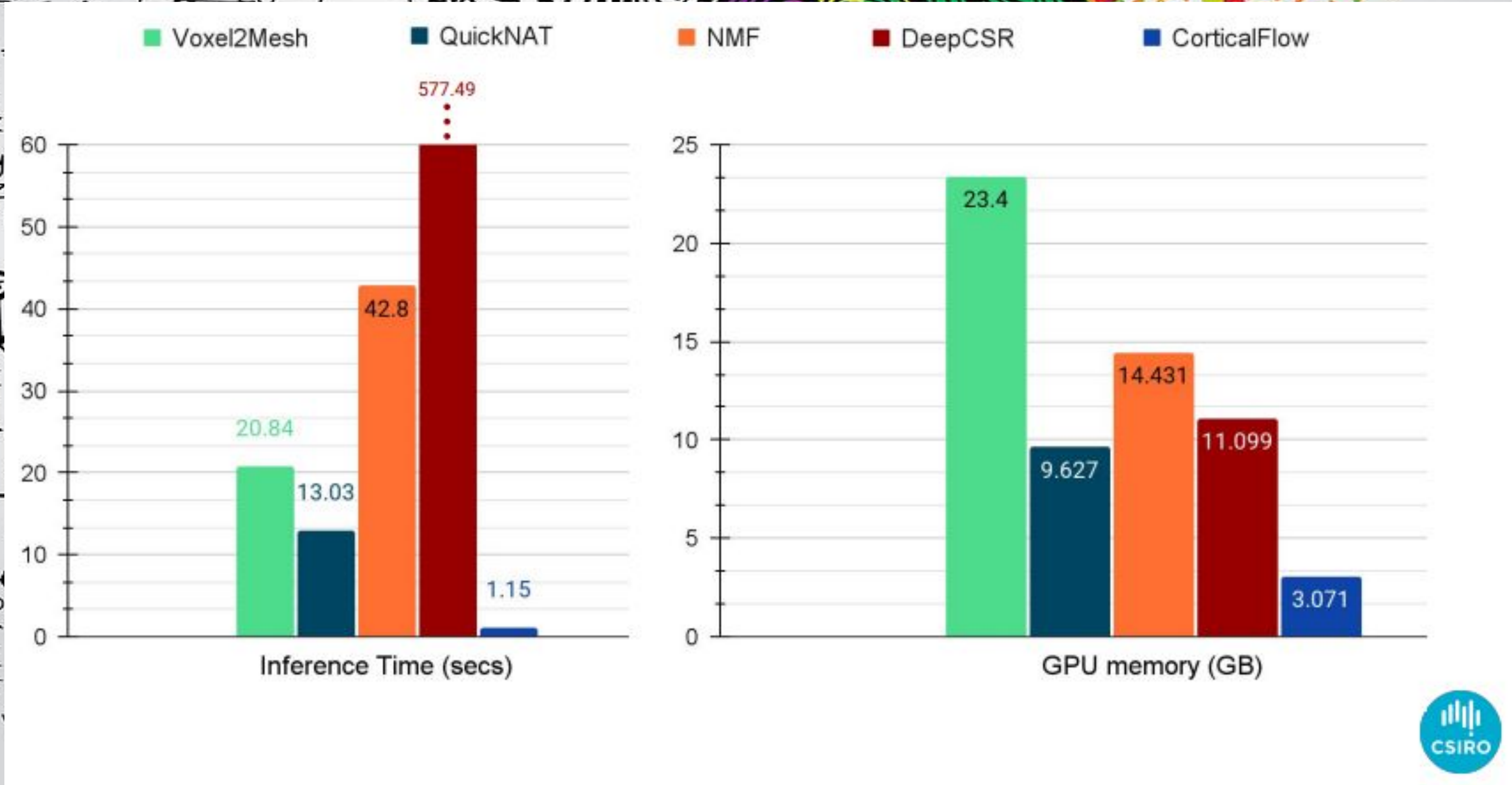
DeepCSR
Anatomical mistakes due to topology correction

CSIRO

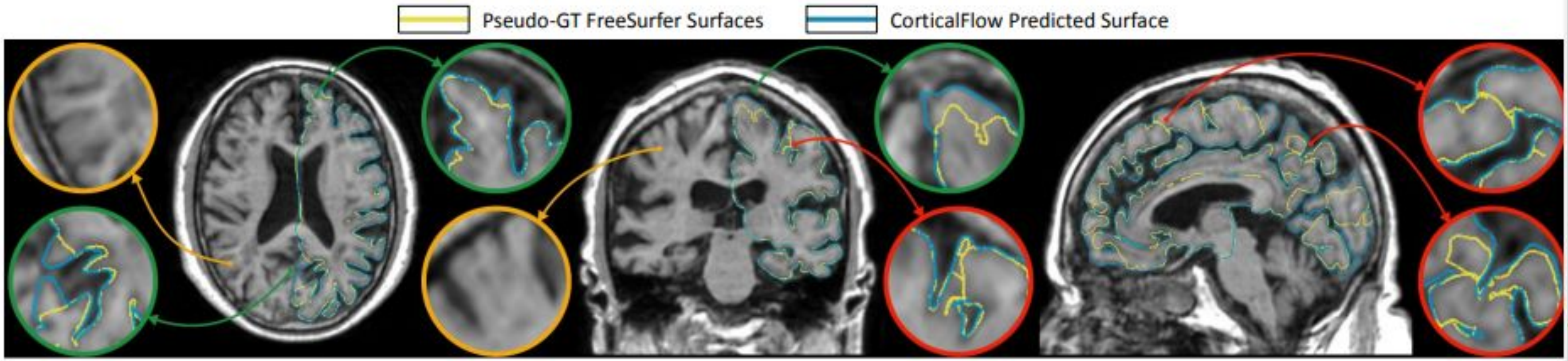
Surface Regularity



Inference time and gpu time



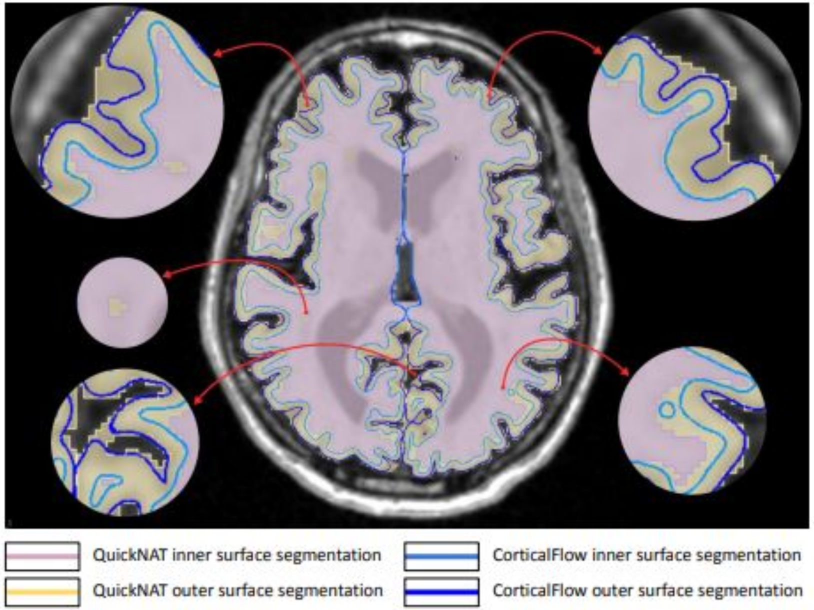
Comparison to Pseudo-Ground-Truth



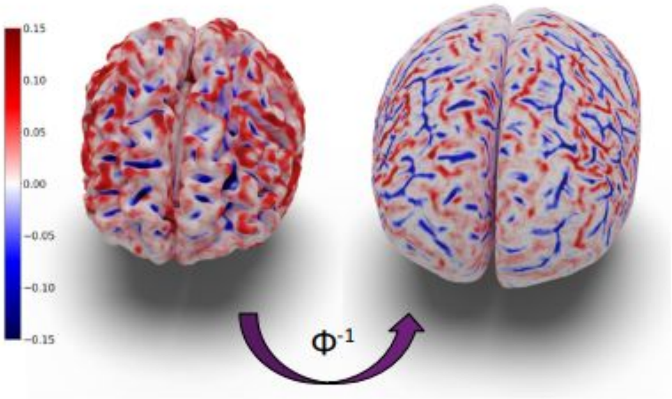
- Orange circles highlight blurry MRI regions.
- Green circles highlight FreeSurfer's underestimated area.
- Red circles highlight non-plausible predictions avoided by CorticalFlow thanks to the diffeomorphism of its predicted deformations.

Future Applications

Segmentation at subvoxel resolution



Analysis of surface descriptors on a common reference surface



CSIRO

Conclusion

This paper ...

- introduces **CorticalFlow** - a novel geometric deep learning model for efficiently reconstructing high-resolution, accurate, and regular triangular meshes from volumetric images.
- derives a **diffeomorphic mesh deformable (DMD) module** that efficiently produces diffeomorphic mappings from stationary velocity field.
- shows that CorticalFlow is more accurate, robust, faster and memory efficient than state-of-the-art models in the **cortical surface reconstruction problem** which can facilitate large-scale medical studies and support new healthcare applications.

CSIRO

References

CorticalFlow_NeurIPS21 (lebrat.github.io)

https://lebrat.github.io/CorticalFlow/assets/pdf/CorticalFlow_NeurIPS21.pdf

https://lebrat.github.io/CorticalFlow/