Lyon, Avril 17– 21
DLMI2023

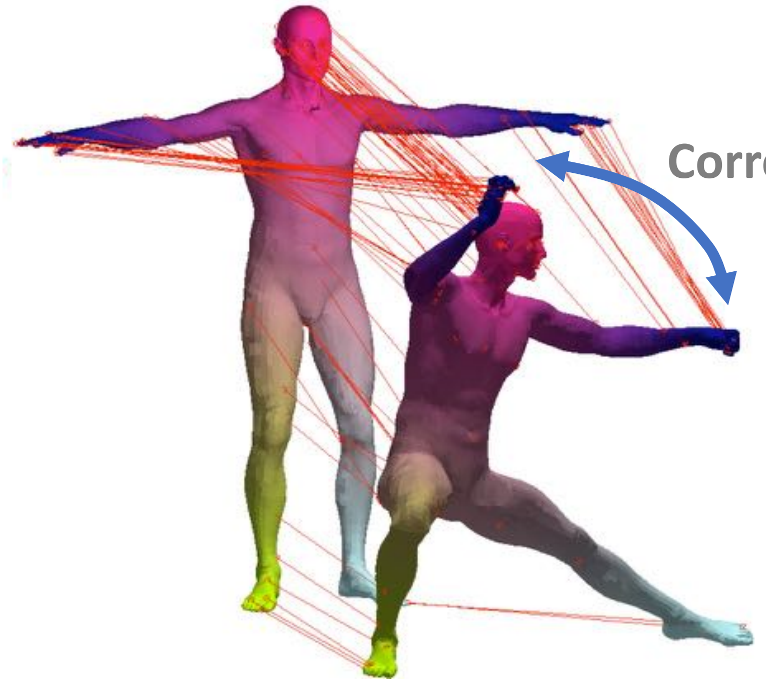
Geometric Deep Learning in Medical Imaging

*Prof. **Hervé Lombaert**, ETS Montreal*

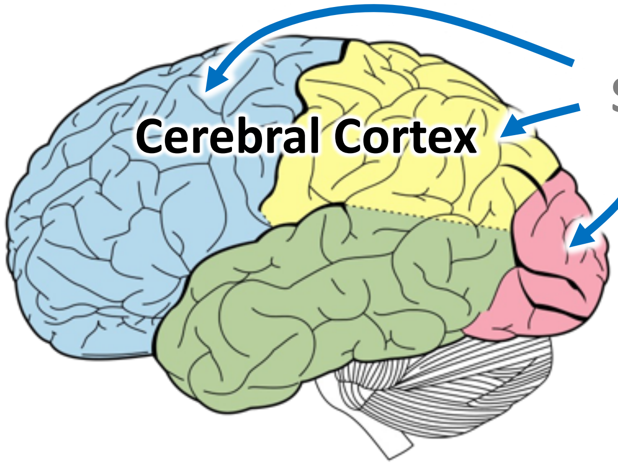
***Spring School** on Deep Learning for Medical Imaging 2023*

Geometry & Machine Learning

- How to exploit **Shapes & Geometry** for learning complex data?



Correspondence



Segmentation

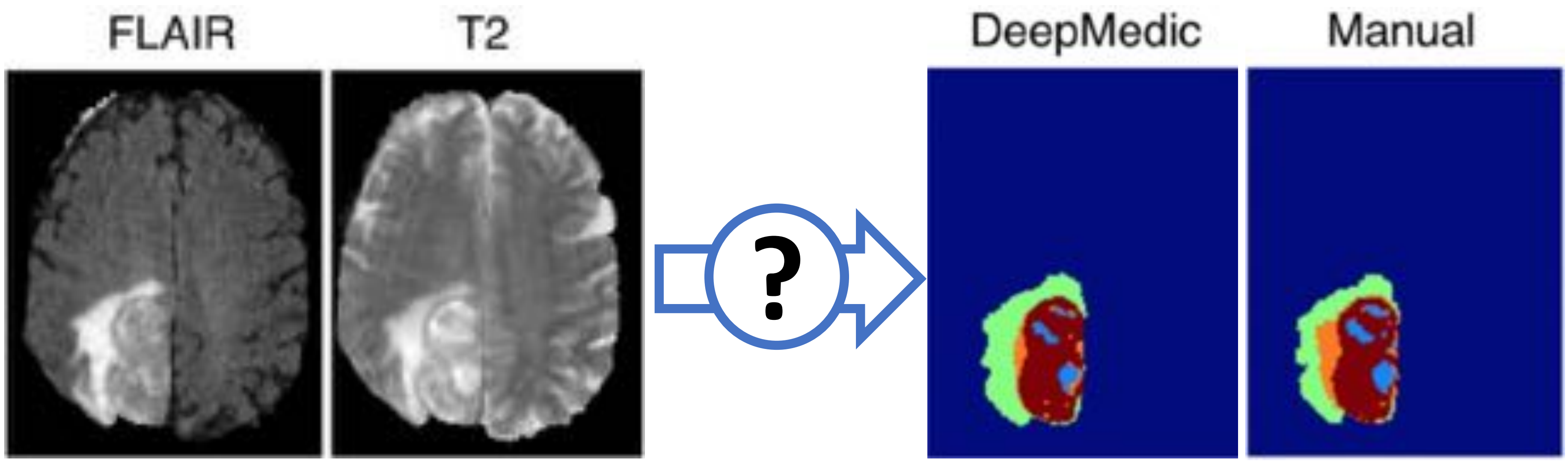
Classification

Regression

Key role in **cognition, planning & perception**

Segmentation on Medical Images

- **One Example** – Finding **Lesions** on Brain MRIs



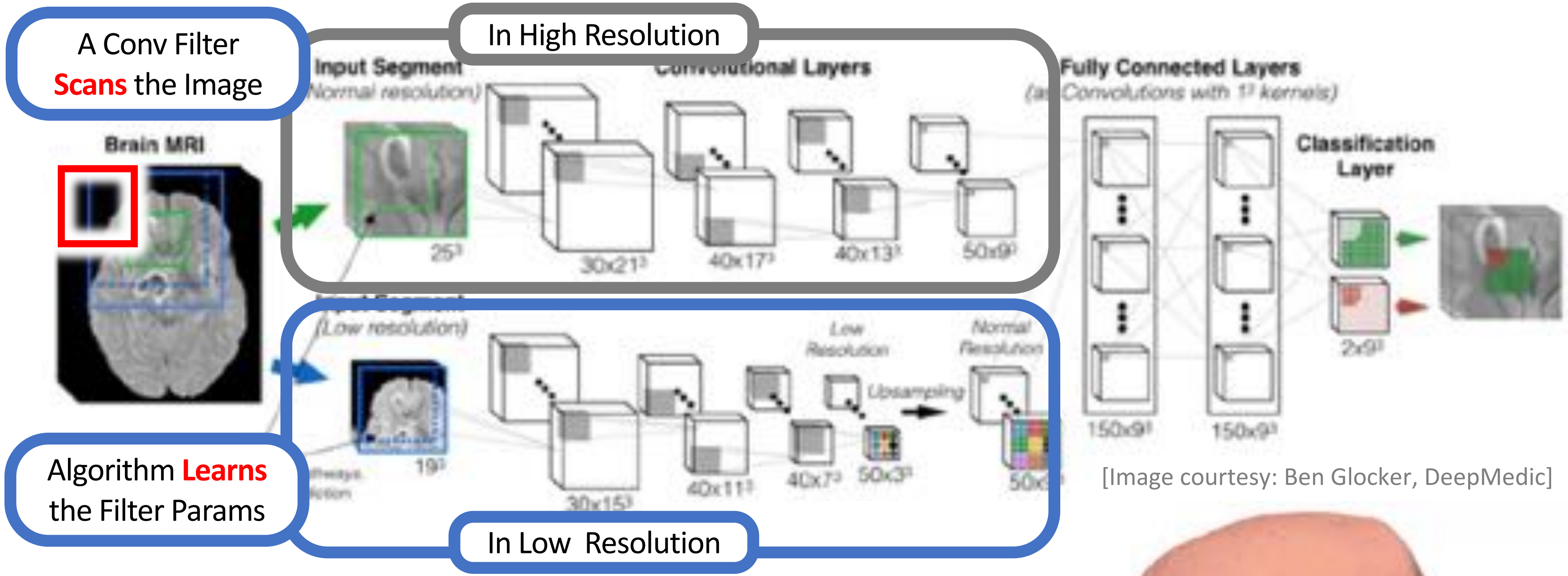
(Finding Lesions)
Blue – Necrosis
Red – Active Core
Green – Oedema

Trained from
Manual Segmentations

Kamnitsas et al, DeepMedic, MedIA 2017

Segmentation on Medical Images

- **Conv Nets** (CNNs) on Images



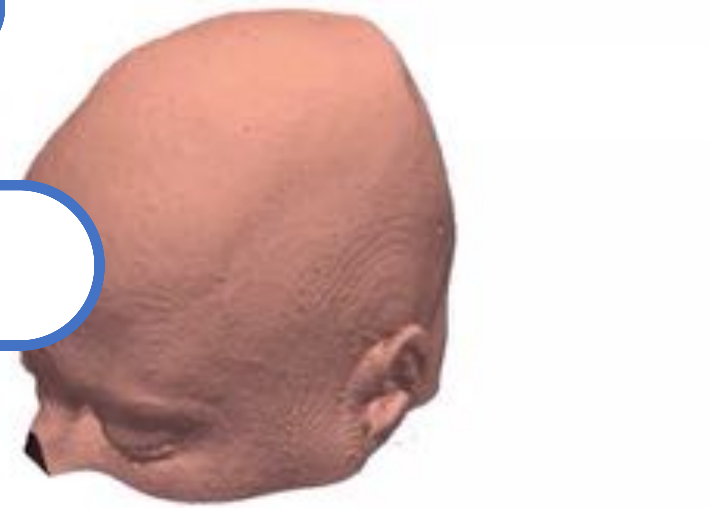
[Image courtesy: Ben Glocker, DeepMedic]

Many works on CNN Segmentations

Zikic et, 2014
Pereira et al, 2015
Prasson et al, 2013
Roth et al, 2014
Ciresan et al, 2013
Dolz et al, 2018
Litjens et al, 2017 – **and more**

Ronneberger et al, 2015
Zhao et al, 2018

One issue



Kamnitsas et al, DeepMedic, MedIA 2017

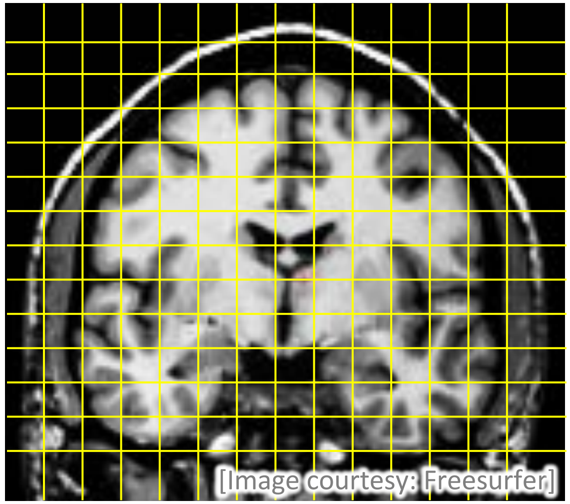
1 2 3 4 5 6 7

From Images to **Surfaces**

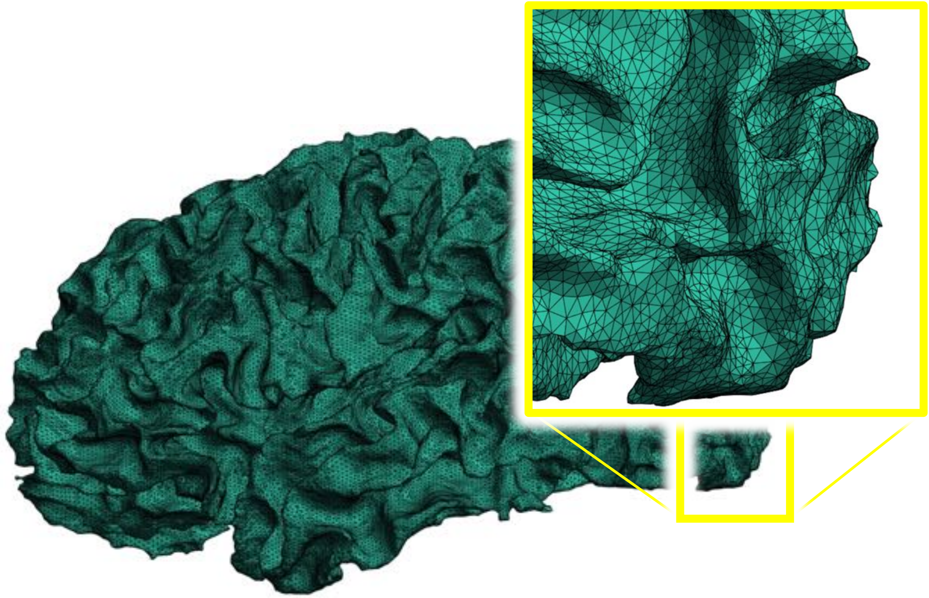
Why a need to work on Surfaces?

Images vs Surfaces

- Algorithms **rely** on an **Image Grid**

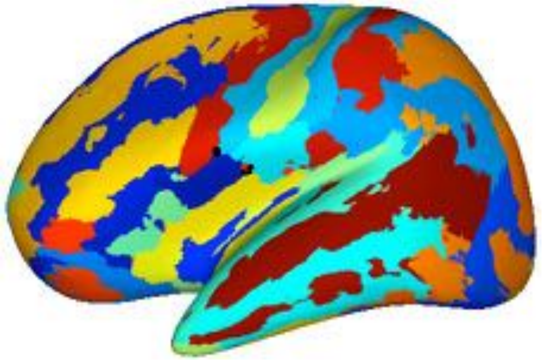


Point Coordinates
defined as (*x,y,z*) Coordinates

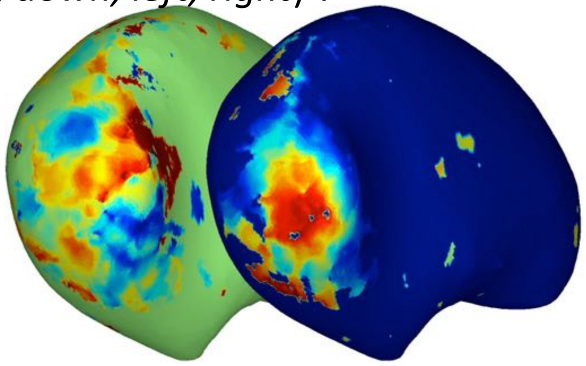


Neuroimaging – Data is often on surfaces
where is (*up, down, left, right*) ?

Why Learning on Surfaces?



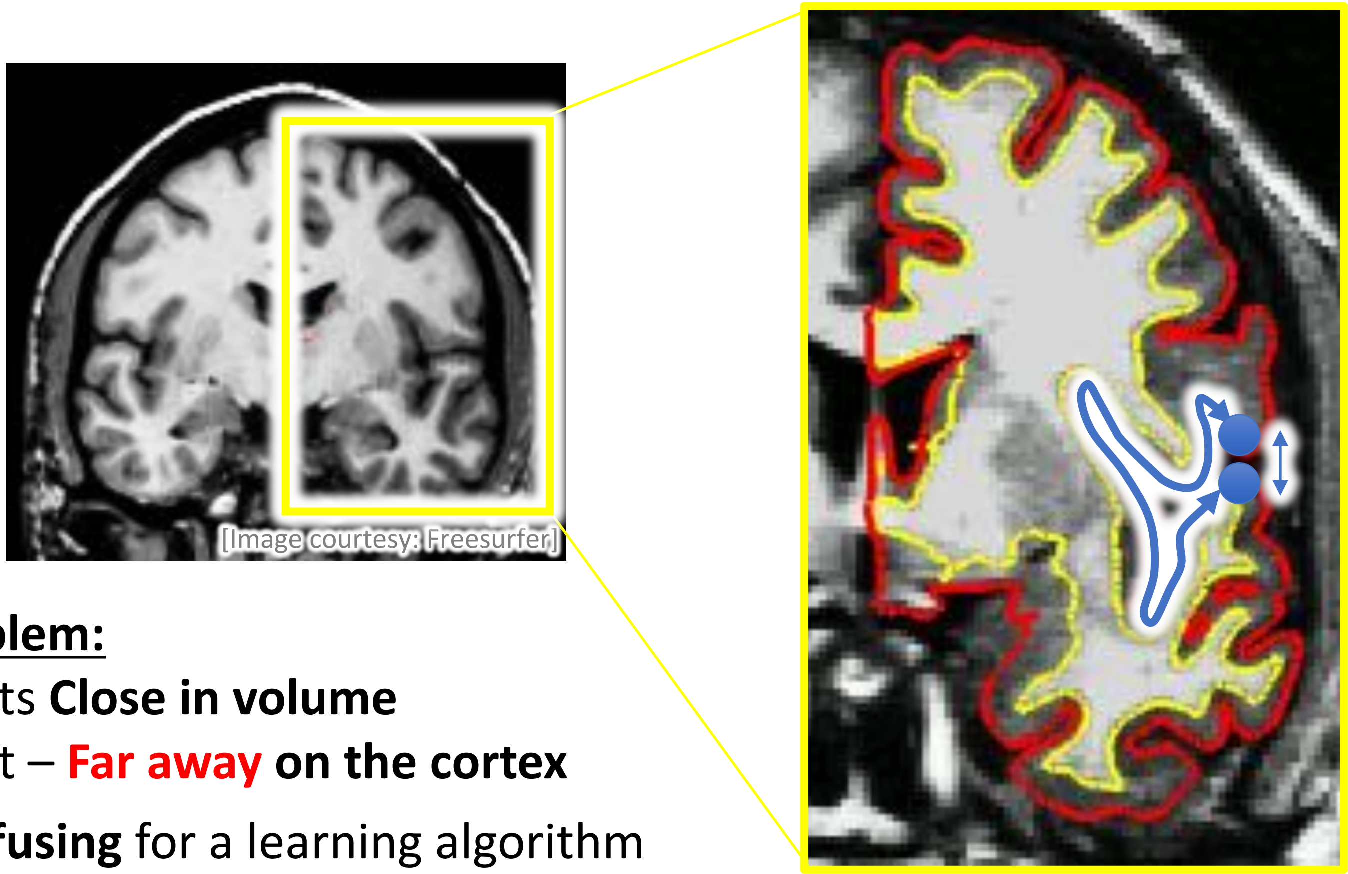
Cortical Parcellation



Functional Imaging

Images vs Surfaces

- Exploiting the **Surface Geometry**



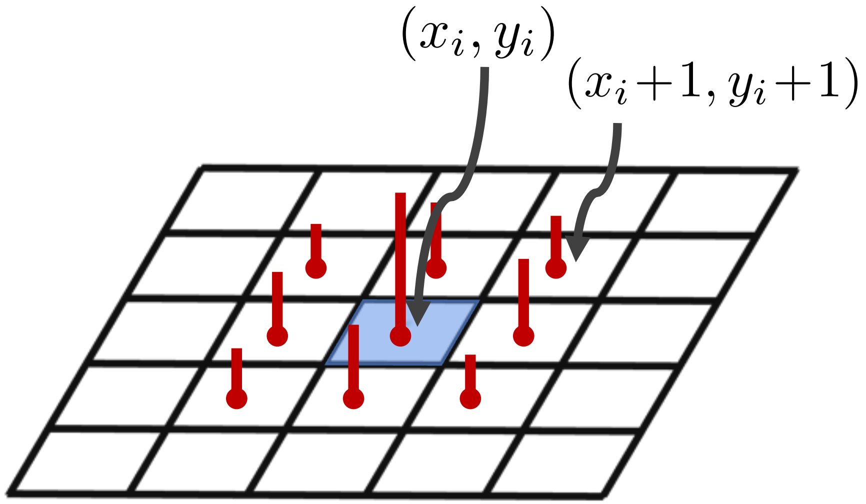
Problem:
Points **Close in volume**
– but – **Far away on the cortex**

Confusing for a learning algorithm

How to Learn on Surfaces?

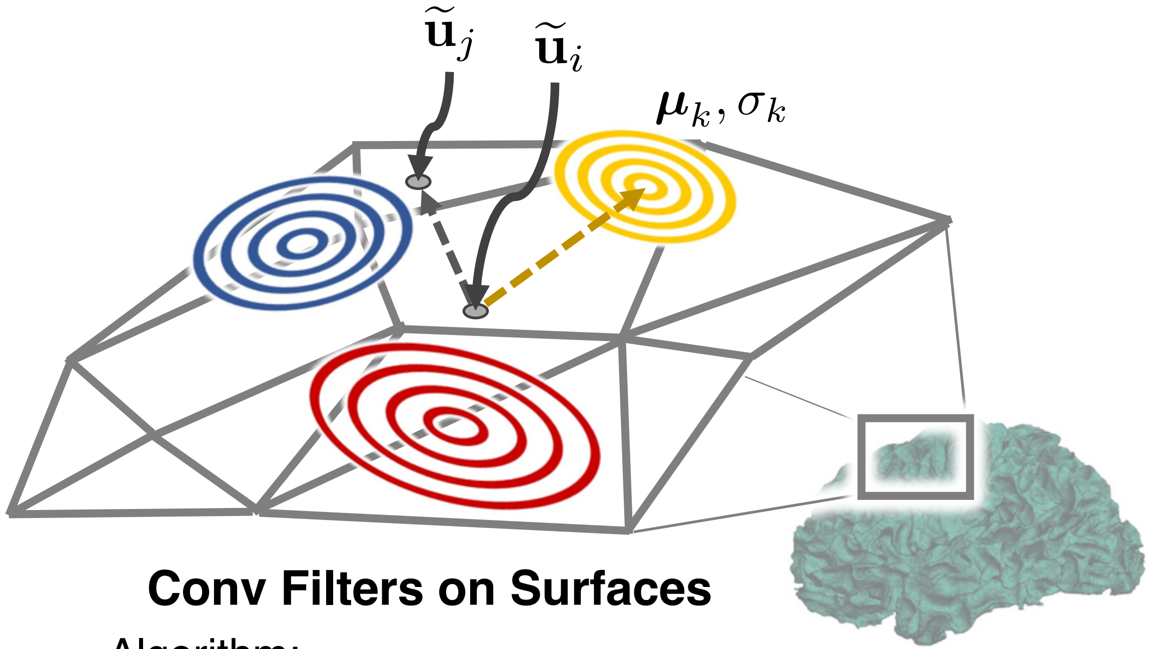
Convolutions on Surfaces

- Defining Kernels on **Curved Spaces**



Conv Filter on a Grid

Algorithm:
– **Learns** the Filter parameters (the red bars)
– Supposes neighbors are **on a grid**

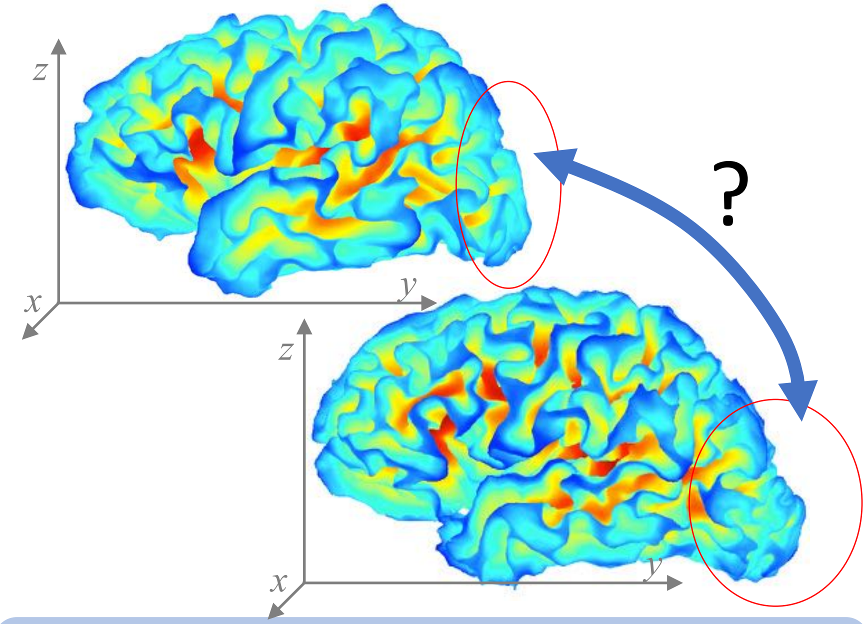


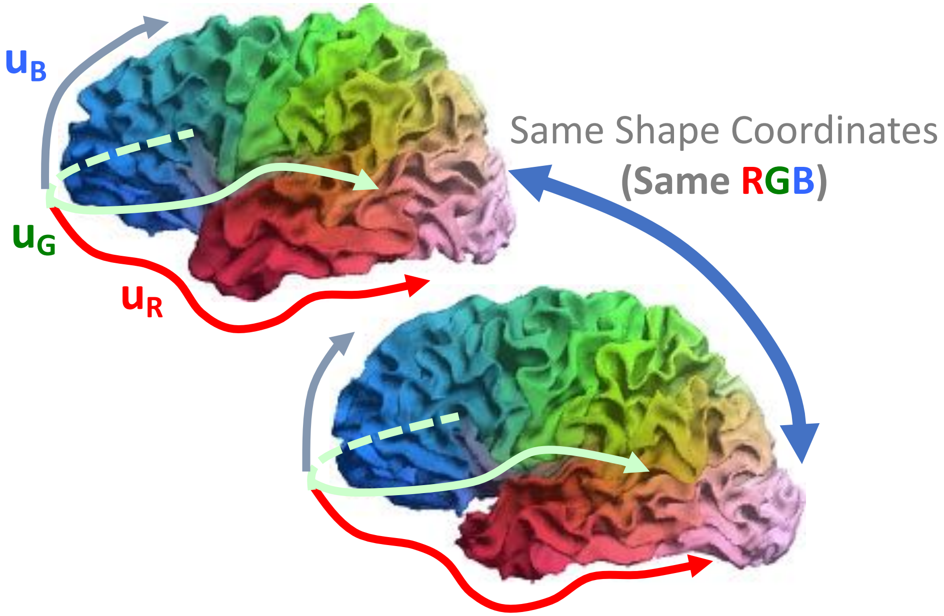
Conv Filters on Surfaces

Algorithm:
– **Learns** the Filter parameters (μ's and σ's)
– Requires **Graph Neighborhoods**

Parameterization – Euclidean vs Spectral Coordinates

Cartesian Coordinates versus Shape (Spectral) Coordinates





Cartesian Coordinates
Equivalent Points → **May NOT Overlap in Space**

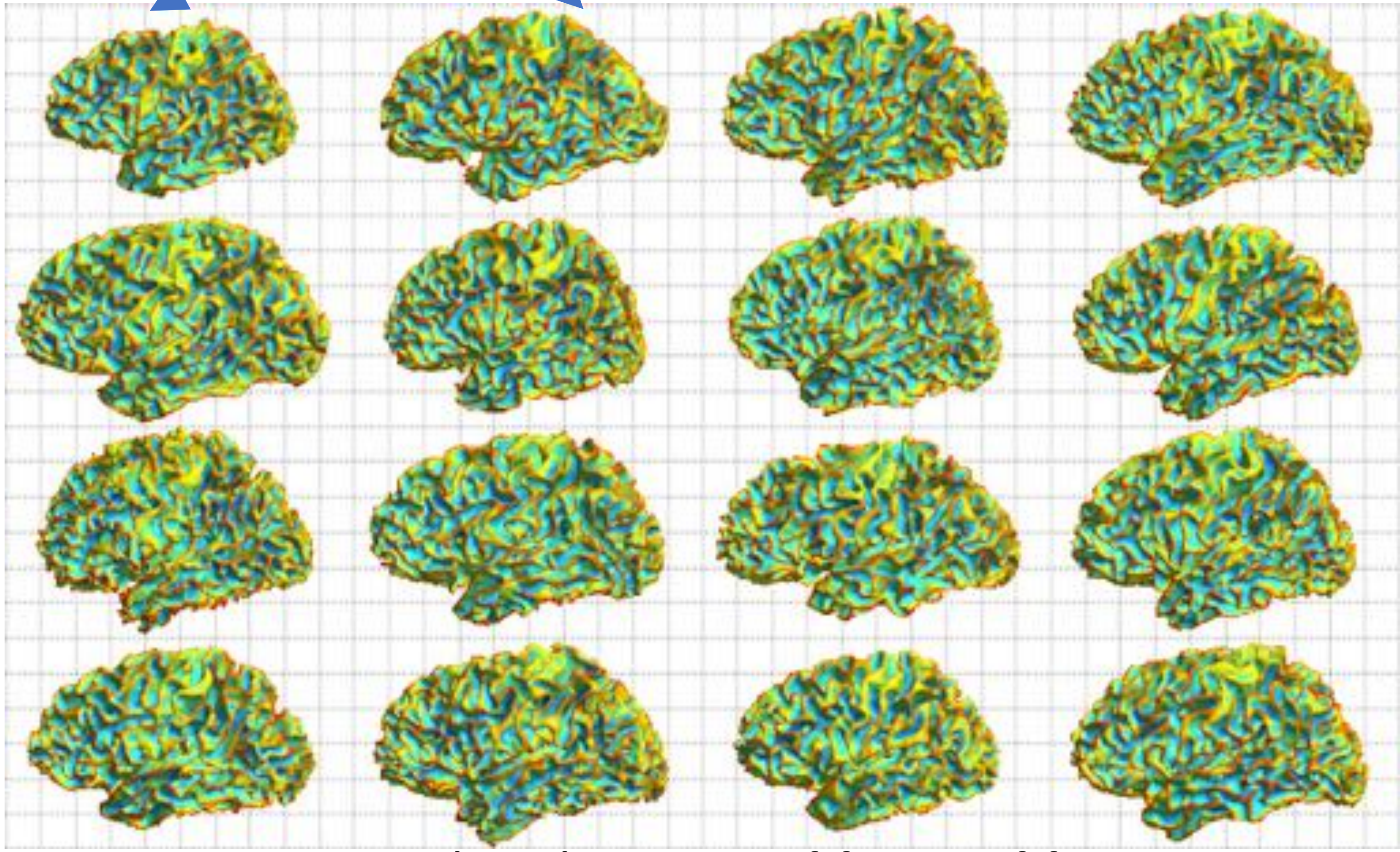
Shape Coordinates
Equivalent Points → **Similar Shape Characteristics**

Core Idea
Use **Shape Coordinates** for Matching

Reuter, IJCV (2009)
Niethammer, Reuter, Wolter, Bouix, Peinecke, Koo, Shenton, MICCAI (2007)
Qiu, Bitouk, Miller, TMI (2006)
Shi, Lai, Wang, Pelletier, Mohr, Sicotte, Toga, TMI (2014)
Germanaud, Lefevre, Toro, Fischer, Dubois, Hertez, Mangin, Neuroimage (2012)

Challenge – Anatomical Variability

Same Point?



Complex Shapes, **Highly variable**

How to find **point correspondences?**

Challenge – Anatomical Variability

One Related Problem –
Matching Points between Brains

Flowing Surfaces

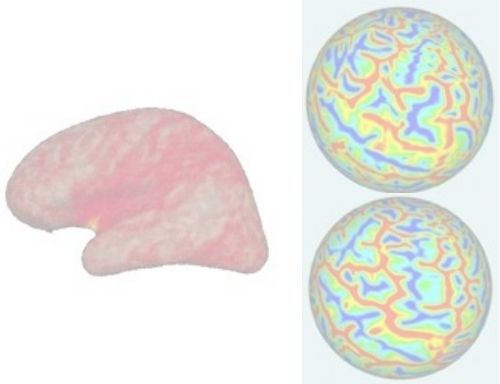
• Costly **(CPU, mesh size)**

(LDDMM and variants)

$$E(v) = \int_0^1 |v_t|^2 dt + \int_0^1 \left|I \circ \phi_1^{-1}(y) - J(y)\right|^2 dy$$

Sphere Inflations

• Costly **(3 to 4 hours)**



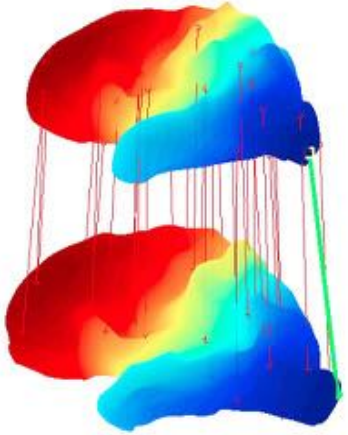
(FreeSurfer, Spherical Demons)

$$\phi^{(t)} = c^{(t)} \circ \exp\left(v^{(t)}\right) \text{ on } S^2$$

Spectral Matching

✓ Fast **(Few seconds)**

✓ Accurate (as FreeSurfer)



Proposal → **Fast**, as **Accurate**

Dense Point Correspondence
300k+ meshes

Beg, Miller, Trouvé, Younes, IJCV (2005)
Fischl, Sereno, Tootell, Dale, HBM (1999)
Yeo, Sabuncu, Vercauteren, Ayache, Fischl, Golland, TMI (2010)
Lombaert, Grady, Polimeni, Cheriet, **PAMI (2013)**

1 **2** 3 4 5 6 7

Background on Spectral Shape Analysis

How to Represent and Exploit Surfaces?

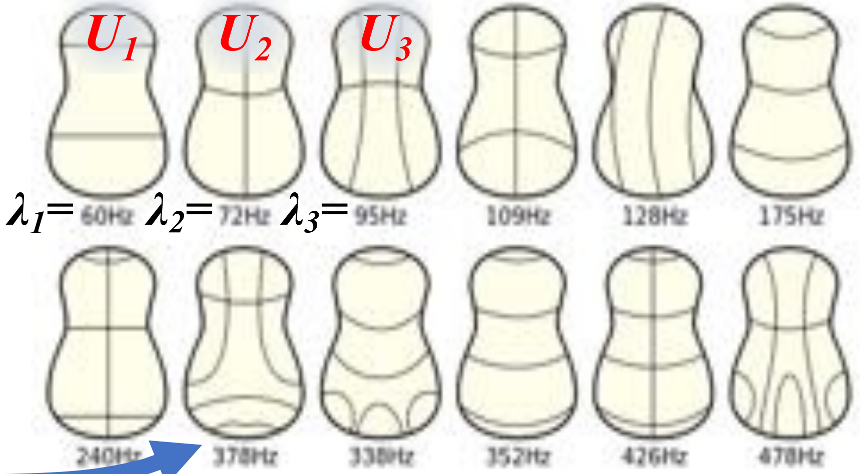
Spectral Signature

Vibration patterns *governed by* Shape

Cymatics: Science vs Music, Nigel Stanford

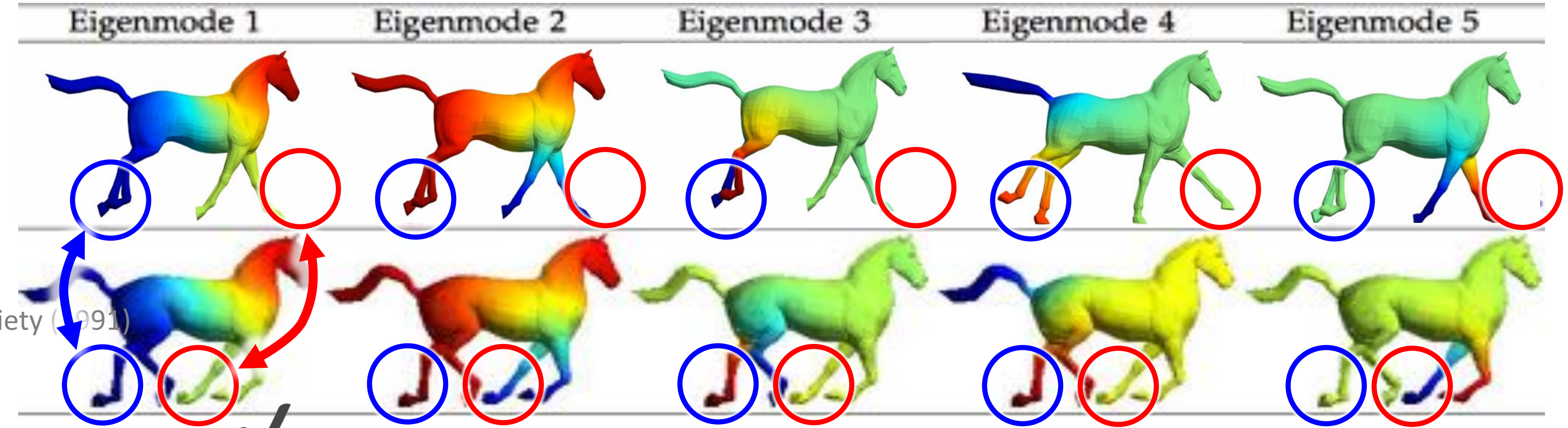
Spectral Signature

Shape Vibration → Unique **intrinsic** Shape Signature



Spectral Decomposition

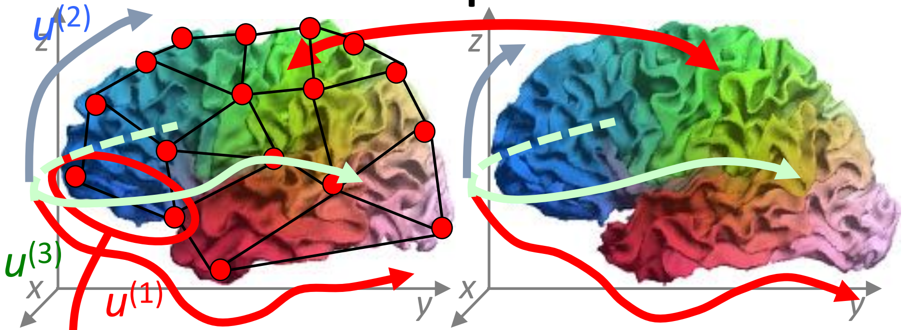
***Spectral* Signature**



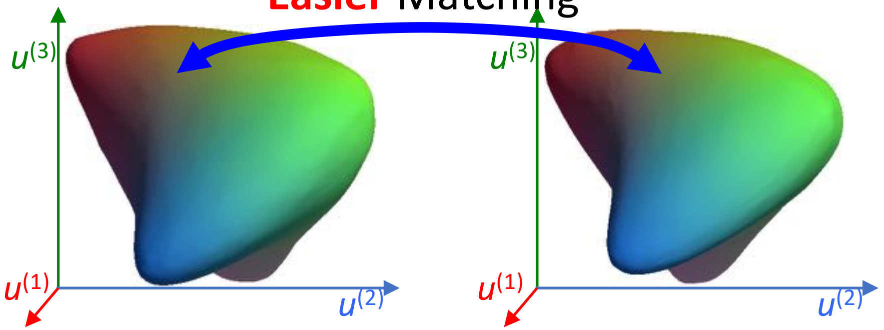
✓ **Good: Equivalent Points → Same** (shape) **Spectral Coordinates**

Umeyama, PAMI (1988)
Scott & Longuet-Higgins, Royal Society (1991)
Shapiro & Brady, IVC (1992)
Mateus, CVPR (2008)
Jain & Zhang, ICSMA (2006)
Reuter *et al.*, MICCAI (2007), CAD (2009)
Ovsjanikov et al., SIGGRAPH (2012)
Lombaert, Grady, Polimeni, Cheriet, **IPMI (2011), PAMI (2012)**

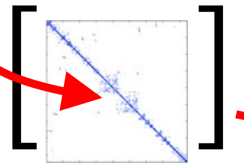
Method – Spectral Shapes



Surfaces in **Euclidean Space**



Surfaces in **Spectral Space**

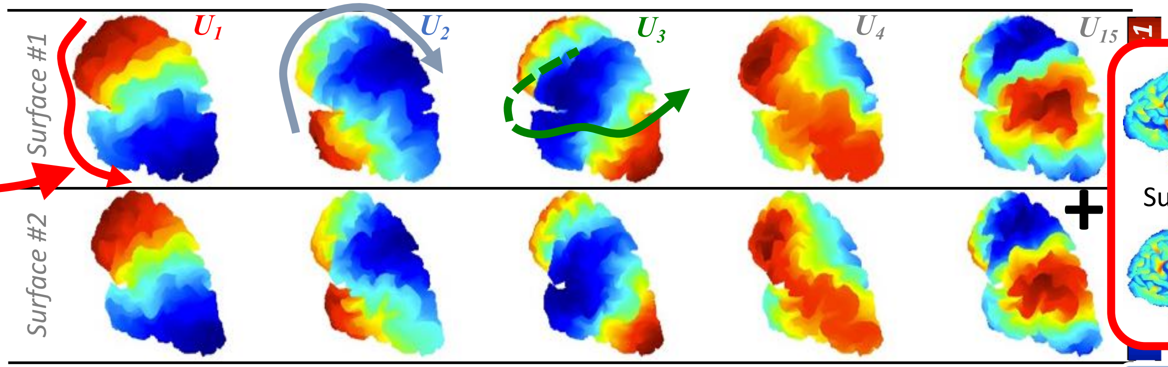
$L =$ 

Graph Laplacian

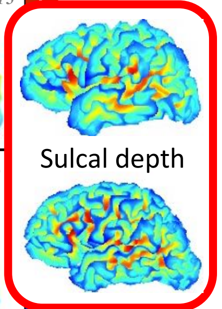
Edge (i,j) : Row/Column (i,j)

Spectral Decomposition

$$L = U^T \Lambda\, U$$



Spectral Decomposition (Eigenvectors of **L**)



Add Data

Energy:

$$\phi(v_i) = \underset{\phi(v_i)\in B}{\operatorname{argmin}} \left\| D_A(v_i) - D_B(\phi(v_i)) \right\|^2 +$$

Data Term

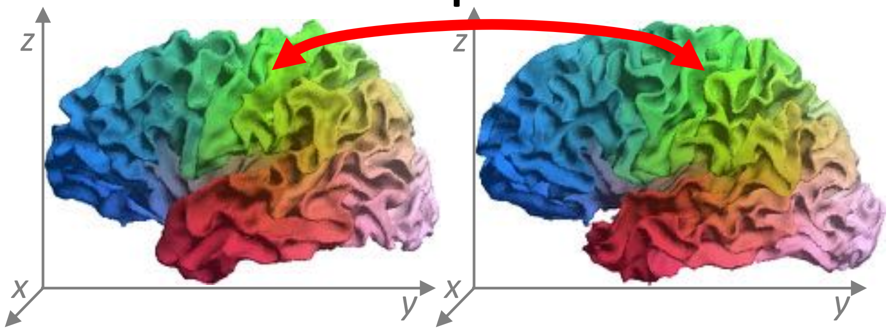
Spatial Term

Nearest Neighbor Search between vectors $[D_A U_A]$ & $[D_B U_B]$

Lombaert, Ayache, **IPMI (2015)**
Lombaert, Sporring, Siddiqi, IPMI (2013)
Lombaert, Grady, Pennec, Ayache, Cheriet, **ECCV (2012), IJCV (2014)**
Lombaert, Grady, Polimeni, Cheriet, IPMI (2011), **PAMI (2012)**

Method – Spectral Shapes

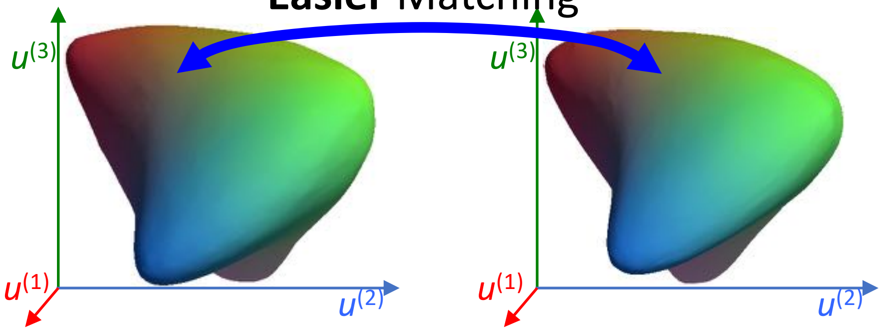
Correspondence?



Surfaces in **Euclidean Space**

✗ Before (FreeSurfer): CPU (3-4hrs)

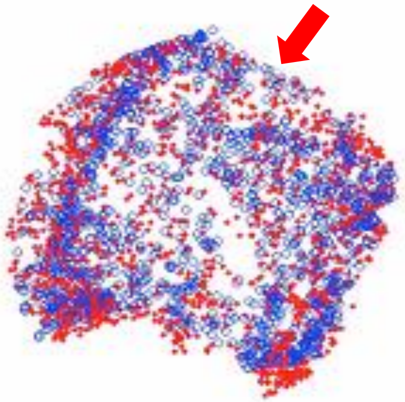
Easier Matching



Surfaces in **Spectral Space**

✓ After (Spectral Matching): < 1 min

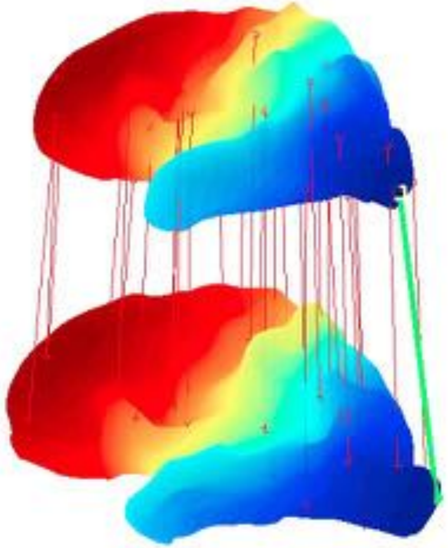
Correspondence



Spectral Matching

Surface 1 in Red

Surface 2 in Blue



***Point-to-point* Correspondence**

300K nodes, < 1 min

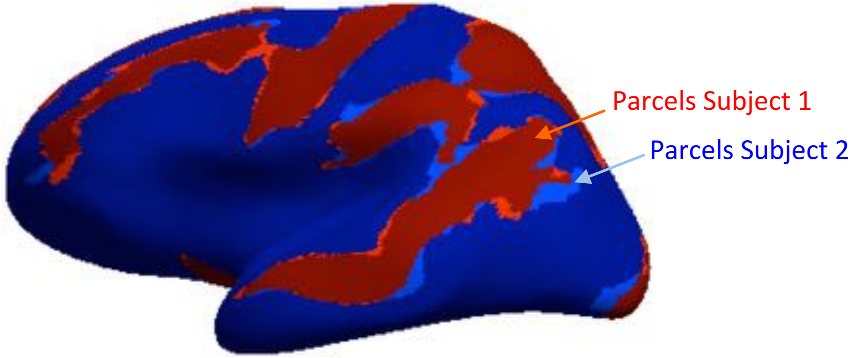
Lombaert, Ayache, **IPMI (2015)**

Lombaert, Sporring, Siddiqi, IPMI (2013)

Lombaert, Grady, Pennec, Ayache, Cheriet, **ECCV (2012), IJCV (2014)**

Lombaert, Grady, Polimeni, Cheriet, IPMI (2011), **PAMI (2012)**

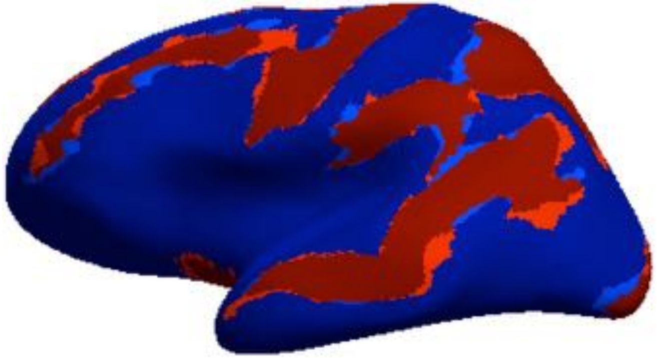
Comparison with State-of-the-Art



FreeSurfer

✔ Average Dice = **0.84** (±0.08)

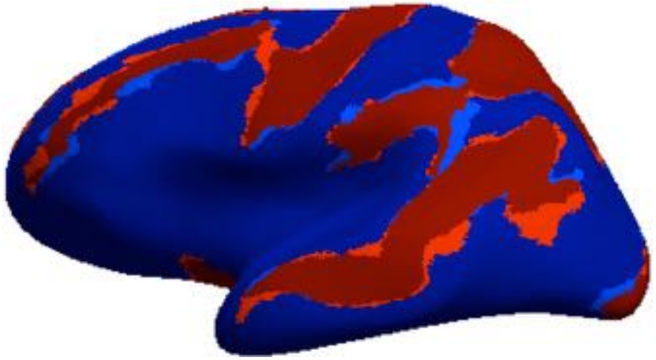
✗ 2hrs + 1hrs for one matching
(inflation) (matching)



Spherical Demons

✔ Average Dice = **0.85** (±0.07)

✗ 2hrs + 3mins for one matching
(inflation) (matching)



Spectral Matching

Average Dice = **0.83** (±0.08)

<1min for one matching
(total)

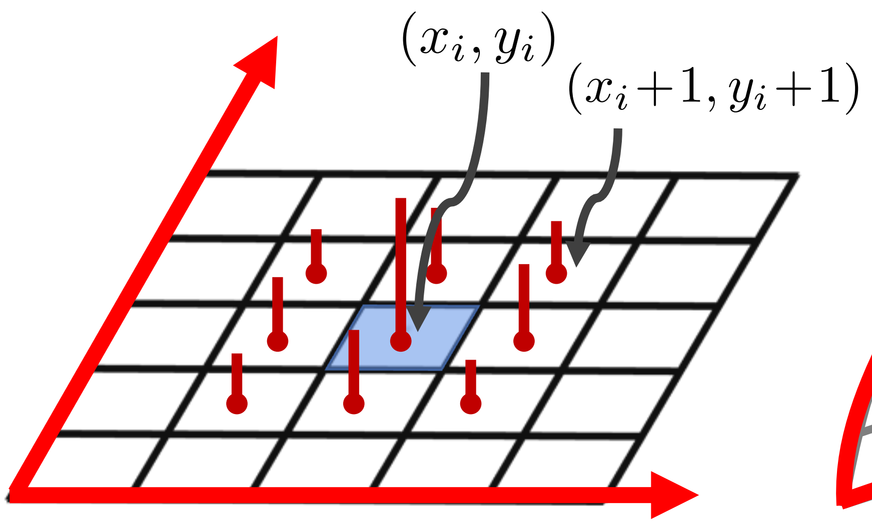
Fischl, Sereno, Tootell, Dale, HBM (1999)
Yeo, Sabuncu, Vercauteren, Ayache, Fischl, Golland, TMI (2010)
Lombaert, Ayache, **IPMI (2015)**
Lombaert, Sporring, Siddiqi, IPMI (2013)
Lombaert, Grady, Pennec, Ayache, Cheriet, **ECCV (2012), IJCV (2014)**
Lombaert, Grady, Polimeni, Cheriet, IPMI (2011), **PAMI (2012)**

Learning?

Moving Learning to the Spectral Domain

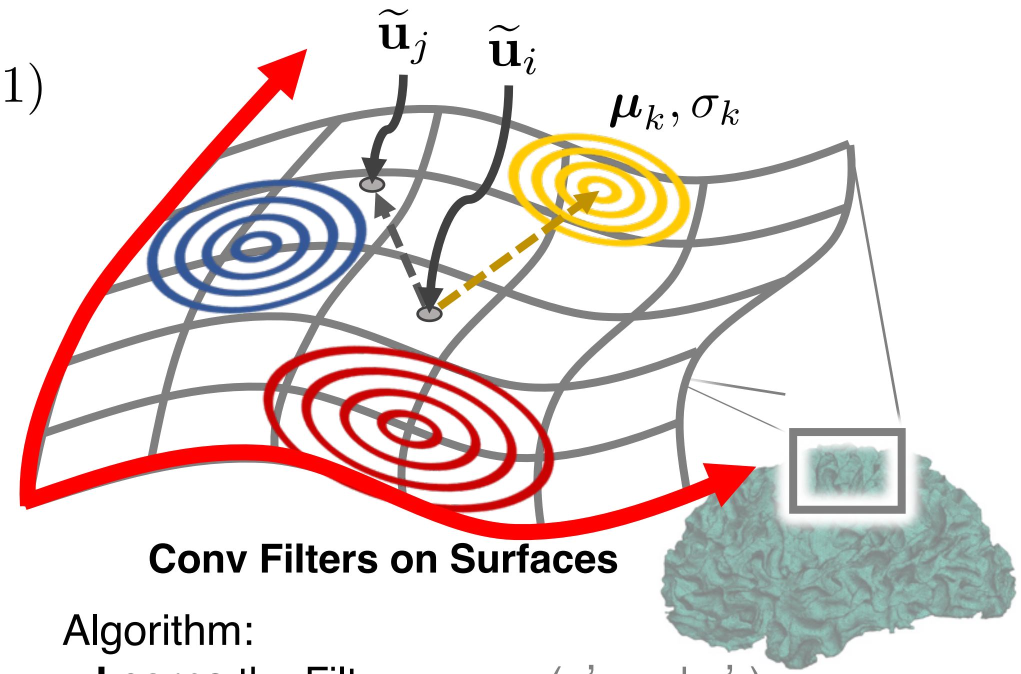
Convolutions on Surfaces

- Defining Kernels on **Curved Spaces**



Conv Filter on a Grid

Algorithm:
– **Learns** the Filter params (the red bars)
– Supposes neighbors are **on a grid**



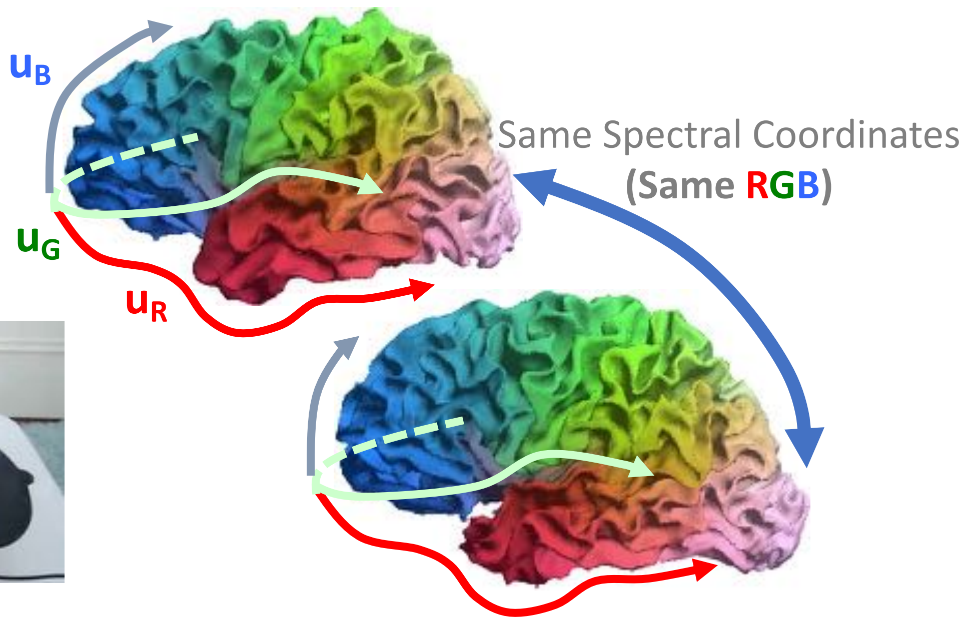
Conv Filters on Surfaces

Algorithm:
– **Learns** the Filter params (μ's and σ's)
– Requires **Graph Neighborhoods**

Intrinsic Shape Parameterization

Intrinsic Surface Parameterization

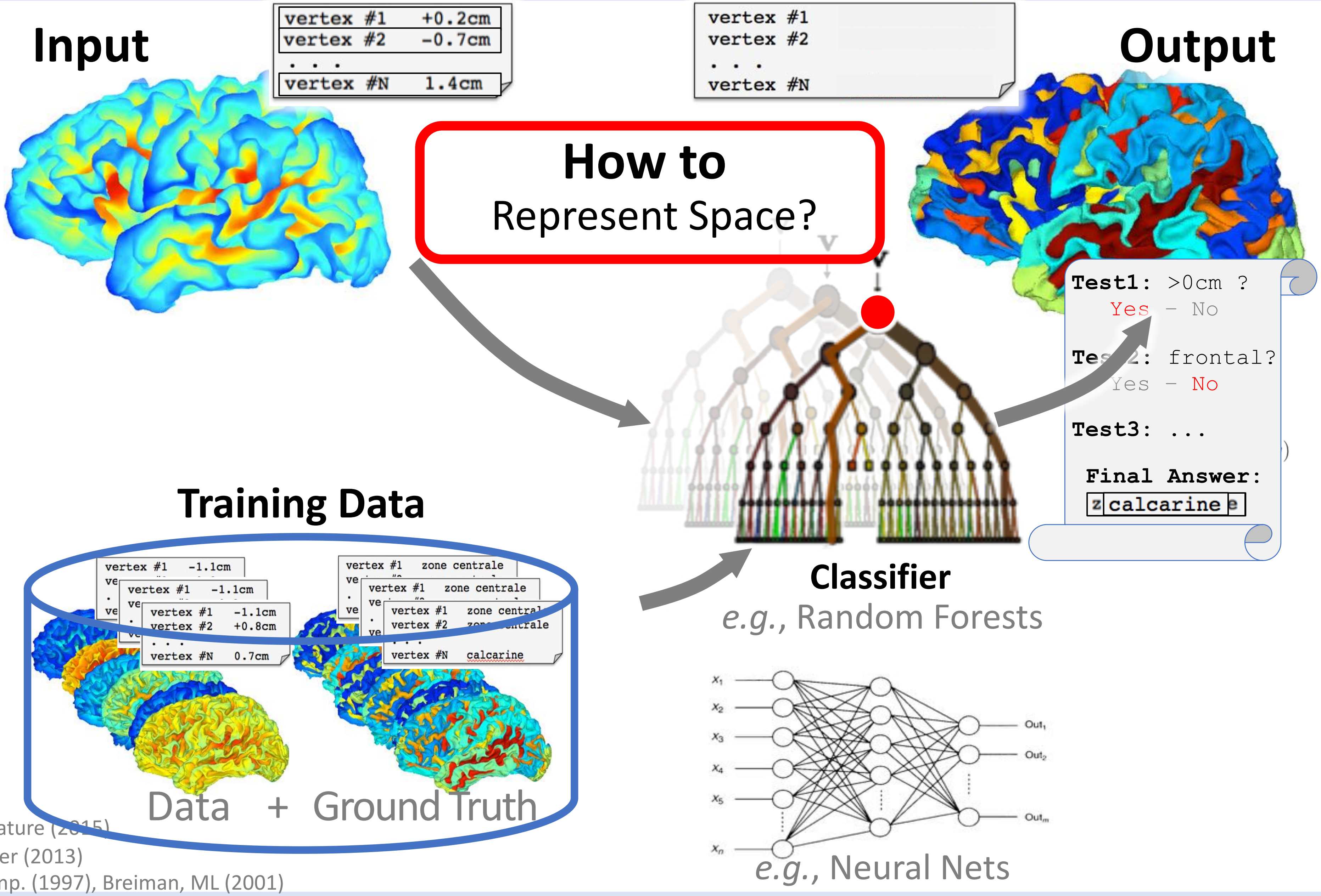
- **Spectral Coordinates**
  - an Intrinsic Surface Parameterization



Spectral Coordinates
Equivalent Points → **Similar Shape Characteristics**

Approach: Learning on Surfaces

[Lombaert MICCAI'15]

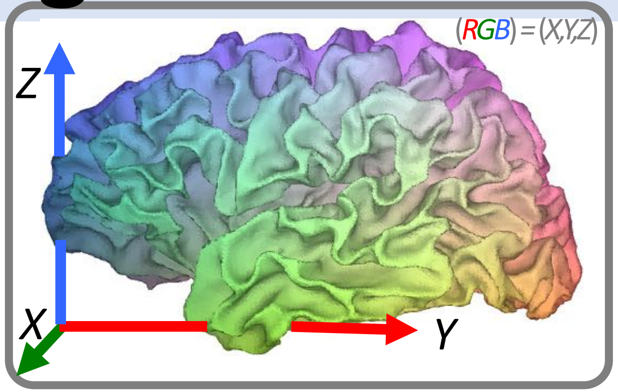


LeCun, Bengio, Hinton, Nature (2015)
Criminisi, Shotton, Springer (2013)
Amit, Geman, Neural Comp. (1997), Breiman, ML (2001)

Learning on Surfaces

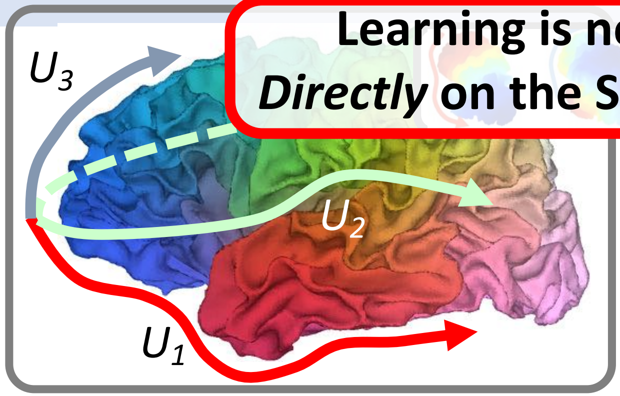
[Lombaert MICCAI'15]

Learning is now *Directly* on the Surface



Standard Euclidean Forests

Based on Euclidean Coordinates



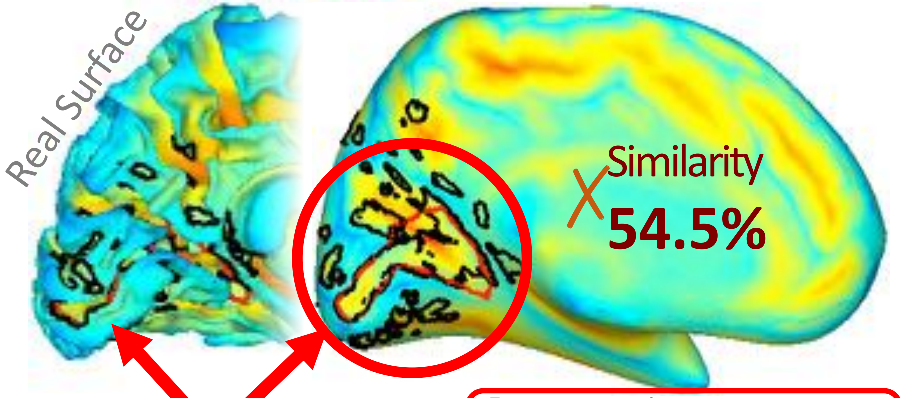
Spectral Forests

Spectral Coordinates **is Geometry Aware**

v = [Data, $\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}$] — **Simple Change** → v = [Data, $\boldsymbol{U_1},\boldsymbol{U_2},\boldsymbol{U_3}$]

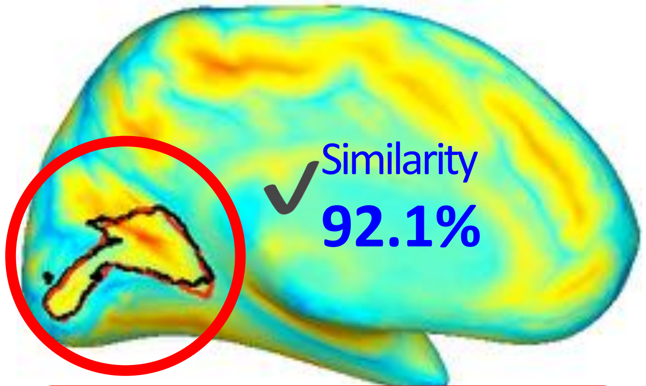
```
vertex #1   -1.1cm   (pos) 0.1, 12.4, 1.3
vertex #2   +0.8cm   (pos) 8.2,  1.3, 7.4
. . .
vertex #N   +0.7cm   (pos) 9.8, 19.7, 8.9
```

```
vertex #1   -1.1cm (s.pos) 0.7,-0.2, 0.3
vertex #2   +0.8cm (s.pos) 0.8,-0.1, 0.4
. . .
vertex #N   +0.7cm (s.pos) 0.9, 0.8, 0.9
```



Visual Cortex (Red)

Reason: Ignore Complex Geometry

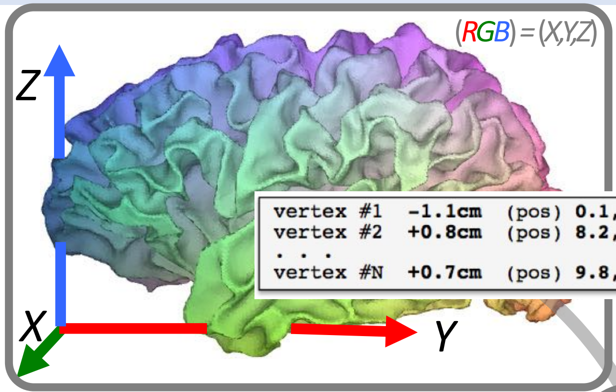


Reason: Learning exploits the **geometry** of the shape

Lombaert, Criminisi, Ayache, **MICCAI (2015)**

Application: Learning on Surfaces

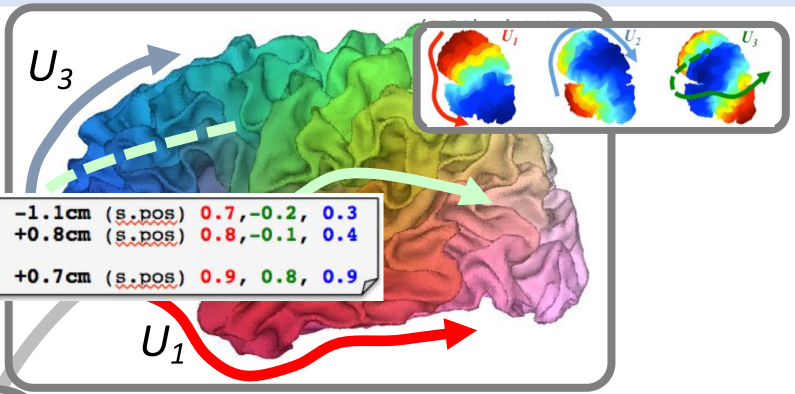
[Lombaert MICCAI'15]




```
vertex #1   -1.1cm  (pos) 0.1, 12.4, 1.3
vertex #2   +0.8cm  (pos) 8.2,  1.3, 7.4
. . .
vertex #N   +0.7cm  (pos) 9.8, 19.7, 8.9
```

Standard Forests

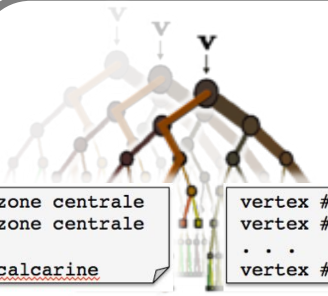
Spatial Representation is extrinsic




```
vertex #1   -1.1cm  (s.pos) 0.7,-0.2, 0.3
vertex #2   +0.8cm  (s.pos) 0.8,-0.1, 0.4
. . .
vertex #N   +0.7cm  (s.pos) 0.9, 0.8, 0.9
```

Spectral Forests

Learning is *Directly* on the surface




```
vertex #1   zone centrale
vertex #2   zone centrale
. . .
vertex #N   calcarine
```

```
vertex #1   zone centrale
vertex #2   zone centrale
. . .
vertex #N   calcarine
```

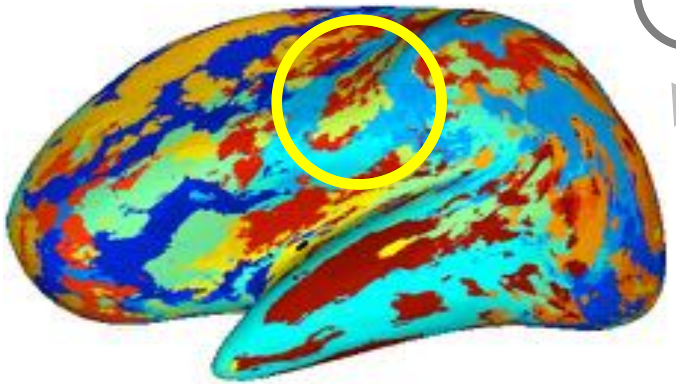
Classifier

Random Forests

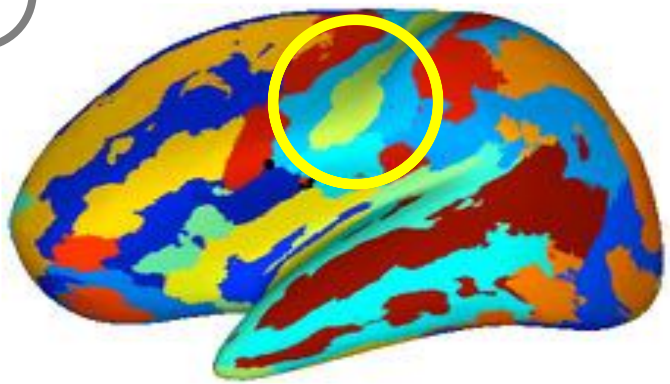
✗ **Ignore the Geometry**
(**Complex Shapes** of surfaces)

Now Geometry Aware
(**Learning directly** on surfaces)

Complete Segmentation
(77 regions)



Avg. Dice = **31.0%** (±15.5)
Avg. Dist. = 5.80mm (±4.24, max 38.02)



Avg. Dice = **77.6%** (±11.41)
Avg. Dist. = 2.02mm (±1.67, max 17.56)

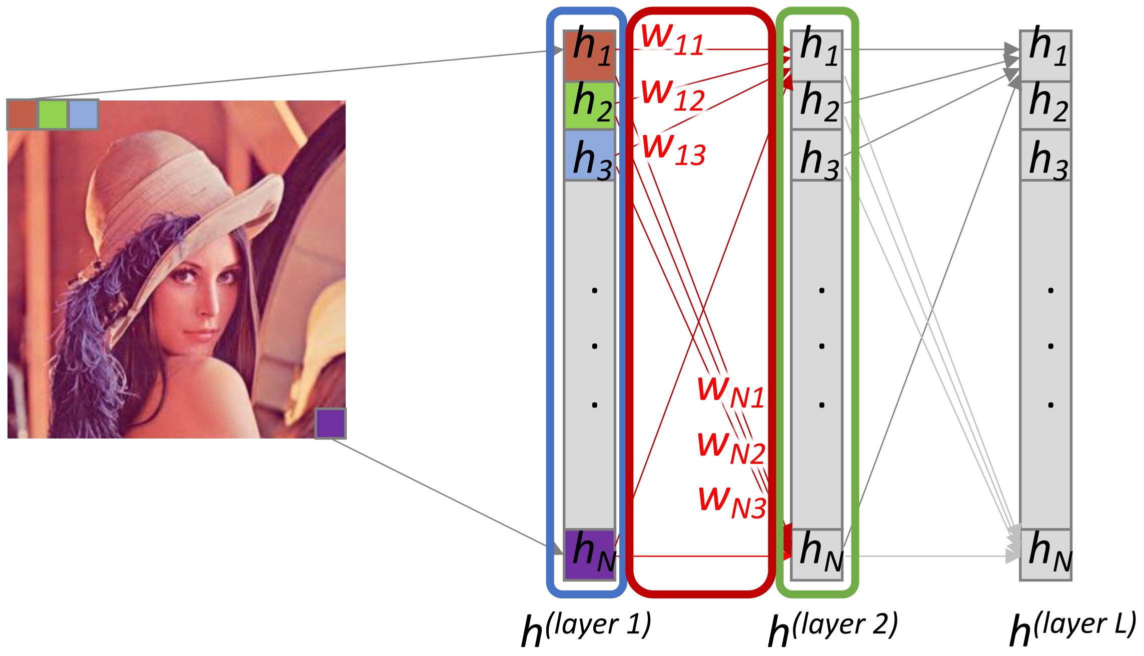
Lombaert, Criminisi, Ayache, **MICCAI (2015)**

1 2 3 4 5 6 7

Background on Geometric Deep Learning

How to Learn on Graph Node Data?

Neural Network on Images



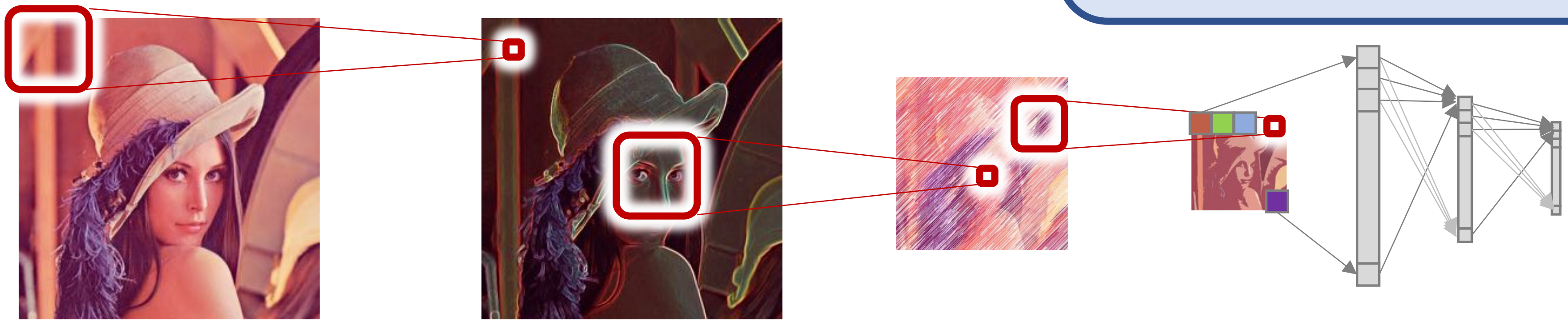
$$\boxed{h_i^{(l+1)}(x)} = \sigma\left(\sum_{j=1}^{N} \boxed{h_j^{(l)}(x)} \cdot \boxed{w_{ij}}\right)$$

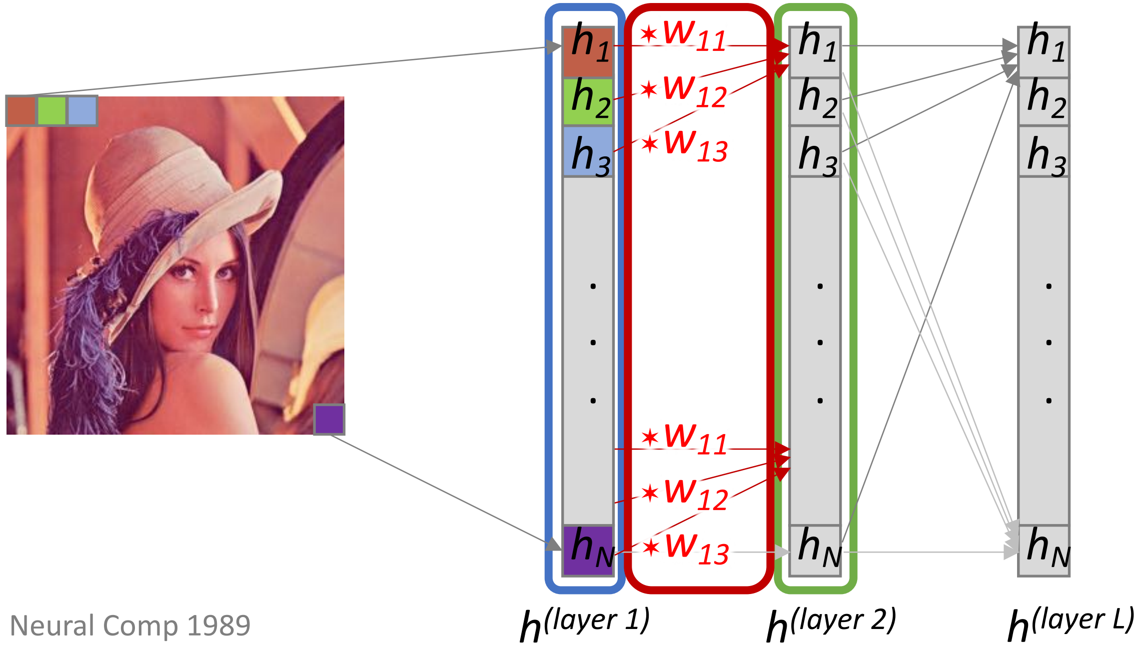
Problem if image content **moves**

✗ No invariance to translation

Convolutions on Images

One Solution: Let's move along the image
✓ Invariance to translation





Convolution

$$h_i^{(l+1)}(x) = \sigma\left(\sum_{j=1}^{N}\left(h_j^{(l)} \star w_{ij}\right)(x)\right)$$

Well defined on Images
How to do this on Graphs?

LeCun *et al*, Neural Comp 1989
Denkel *et al*, NeurIPS 1988
Fukushima *et al*, BioCyber 1980

Convolutions on Graphs

- **Remember**: Convolutions and **Fourier**
  - **Convolution** in Euclidean space ←→ **Multiplication** in **Fourier Space**

$$\mathcal{F}\{f \star g\} = \mathcal{F}\{f\} \cdot \mathcal{F}\{g\}$$

An image   A conv filter

How to use such **filters on Graph?**

Fourier, Théorie de la Chaleur, 1822

Spectral Convolutions on Graphs

- Approximation of **conv. filter** with Chebyshev Polynomials

$$\boxed{f \star g} = \mathcal{F}^{-1}\{\mathcal{F}\{f\} \cdot \mathcal{F}\{g\}\}$$

$$= \mathbf{\Phi}(\mathbf{\Phi}^{\mathbf{T}}\mathbf{g}) \odot (\mathbf{\Phi}^{\mathbf{T}}\mathbf{f})$$ In Fourier Space, matrix notation

$$= \mathbf{\Phi}\,\mathrm{diag}(\mathcal{F}\{g\})\,\mathbf{\Phi}^{\mathbf{T}}\mathbf{f}$$

$\mathrm{diag}(\mathcal{F}\{g\})$ expressed in terms of λ

$$= \mathbf{\Phi}\,\mathrm{diag}(\mathcal{F}\{g(\lambda)\})\,\mathbf{\Phi}^{\mathbf{T}}\mathbf{f}$$

$\mathcal{F}\{g(\lambda)\}$ approximated with Chebyshev Polynomials: $\mathcal{F}\{g(\lambda)\} = \sum_{k=0}^{K} \theta_k T_k(\lambda)$

$$\approx \mathbf{\Phi}\,\mathrm{diag}\left(\sum_{k=0}^{K} \theta_k T_k(\lambda)\right)\mathbf{\Phi}^{\mathbf{T}}\mathbf{f}$$

insert L with $U\hat{g}(\lambda)U^T = \hat{g}(U\lambda U^T) = \hat{g}(L)$

$$= \sum_{k=0}^{K} \theta_k T_k(L)\,\mathbf{f}$$

simplification with First-order Chebyshev

$$\approx \theta_0\,\mathbf{f} - \theta_1\, D^{-\frac{1}{2}} A D^{-\frac{1}{2}}\,\mathbf{f}$$

simplification with single parameter

$$\theta = \theta_0 = -\theta_1$$

$$\boxed{\approx \theta \left(I + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}\right)\mathbf{f}}$$

Easy Convolution **on Graphs**

Hammond *et al*, Harmonic Anal 2011
Defferrard *et al*, NeurIPS 2016
Kipf, Welling, ICLR 2017
Bruna *et al*, ICML 2014
Duvenaud *et al*, NeurIPS 2015
Levie *et al*, ICLR 2018

Spectral Convolutions on Graphs

- Simple Convolution via Graph Laplacians

$$\boxed{f \star g =} \sum_{k=0}^{K} \theta_k T_k(L)\,\mathbf{f}$$

$$\approx \theta_0\,\mathbf{f} - \theta_1\, D^{-\frac{1}{2}} A D^{-\frac{1}{2}}\,\mathbf{f}$$

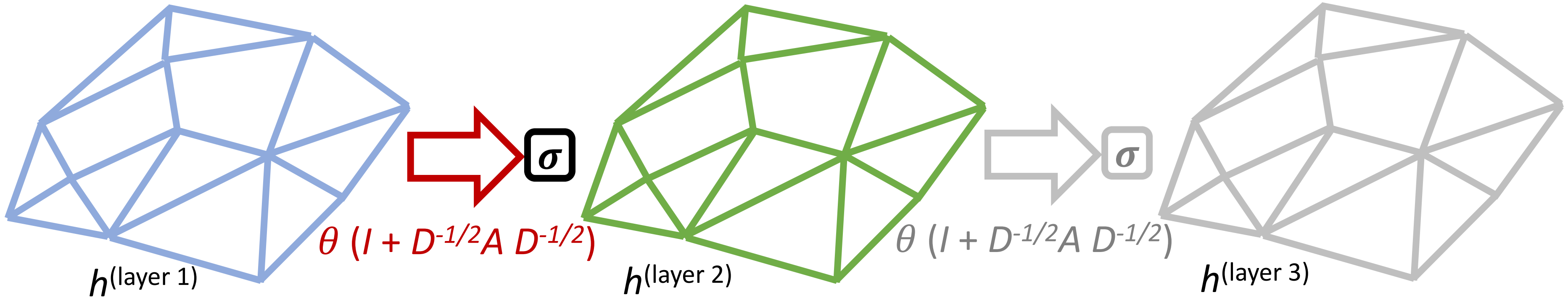
$$\boxed{\approx \theta\left(I + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}\right)\mathbf{f}}$$

Easy Convolution on Graphs

$$h_i^{(l+1)}(x) = \sigma\left(\sum_{j=1}^{N}\left(h_j^{(l)} \star w_{ij}\right)(x)\right)$$

$$\boxed{h_i^{(l+1)}(x)} = \sigma\left(\theta\, h_i^{(l)}(x) + \boxed{\theta\,\frac{1}{\sqrt{d_i}}\sum_{j\in N_i} a_{ij}\,\frac{1}{\sqrt{d_i}}}\,\boxed{h_j^{(l)}(x)}\right)$$

Simple Convolution Layer



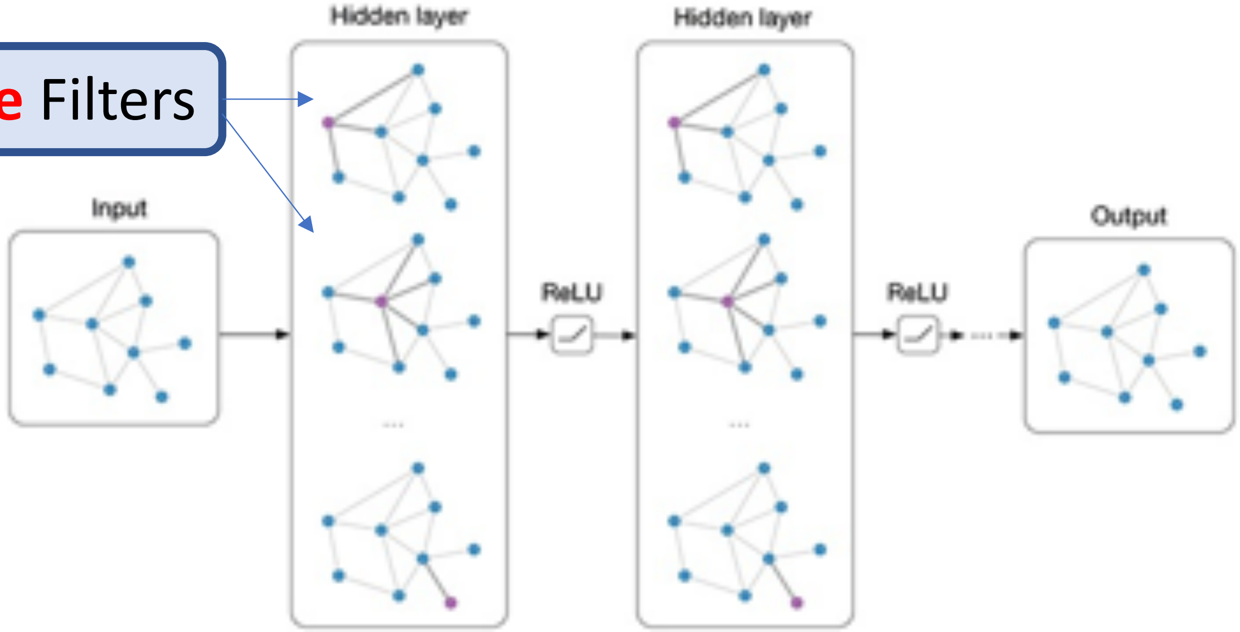
Too Simple for Fun Kernels?

Hammond *et al*, Harmonic Anal 2011
Defferrard *et al*, NeurIPS 2016
Kipf, Welling, ICLR 2017
Bruna *et al*, ICML 2014
Duvenaud *et al*, NeurIPS 2015
Levie *et al*, ICLR 2018

Spectral Convolutions **on Graphs**

- Exploits **Graph Laplacian** and Convolutions over **Graph Neighbors**

Use **Multiple** Filters



[Image Courtesy: Kipf, GCN, 2016]

Problem:

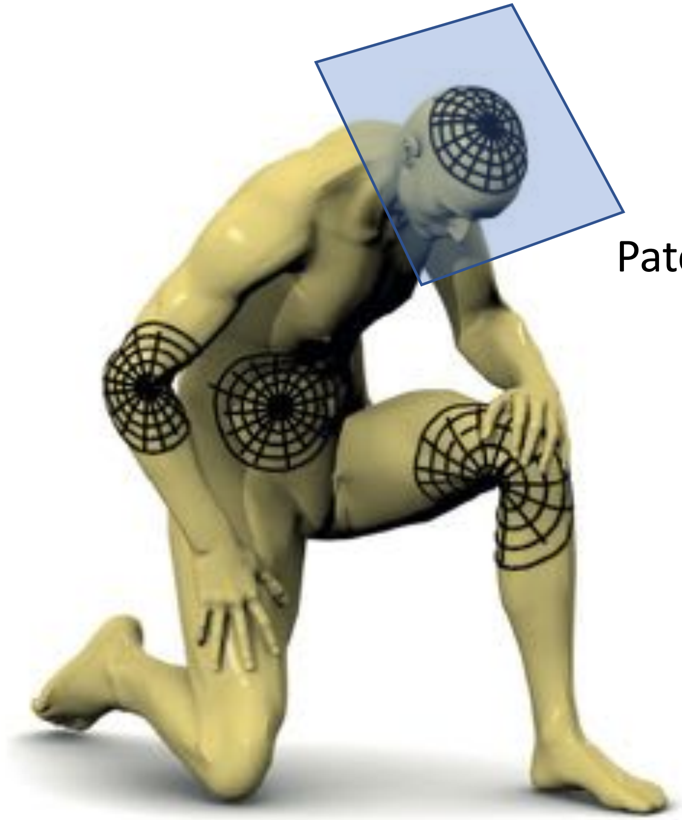
- **Need for Richer kernels**
- **Constrained** to **Fixed Graph Structure**

Hammond *et al*, Harmonic Anal 2011
Defferrard *et al*, NeurIPS 2016
Kipf, Welling, ICLR 2017
Bruna *et al*, ICML 2014
Duvenaud *et al*, NeurIPS 2015
Levie *et al*, ICLR 2018

Spatial Convolutions on Graphs

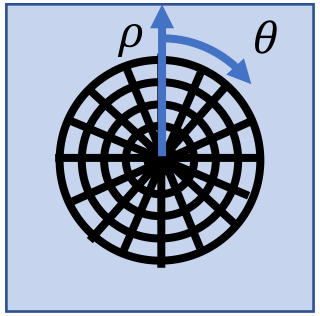
- Richer **Filters** on Tangent Planes of Manifolds

How to Parameterize Kernels ?

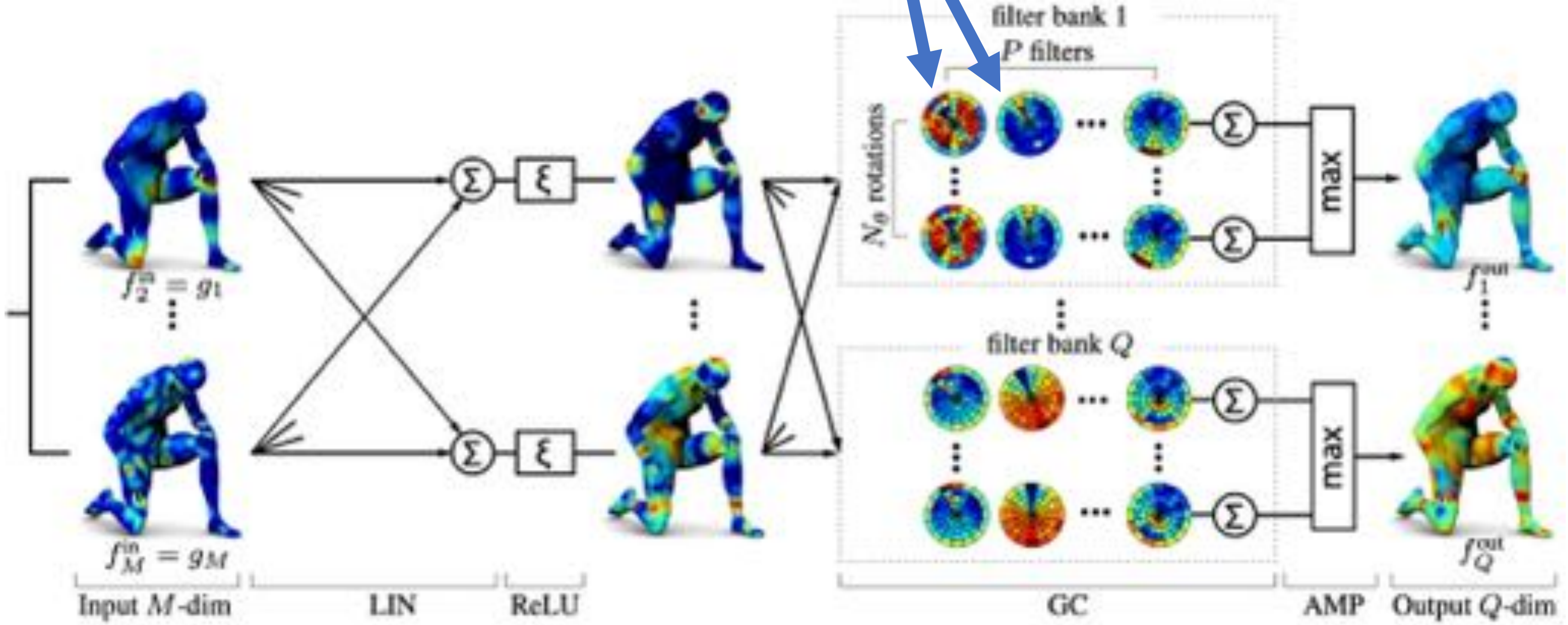


[Image Courtesy: Monti, MoNet, 2017]

Patch Operator
D(x)f



Learn Filter Weights in
Polar Coordinates



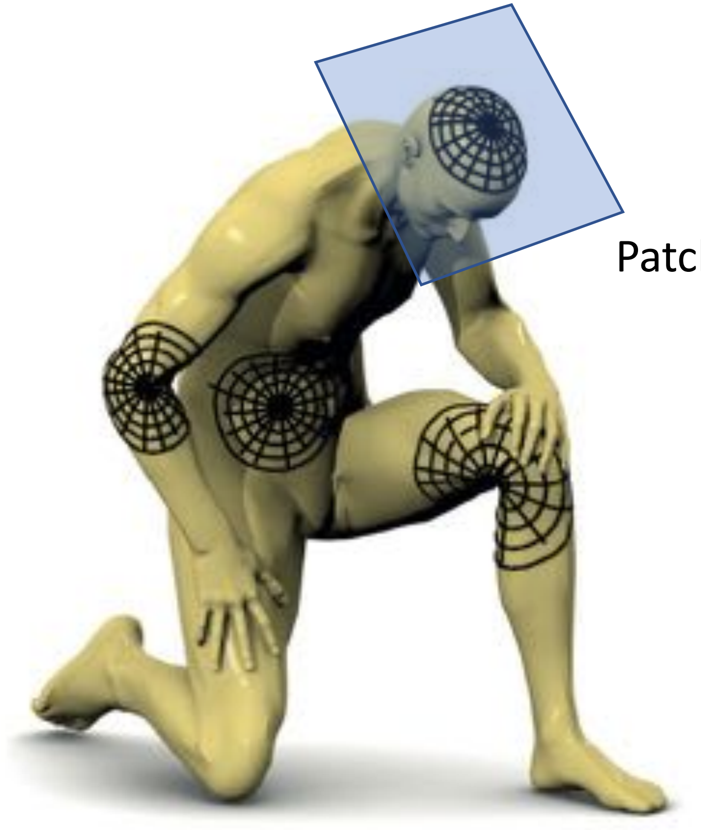
[Image Courtesy: Masci, Geodesic CNN, 2015]

Kokkinos *et al*, CVPR 2012
Masci *et al*, 3dRR 2015
Boscaini *et al*, NeurIPS 2016
Monti *et al*, CVPR 2017
Fey *et al*, CVPR 2018

Spatial Convolutions on Graphs

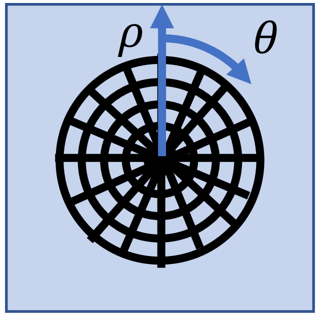
- Richer **Kernels** on Tangent Planes of Manifolds

Patch Orientation?
Patch Construction?



[Image Courtesy: Monti, MoNet, 2017]

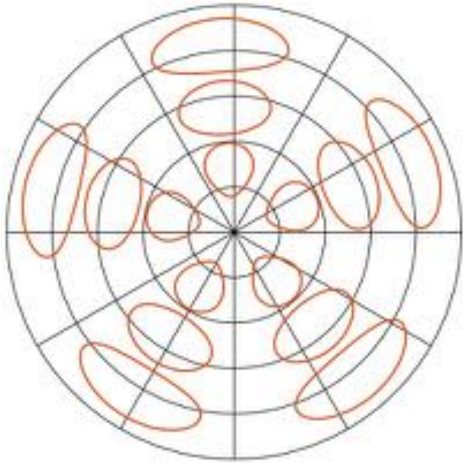
Patch Operator
D(x)f



Learn Kernel Parameters

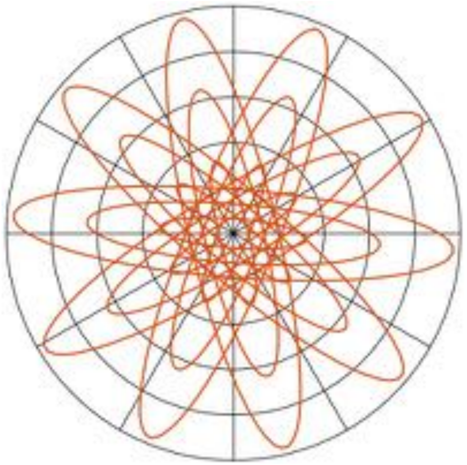
$$h_i^{(l+1)}(x) = \sigma\left(\sum_{j=1}^{\mathcal{N}_i} h_j^{(l)} \star \boxed{w_{ij}}(x)\right)$$

Richer Kernels on Shapes



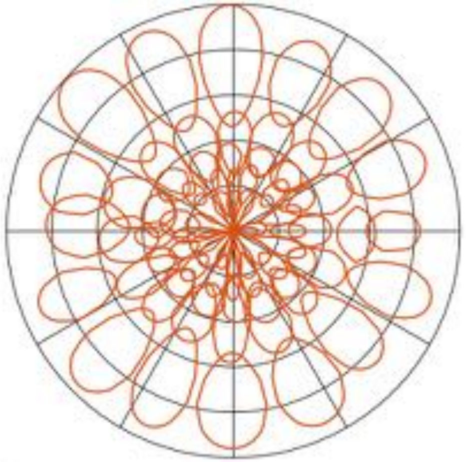
$$\boxed{w(\boldsymbol{u})} = \exp\left(-\frac{1}{2}(\boldsymbol{u}-\boldsymbol{\mu}_{\rho,\theta})^T \begin{pmatrix} \sigma_\rho & \\ & \sigma_\theta \end{pmatrix}^{-1} (\boldsymbol{u}-\boldsymbol{\mu}_{\rho,\theta})\right)$$

Geodesic CNN
[Masci *et al*, 3dRR 2015]



$$\boxed{w(\boldsymbol{u})} = \exp\left(-t\boldsymbol{u}^T \boldsymbol{R} \begin{pmatrix} \alpha & \\ & 1 \end{pmatrix}^{-1} \boldsymbol{R}^T \boldsymbol{u}\right)$$

Anisotropic GCNN
[Boscaini *et al*, NeurIPS 2016]



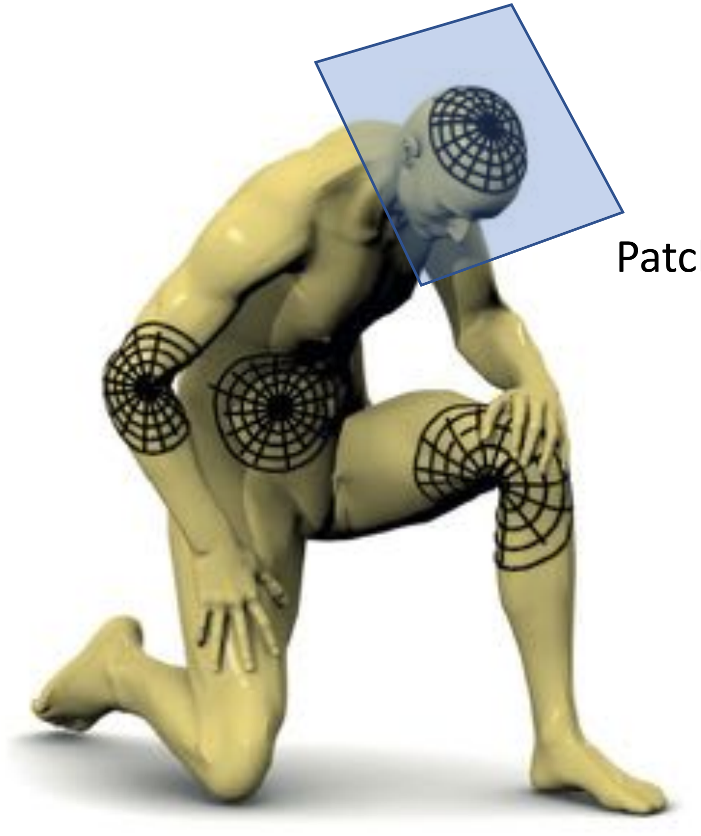
$$\boxed{w(\boldsymbol{u})} = \exp\left(-\frac{1}{2}(\boldsymbol{u}-\boxed{\boldsymbol{\mu}_{\rho,\theta}})^T \boxed{\Sigma_{\rho,\theta}^{-1}} (\boldsymbol{u}-\boldsymbol{\mu}_{\rho,\theta})\right)$$

Mixture of Gaussians GCNN
[Monti *et al*, CVPR 2017]

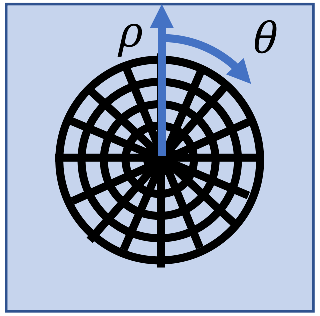
Kokkinos *et al*, CVPR 2012
Masci *et al*, 3dRR 2015
Boscaini *et al*, NeurIPS 2016
Monti *et al*, CVPR 2017
Fey *et al*, CVPR 2018

Spatial Convolutions on Graphs

- Construction of Polar Patches



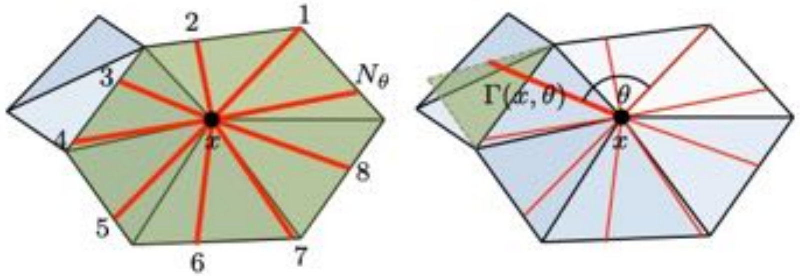
Patch Operator
D(x)f



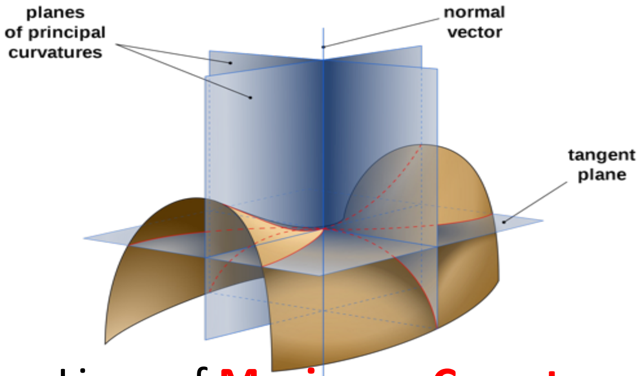
Costly, Arbitrary?

Patch Construction?

Patch Orientation?



Geodesic Polylines across triangles



Lines of **Maximum Curvature**

[Image Courtesy: Monti, MoNet, 2017]

Kokkinos *et al*, CVPR 2012
Masci *et al*, 3dRR 2015
Boscaini *et al*, NeurIPS 2016
Monti *et al*, CVPR 2017
Fey *et al*, CVPR 2018

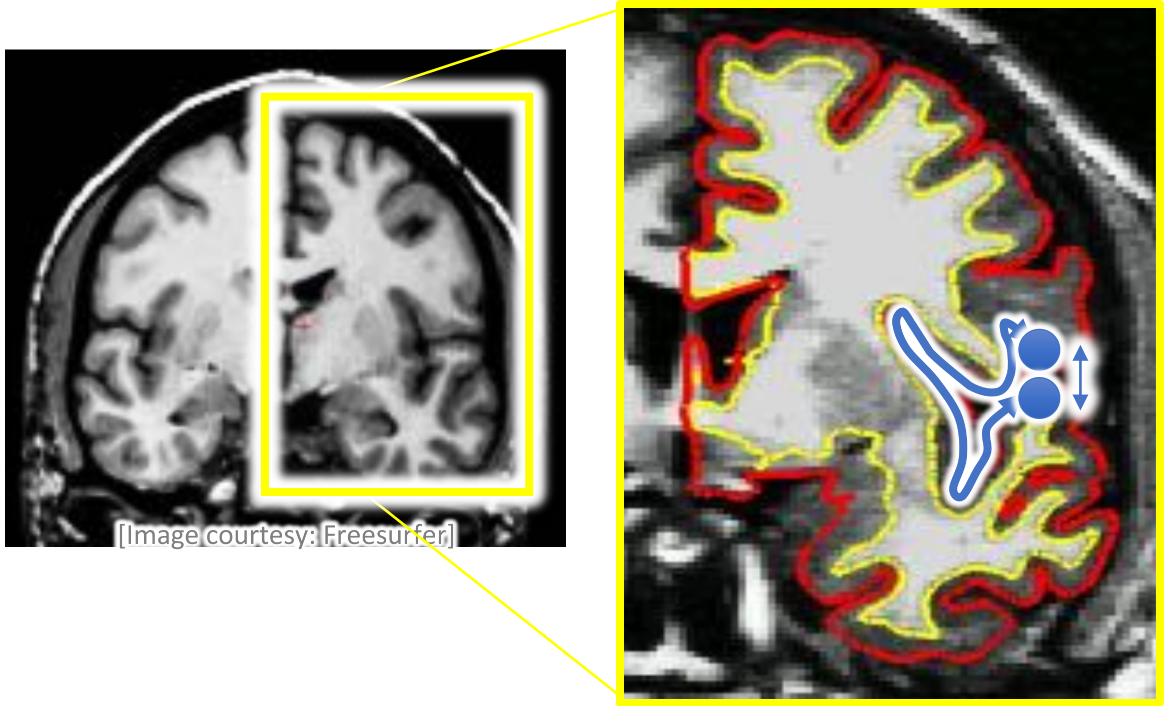
1 2 3 4 5 6 7

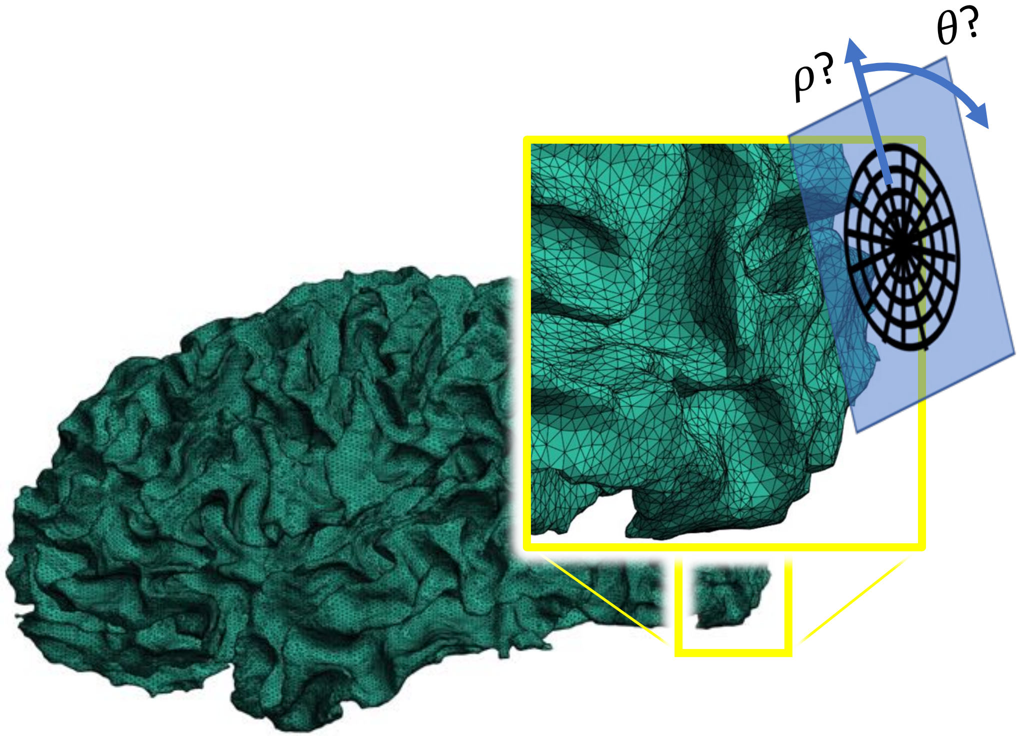
Limitations of Geometric Deep Learning

What is preventing Generalization to Arbitrary Surfaces?

Challenges in Medical Imaging

- **Geometrical Complexity** of Surfaces





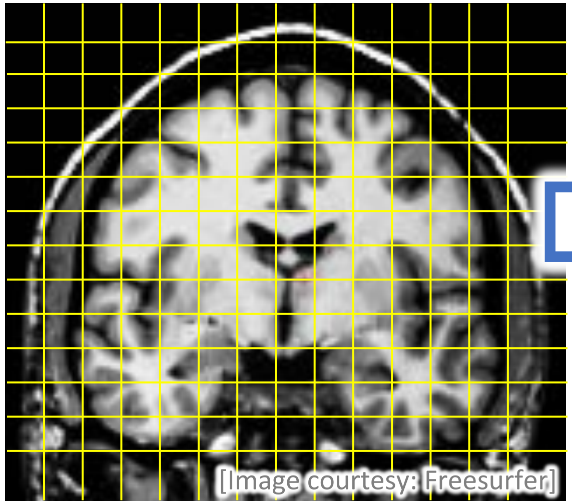
Surfaces – How to Create & Navigate patches
(where is 'up' in a sulcus?)

Problem – Convolutions in Image Space
- **Distance Ambiguity**
- Volumes –*vs*– **Surfaces**
- **Confusing** for Learning **Algorithms**

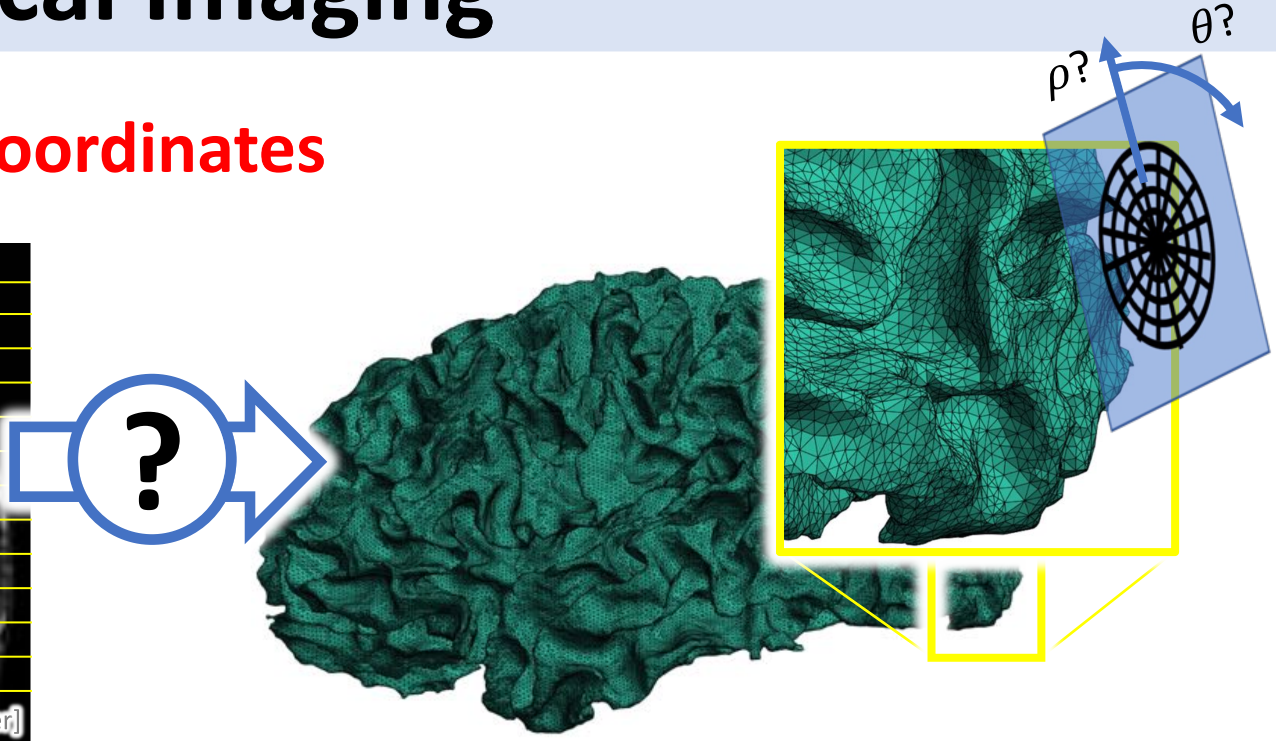
Problem – Convolutions in Mesh Space
- **Patch construction**
- **Highly folded** surfaces
- **Confusing** for Learning **Filters**

Challenges in Medical Imaging

- Representation of **Mesh Coordinates**



Point Coordinates
defined as (x,y,z) Coordinates

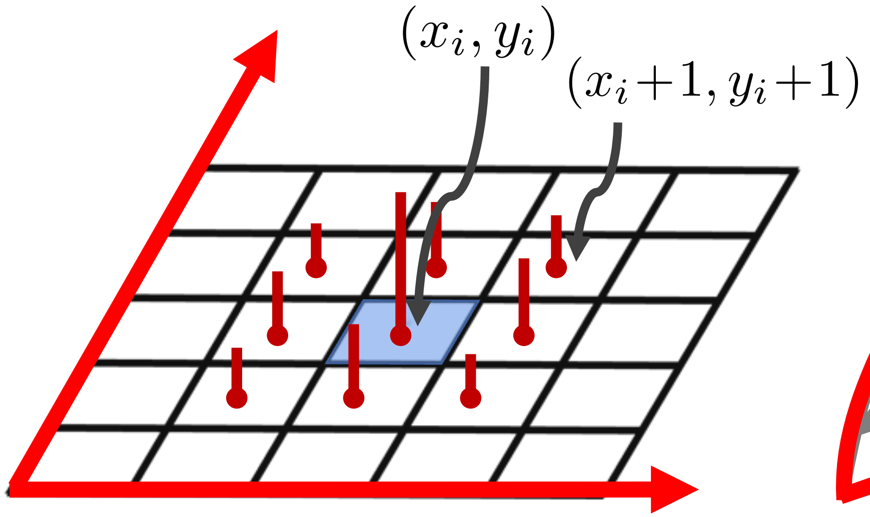


Mesh Coordinates?
(x,y,z); (ρ, θ) inadequate in Euclidean Space

Mesh Coordinates
Inadequate in Euclidean Space

Challenges in Medical Imaging

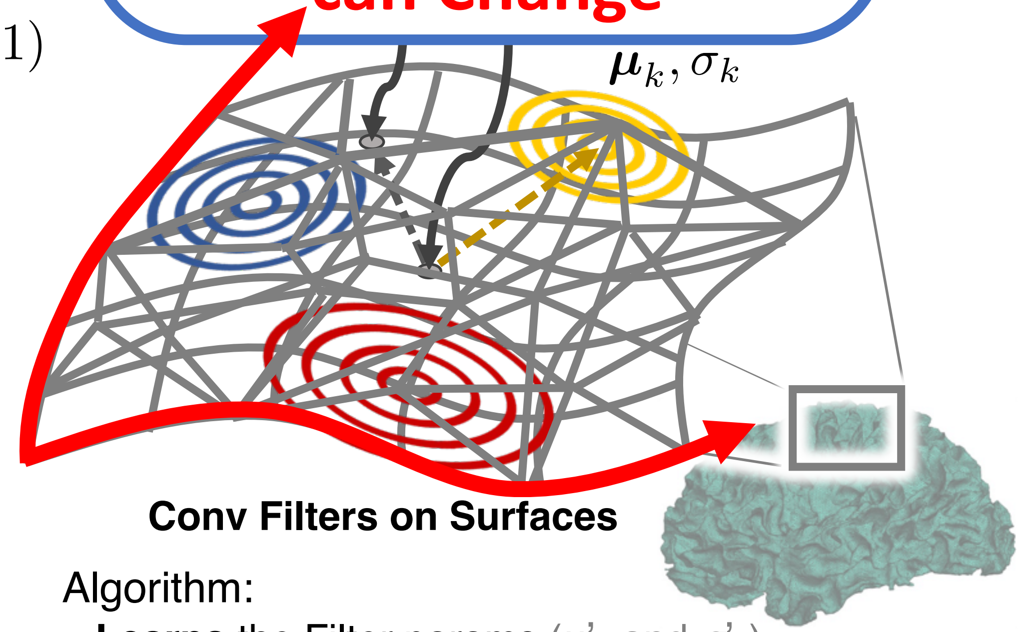
- Varying Mesh **Triangulations**



Conv Filter on a Grid

Algorithm:
– **Learns** the Filter params (the red bars)
– Supposes neighbors are **on a grid**

The Graph Structure can Change



Conv Filters on Surfaces

Algorithm:
– **Learns** the Filter params (μ's and σ's)
– Requires **Graph Neighborhoods**

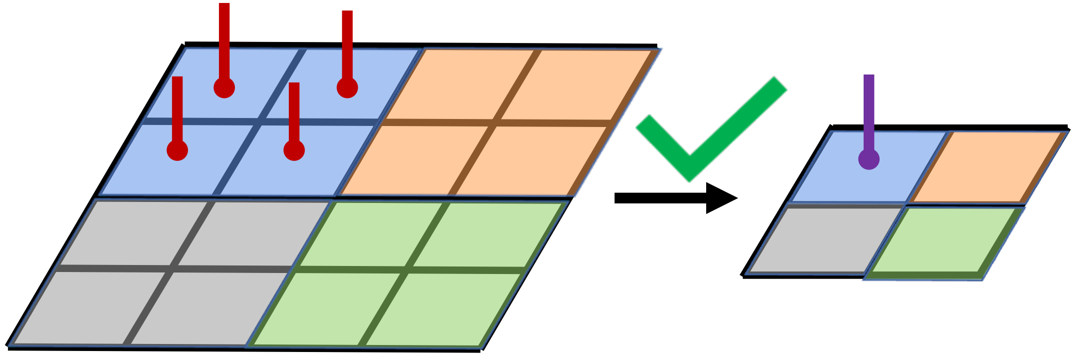
How to Parameterize a Brain Surface?

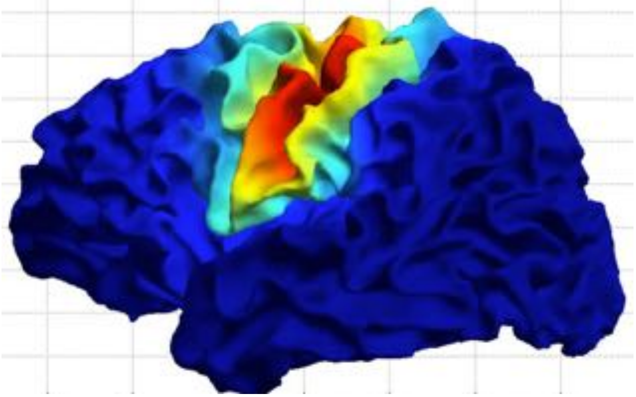
Challenge – Images vs Surfaces

Convolution

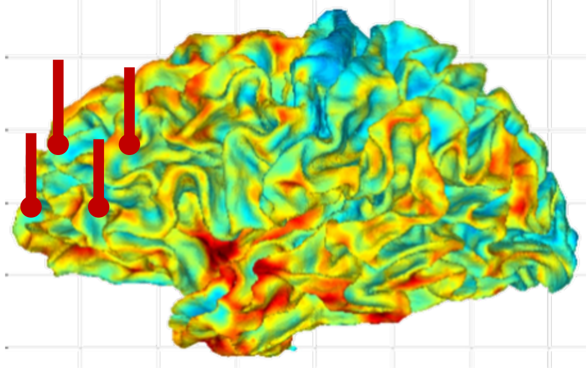
Pooling

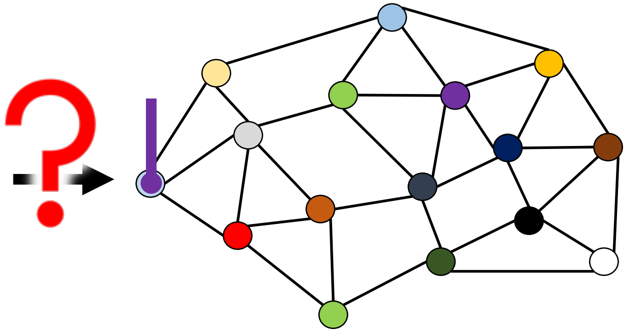
on images





on surfaces



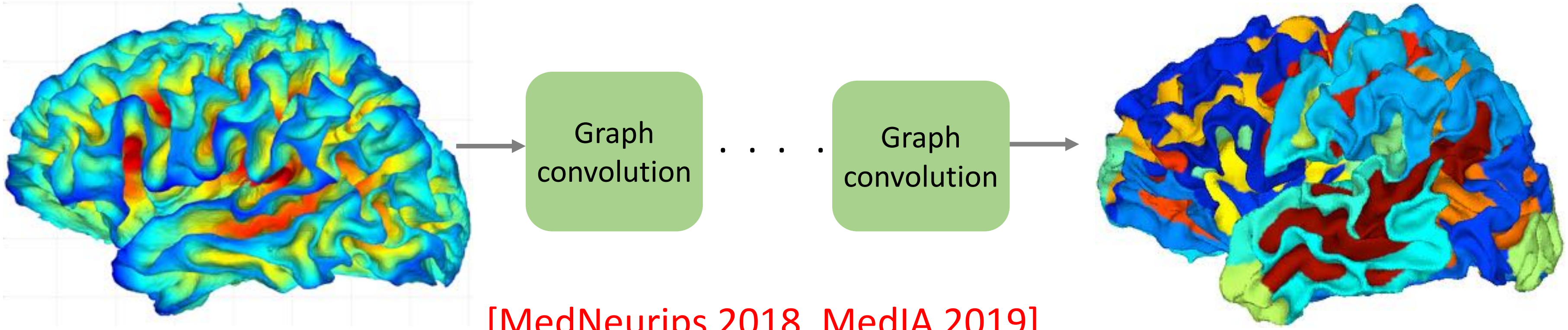


How to learn on surfaces?

Graph Networks – Two Contributions

Graph Convolutions on Spectral Embeddings for Cortical Surface Parcellation

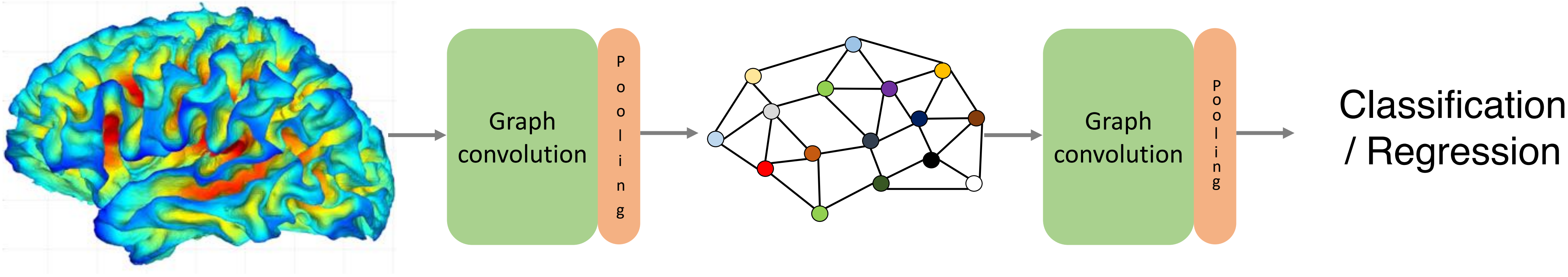
(1)



[MedNeurips 2018, MedIA 2019]

Learnable Pooling in Graph Convolutional Networks for Brain Surface Analysis

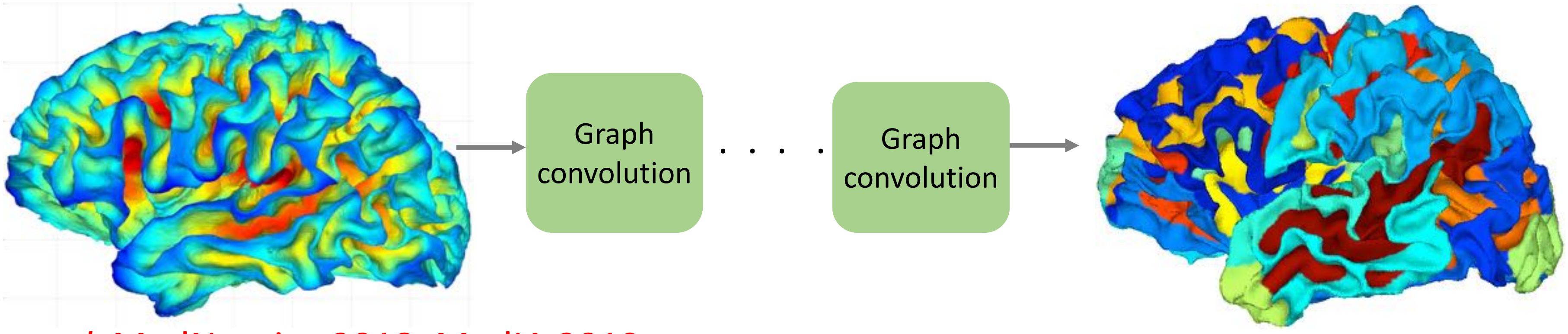
(2)



[IPMI 2019, PAMI 2021]

One Contribution: Localized Graph Convolutions

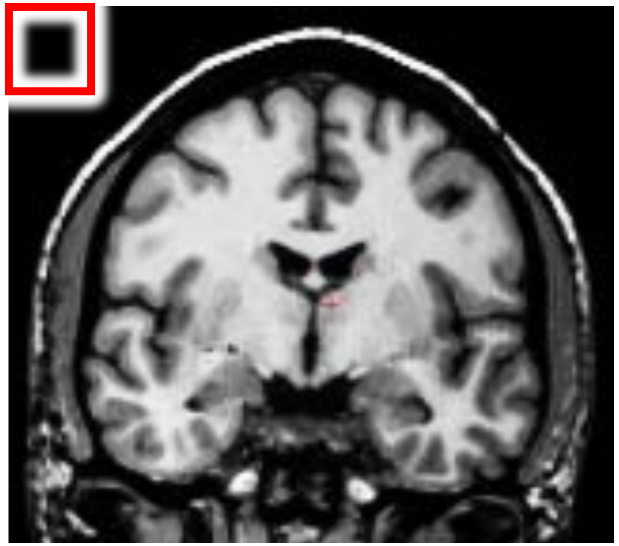
How to **Navigate Graph** Convolutions on Arbitrary Surfaces?



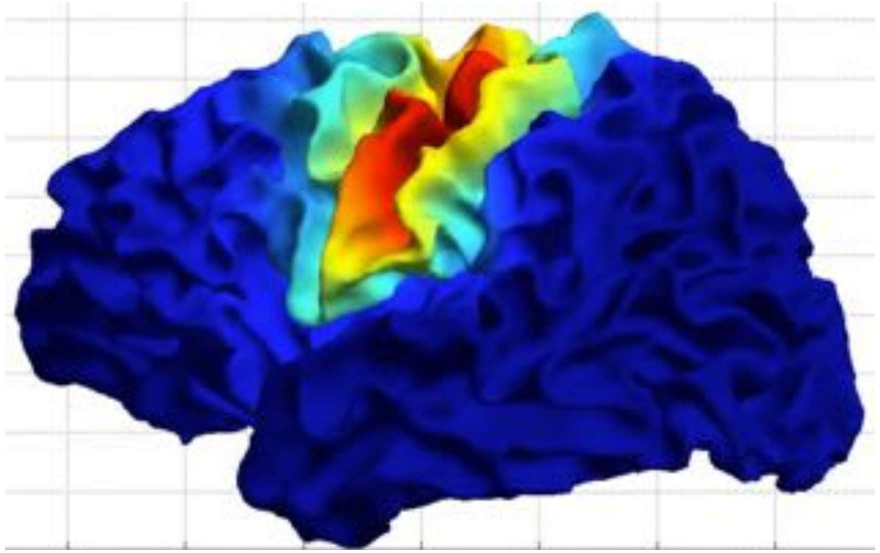
Gopinath *et al,* MedNeurips 2018, MedIA 2019

Convolutions on Surfaces

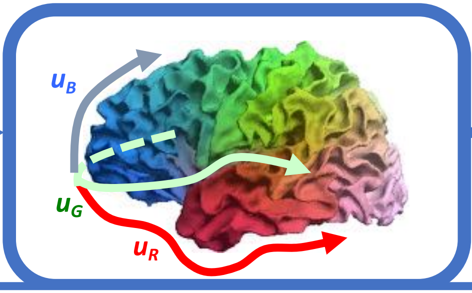
- Convolutions on Spectral Embeddings



Convolution on an Image

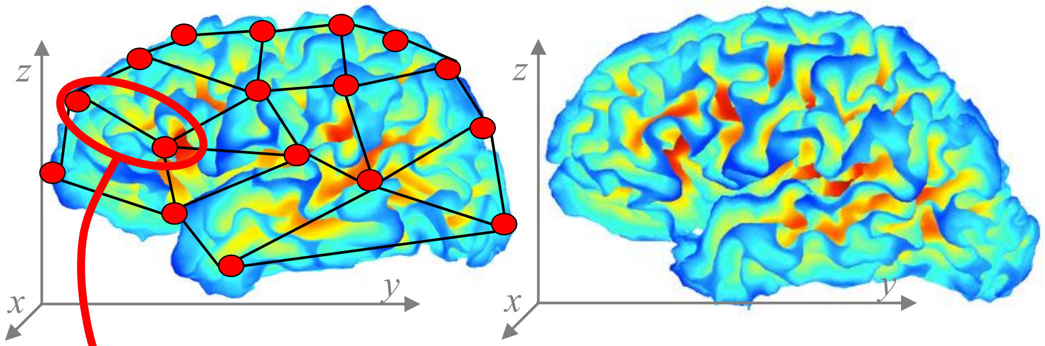


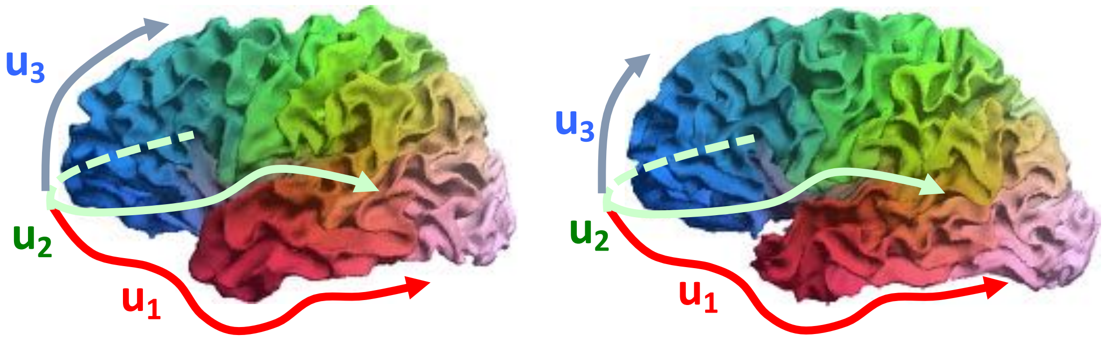
Graph Convolution on a Brain Surface



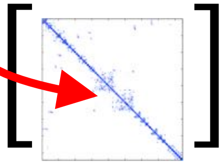
Euclidean → Spectral Coordinates

Spatial Information as Spectral Encoding



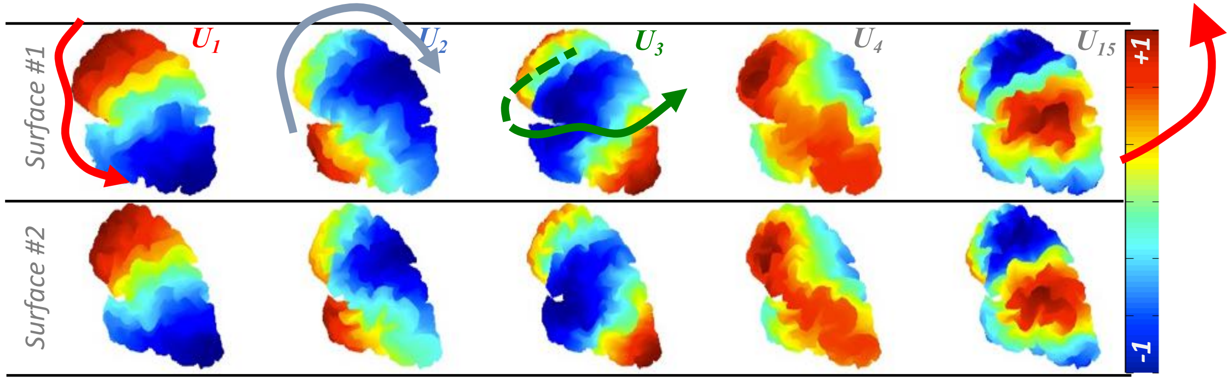


$L =$



Graph Laplacian

Edge (i,j) : Row/Column (i,j)



Spectral Decomposition (Eigenvectors of **L**)

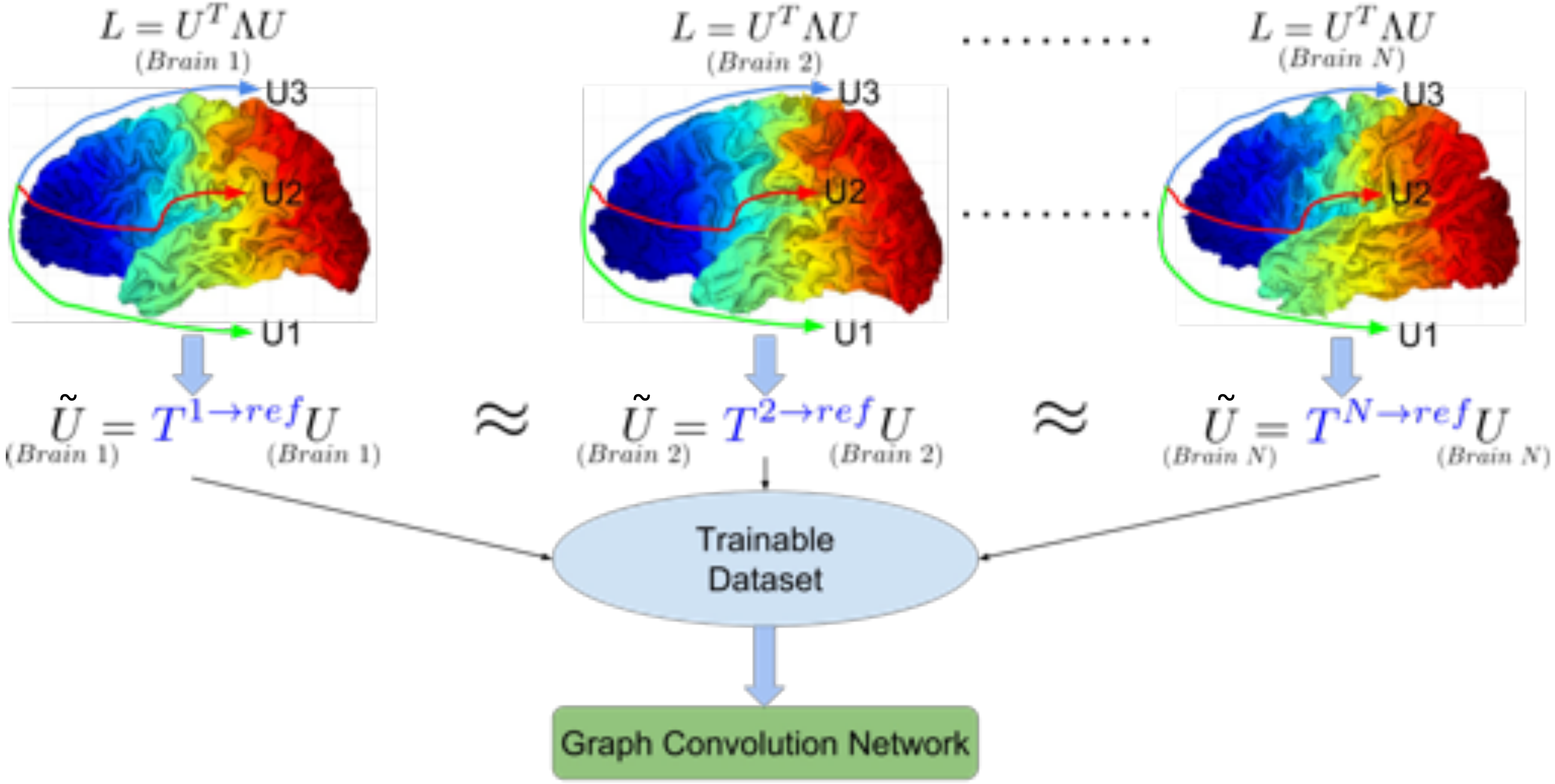
Spectral Decomposition

$$L = U^T \Lambda U$$

Problem: Spectral bases are **ambiguous to rotation**

Lombaert *et al*, IPMI 2015

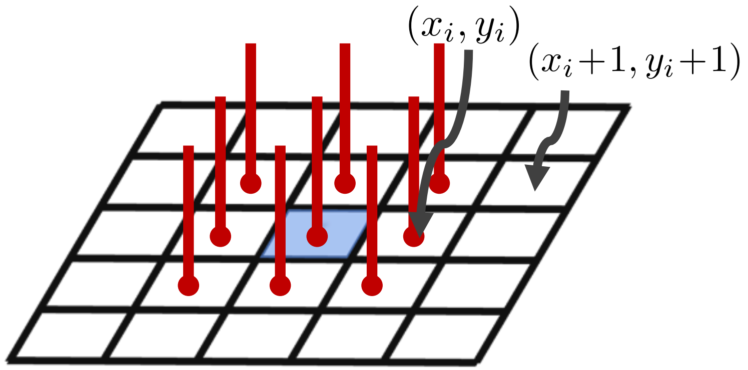
Spectral Alignment



Lombaert *et al*, IPMI 2015

Extension of 2D convolutions to irregular grids

Standard convolution on regular grid:



Grid space

Convolution kernel weights

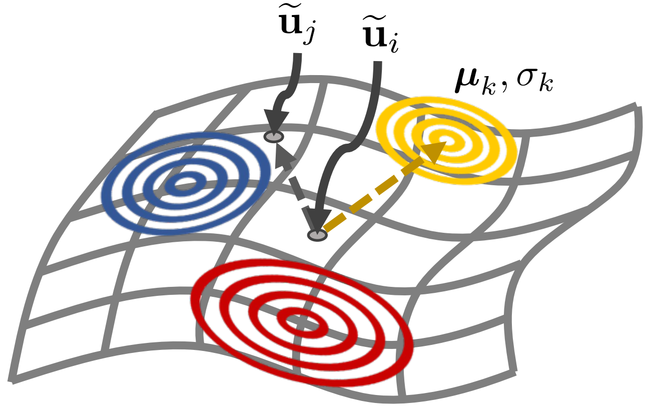
$$z_{ip}^{(l)} = \sum_{q=1}^{M_l} \sum_{k=-K_l}^{K_l} w_{pqk}^{(l)} \cdot y_{i+k,q}^{(l)} + b_p^{(l)}$$

Input feature map

$$y_{ip}^{(l+1)} = f(z_{ip}^{(l)})$$

Non-linear activation (ReLU)

Geometric convolution for embedded graphs:



Graph embedding space

Neighbor nodes on mesh

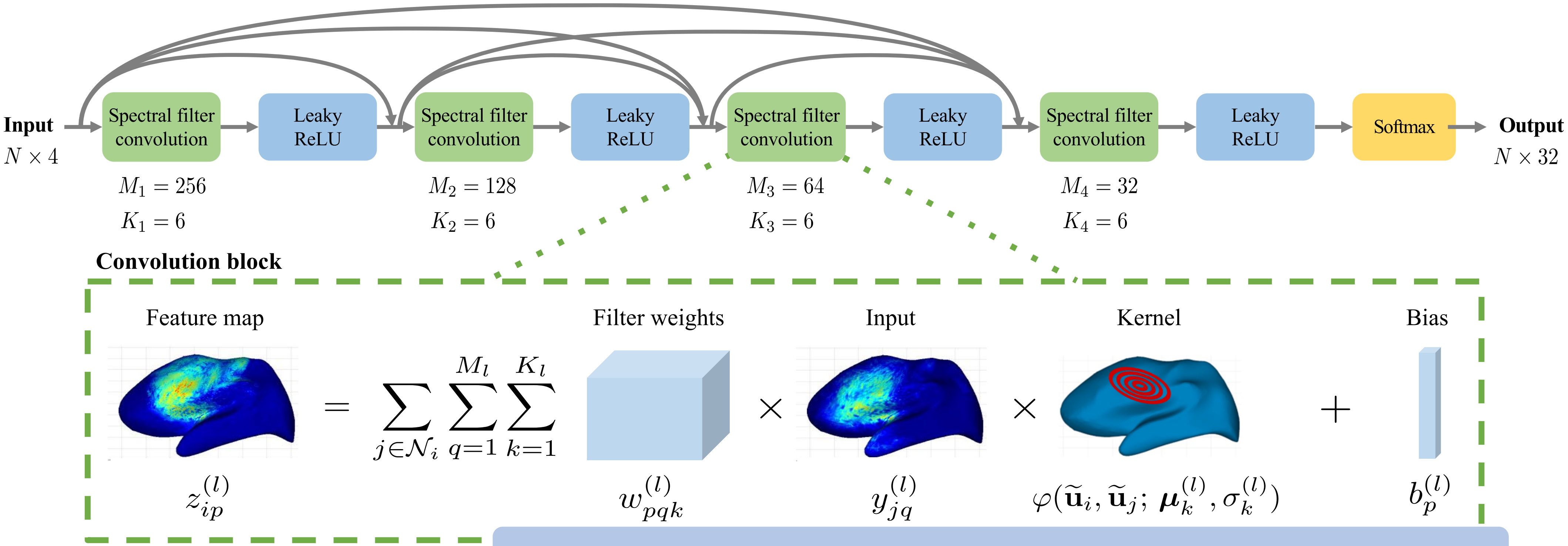
$$z_{ip}^{(l)} = \sum_{j \in \mathcal{N}_i} \sum_{q=1}^{M_l} \sum_{k=1}^{K_l} w_{pqk}^{(l)} \cdot y_{jq}^{(l)} \cdot \varphi(\widehat{\mathbf{u}}_i, \widehat{\mathbf{u}}_j; \boldsymbol{\Theta}_k^{(l)}) + b_p^{(l)}$$

Parameters are **learned**

$$\varphi(\widehat{\mathbf{u}}_i, \widehat{\mathbf{u}}_j; \boldsymbol{\mu}_k, \sigma_k) = \exp\left(- \sigma_k \left\| (\widehat{\mathbf{u}}_j - \widehat{\mathbf{u}}_i) - \boldsymbol{\mu}_k \right\|^2 \right)$$

Spectral Graph Conv Net – Architecture

- Enables **classical architectures** on brain surfaces
  - Operating **in the Spectral Domain** (not the grid Domain)



Features and Filters are now in the Spectral Domain

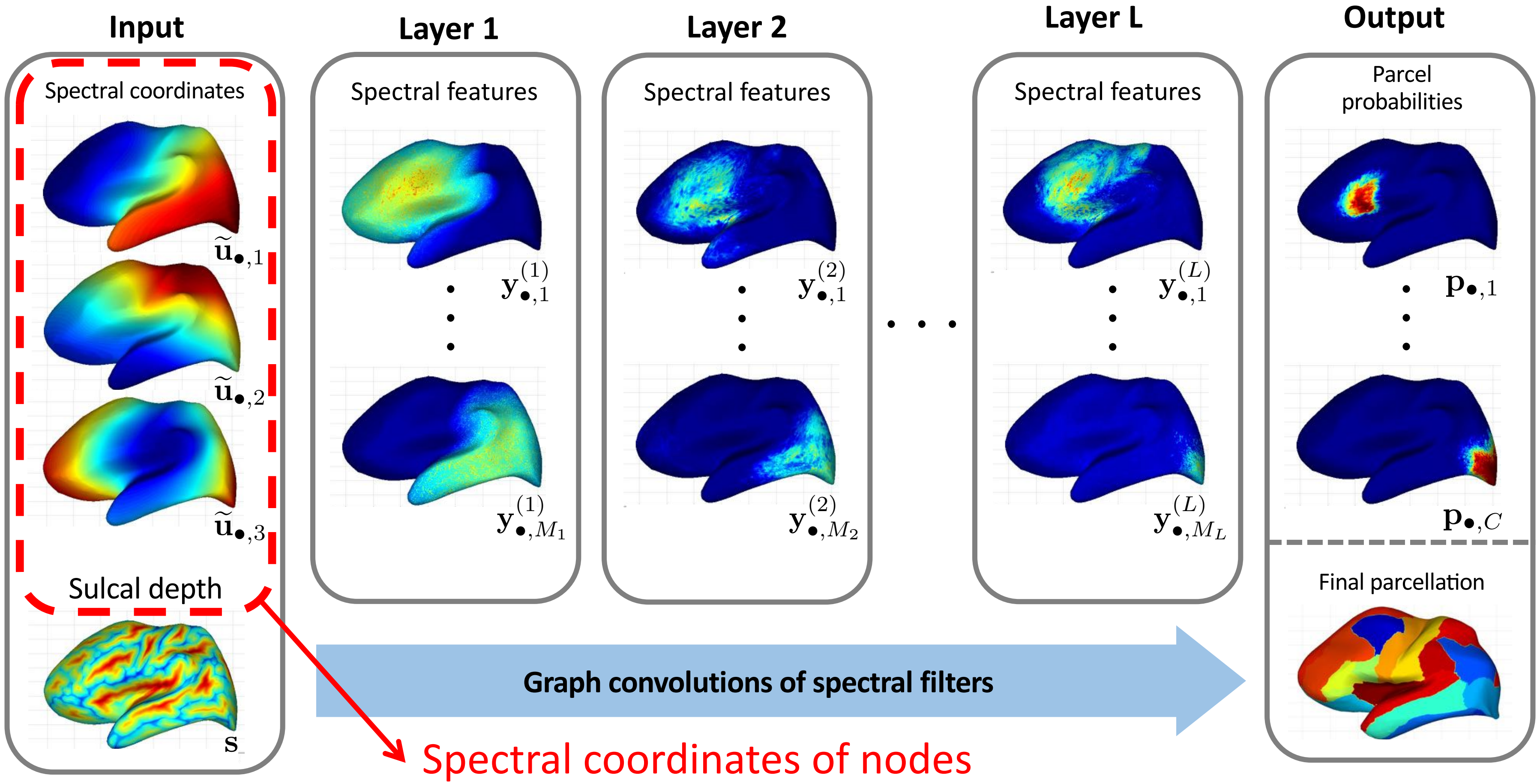
Karthik Gopinath, PhD Student – *Mostly his work*
Gopinath *et al*, 2018

Learning is now *Directly* on the Surface

Gopinath, Desrosiers, Lombaert, Medical Image Analysis 2018

Spectral Graph Conv Net – Feature Maps

- The Spectral Network – *illustrated*



Karthik Gopinath, PhD Student – *Mostly his work*

Gopinath *et al*, 2018

Gopinath, Desrosiers, Lombaert, Medical Image Analysis 2018

Spectral Graph Conv Net – Loss Function

Function **to optimize:**

Ground truth labels

Cross Entropy

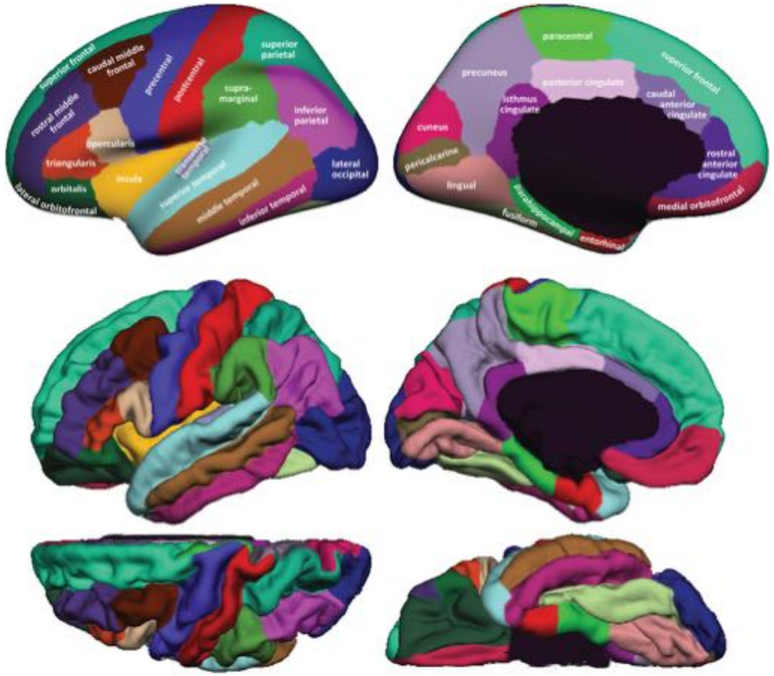
$$E(\mathbf{\Theta}) = -\sum_{i=1}^{N}\sum_{c=1}^{C} \boxed{s_{ic}} \cdot \log \boxed{p_{ic}(\mathbf{\Theta})}$$

Predicted probabilities

$$\mathbf{\Theta} = \{w_{pqk}^{(l)},\ b_p^{(l)},\ \mathbf{\Theta}_k^{(l)}\}$$

To Learn: Kernel weights, bias, parameters (μ,σ)

Experiments and Results



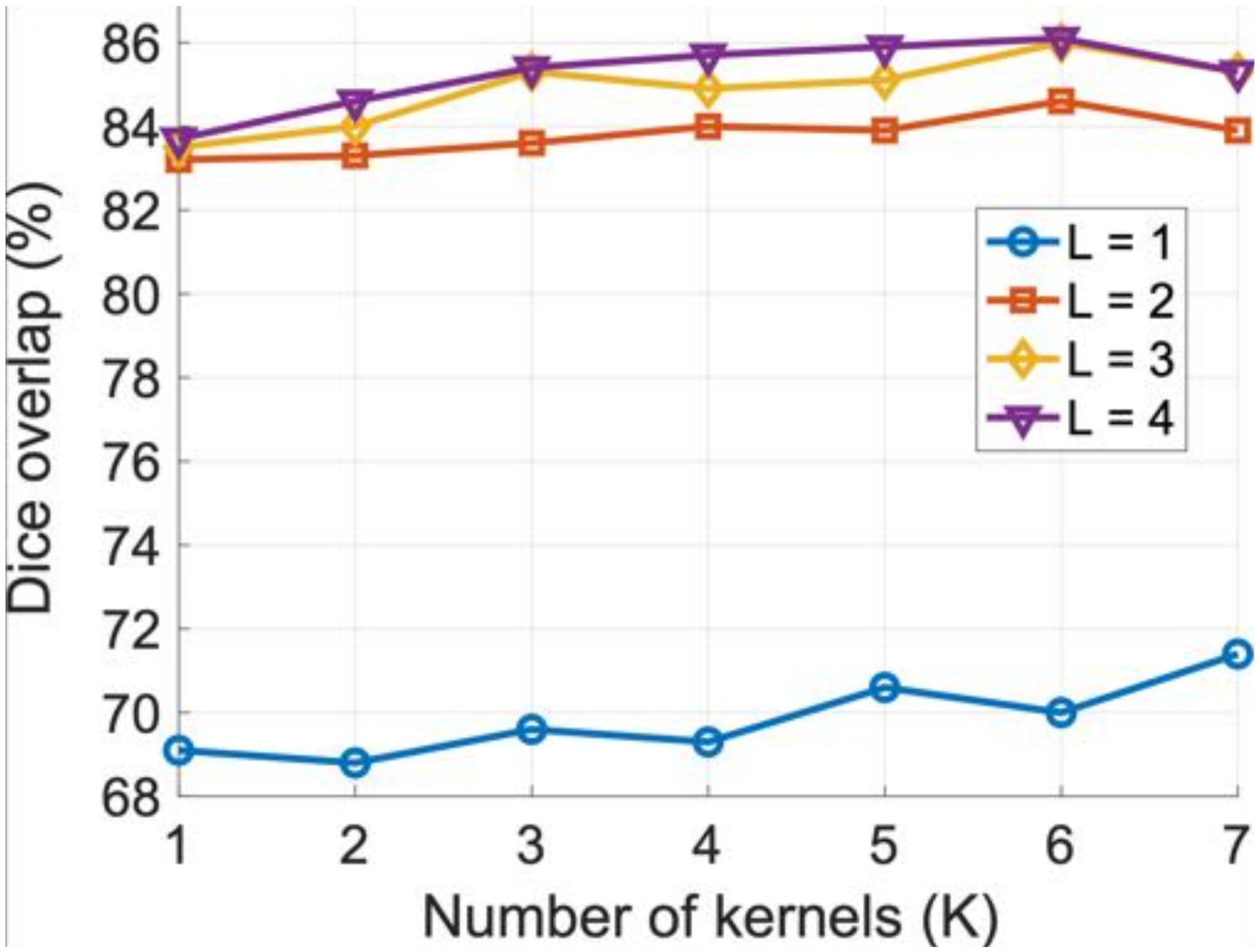
MindBoggle dataset :

- 101 subjects, seven different sites
- Meshes – from 102K to 185K vertices
- 32 manually labeled parcels

Klein *et al*, PLOS 2017

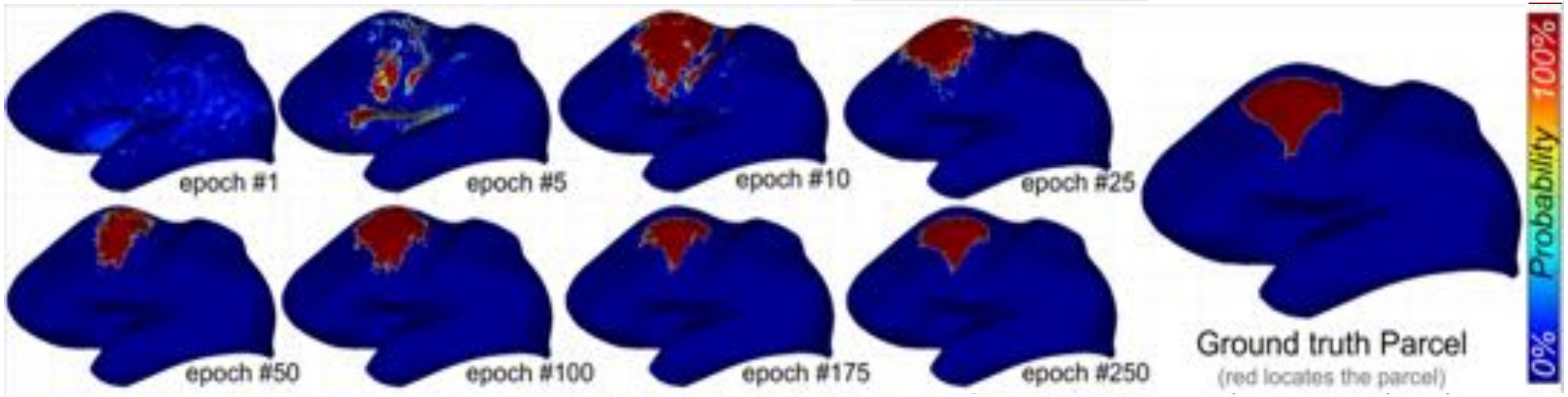
Spectral Graph Conv Net – Hyper-parameter Selection

We choose *L=4, K=6*



Spectral Graph Conv Net – Training Iterations

- **Training a feature map** – **Its evolution**
  - Towards resembling **observed cortical parcels**



Gopinath *et al*, MedIA 2018

Spectral Graph Conv Net – Results for Parcellation

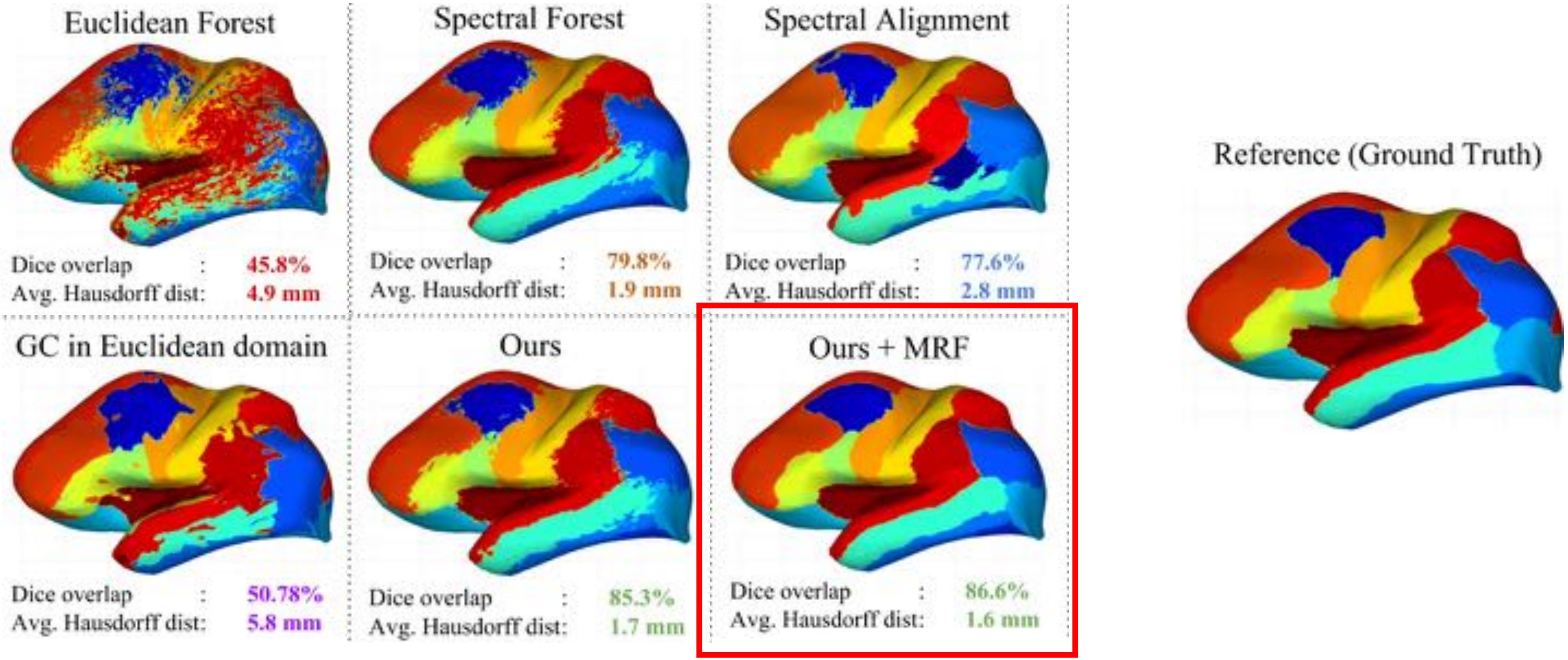
- Quantitative Results (86.6% vs FS: 84.4%)

Method	Dice overlap (%)	Accuracy (%)	Avg. Hausdorff (mm)
Euclidean forest	45.87 ± 8.74	49.26 ± 8.32	4.97 ± 1.11
GC on Euclidean	50.78 ± 10.78	54.24 ± 10.33	5.82 ± 1.66
Spectral alignment	77.67 ± 3.65	81.87 ± 3.39	2.87 ± 0.47
Spectral forest	79.89 ± 2.62	81.94 ± 2.54	1.97 ± 0.40
FreeSurfer	84.39 ± 1.91	85.19 ± 1.98	2.11 ± 0.29
Ours	85.37 ± 2.36	86.97 ± 2.43	1.75 ± 0.35
Ours + MRF	$\mathbf{86.61 \pm 2.45}$	$\mathbf{88.08 \pm 2.47}$	$\mathbf{1.66 \pm 0.44}$

Gopinath *et al*, MedIA 2018

Spectral Graph Conv Net – Results for Parcellation

- Qualitative Results (86.6% vs FS: 84.4%)



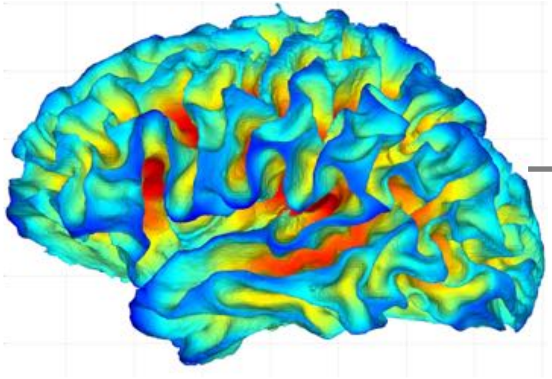
Advantage: Only **18 seconds per subject** VS hours for FreeSurfer

Gopinath *et al*, MedIA 2018

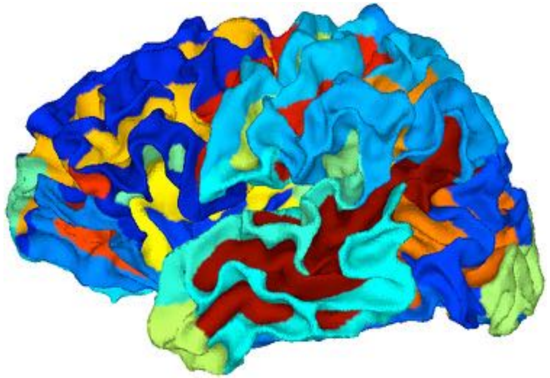
Contributions: Graph Conv

Graph Convolutions on Spectral Embeddings for Cortical Surface Parcellation

(1)



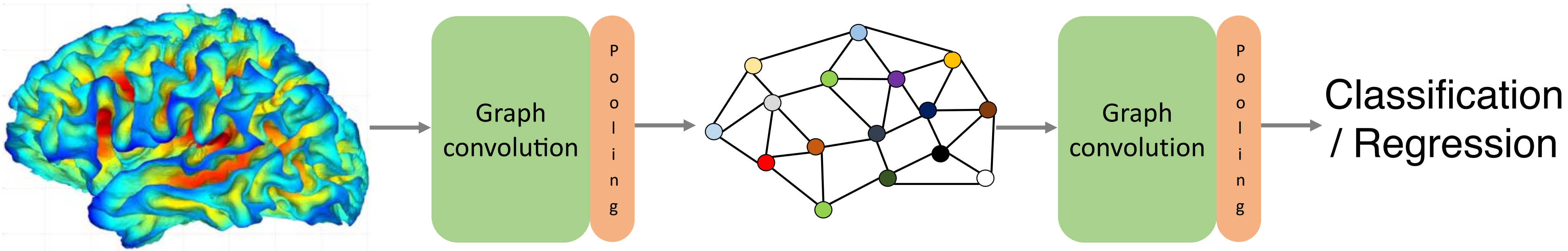
Graph convolution . . . . Graph convolution



[MedNeurips 2018, MedIA 2019]

One Contribution:
Learnable **Graph Pooling**

How to **Learn Graph Pooling** Patterns on Arbitrary Surfaces?

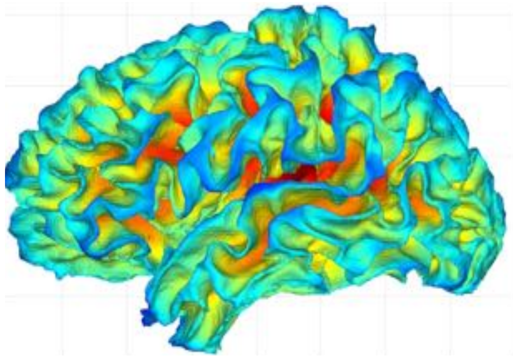


Gopinath *et al,* IPMI 2019, PAMI 2021

Related Work – Global Average Pooling

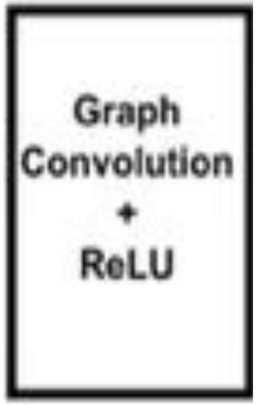
- Pool **from N nodes** to ***1* node**

How to Pool from $N^{(\text{layer 1})}$ to $N^{(\text{layer 2})}$ nodes?

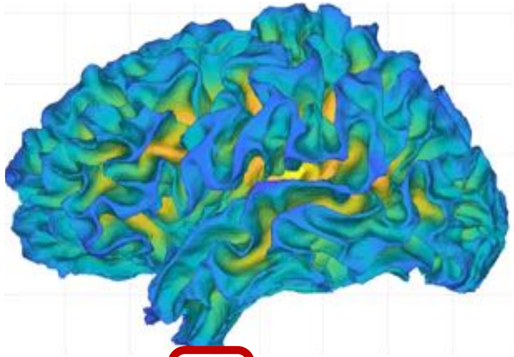


N x F_1
Nodes x # Feature maps

→



→



$\boxed{N}$ x F_2
Nodes x # Feature Maps

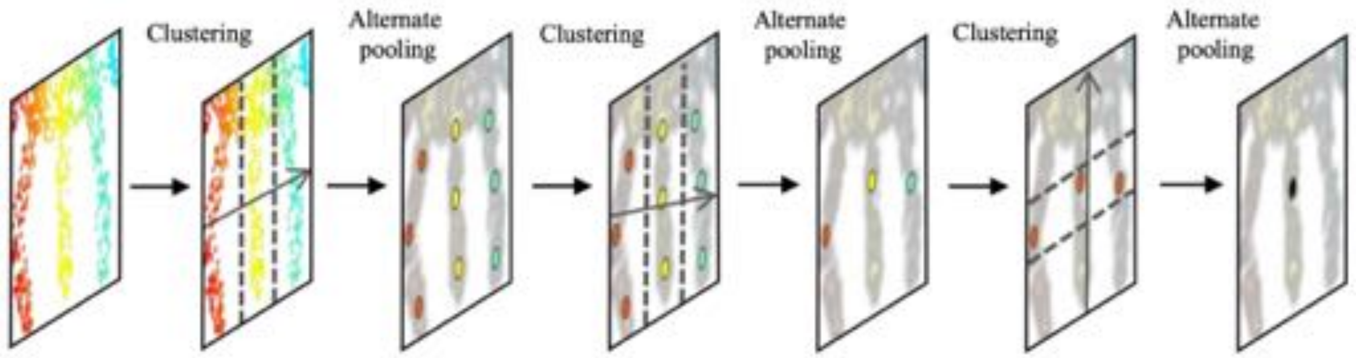
→ Global Average Pooling → Classification or Regression

$\boxed{1}$ x F_3
1 x # Feature Maps

Considers all nodes equally
(whole brain is 1 cluster)

Loss of shape information when pooling

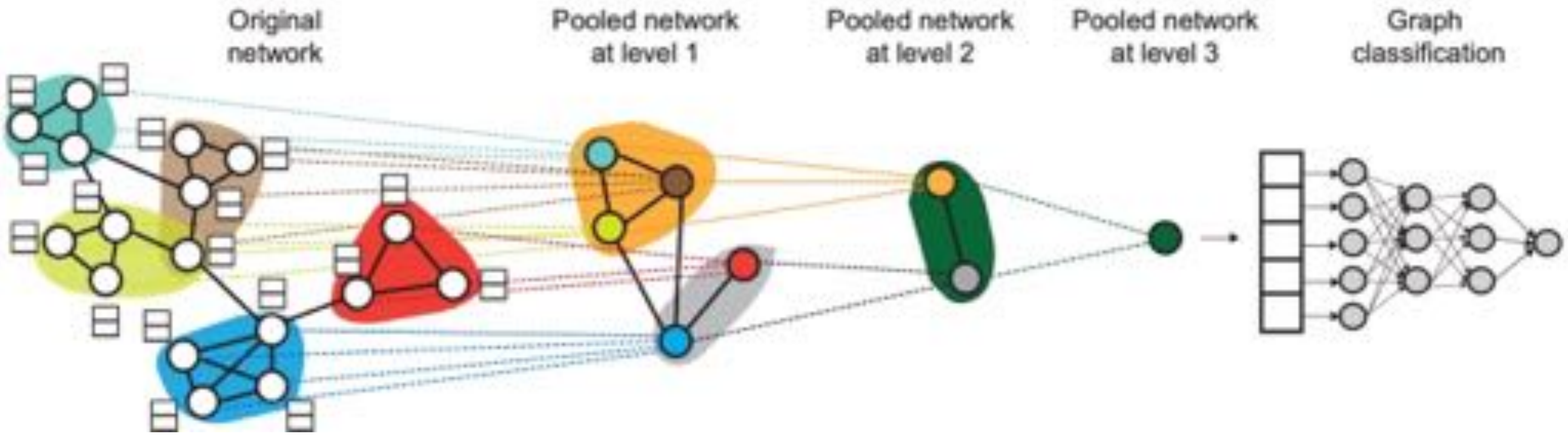
Related Work – Hierarchical Differentiable Pooling



Cluster with Spectral K-Means

Wang *et al*, ECCV 2018

Fixed number of cluster nodes



Nodes lacking intrinsic localization

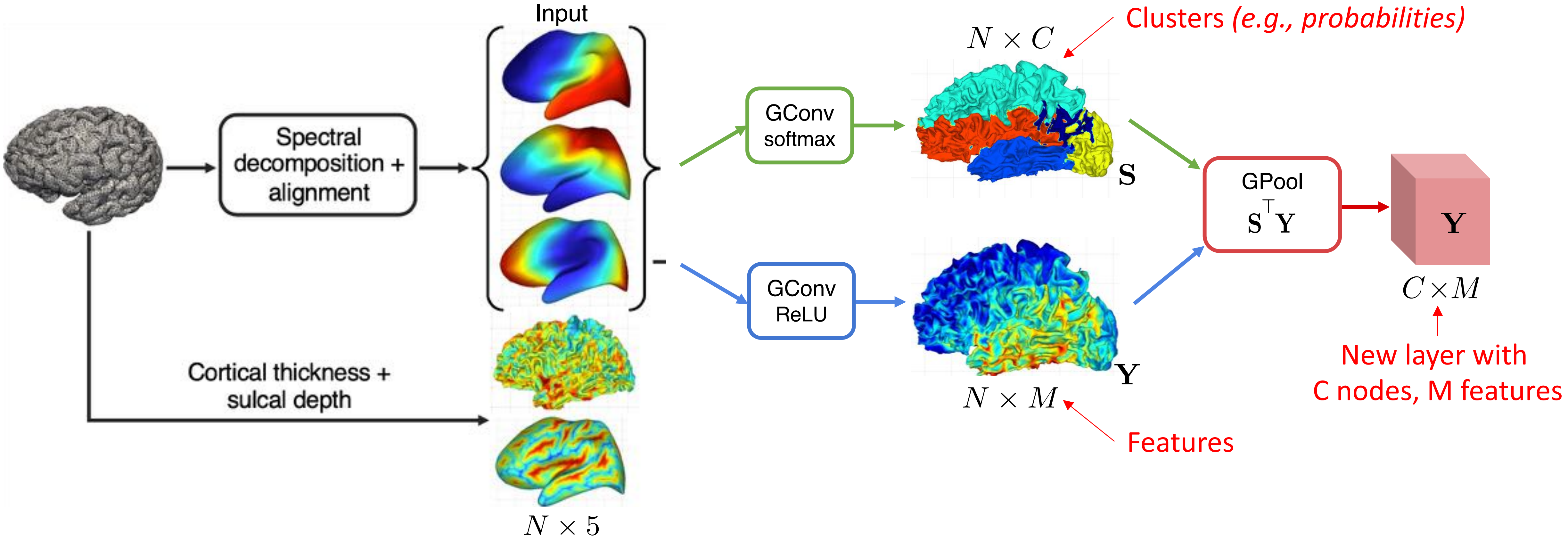
Learn node-cluster assignments

Ying *et al*, NeurIPS 2018

Wang *et al*, Pointset Learning, ECCV 2018
Ying *et al*, DiffPool, NeurIPS 2018

Proposed: Learnable Graph Pooling

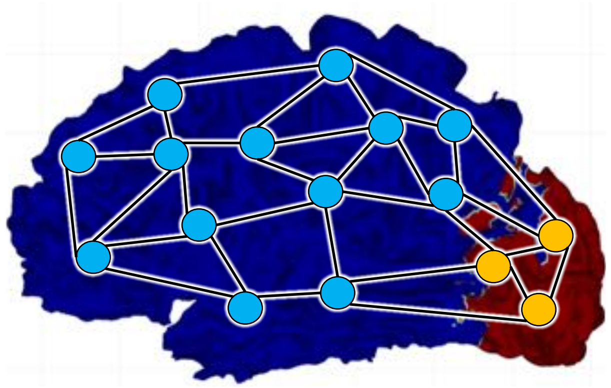
Uses two paths
1. Node to Cluster Assignment
2. Node features

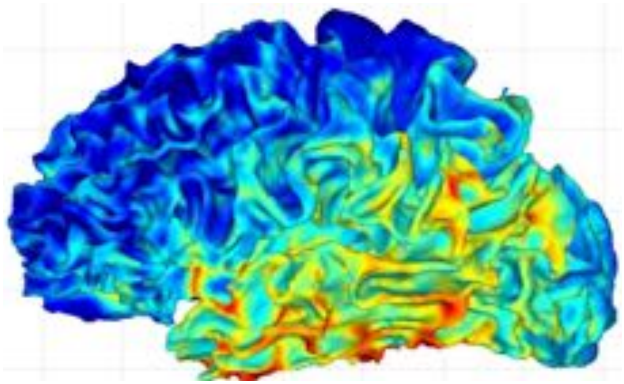


Learnable Graph Pooling – Building Nodes

Cluster Assignment $^{\top}$ · **Node features** =

From $N^{(\text{layer 1})}$ to $N^{(\text{layer 2})}$ nodes





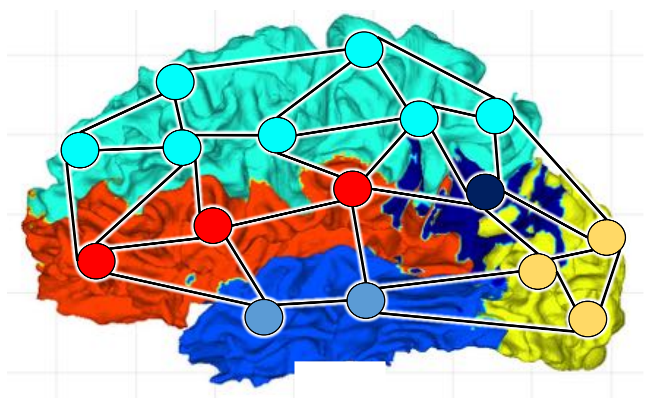
$\mathbf{S}^{\top}$ $\quad$ $\mathbf{Y}^{(l)}$ $\quad$ $\mathbf{Y}^{\text{pool}}$

$$y_{cp}^{\text{pool}} = \sum_{i=1}^{N} \boxed{s_{ic}} \cdot \boxed{y_{ip}^{(l)}}$$

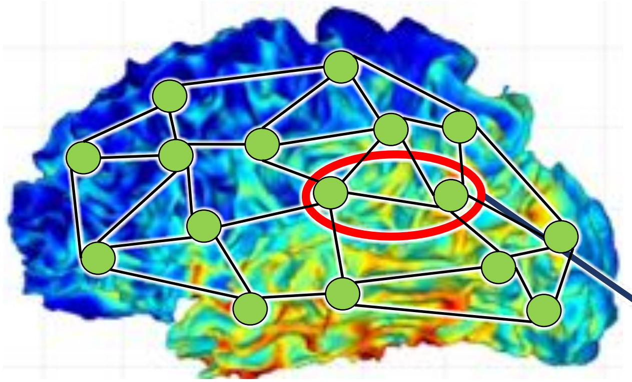
*Expected **convolution value** over a cluster*

Learnable Graph Pooling – Building Edges

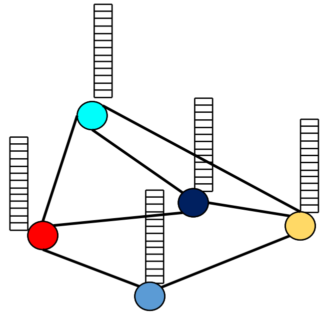
Cluster Assignment $^\top$ · **Node features** = **From $N^{(\text{layer 1})}$ to $N^{(\text{layer 2})}$ nodes**



$\mathbf{S}^\top$



$\mathbf{Y}^{(l)}$



$\mathbf{Y}^{\text{pool}}$

$$y_{cp}^{\text{pool}} = \sum_{i=1}^{N} s_{ic} \cdot y_{ip}^{(l)}$$

*Expected **convolution value** over a cluster*

$$a_{cd}^{\text{pool}} = \sum_{i=1}^{N} \sum_{j=1}^{N} s_{ic} \cdot s_{jd} \cdot \boxed{a_{ij}}$$

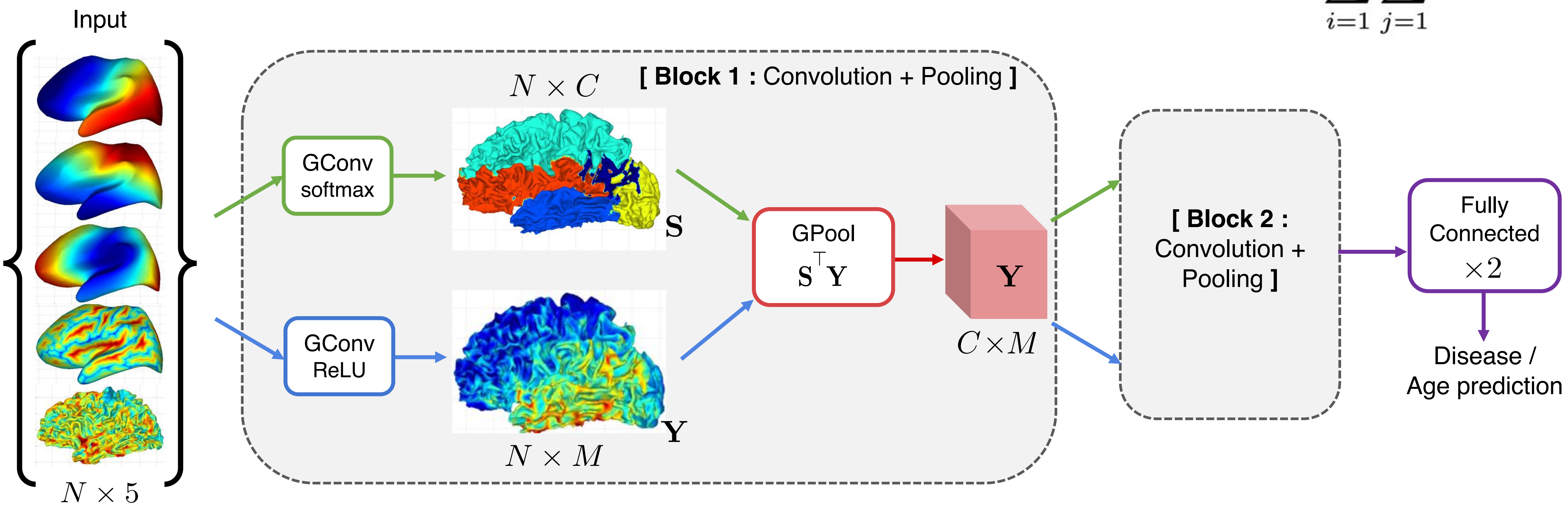
*Expected **edge weight** between clusters (c,d)*

Learnable Graph Pooling – Multiple Layers

- Adding [Conv+Pool] Blocks

New Node Value: $$y_{cp}^{\text{pool}} = \sum_{i=1}^{N} s_{ic} \cdot y_{ip}^{(l)},$$

New Edge Weight: $$a_{cd}^{\text{pool}} = \sum_{i=1}^{N} \sum_{j=1}^{N} s_{ic} \cdot s_{jd} \cdot a_{ij}$$



Learnable Graph Pooling – Loss Function

Function **to optimize**:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\text{out}}(\boldsymbol{\theta}) + \alpha \mathcal{L}_{\text{reg}}\big(\mathbf{S}(\boldsymbol{\theta})\big)$$

Cross-Entropy /
Mean square error

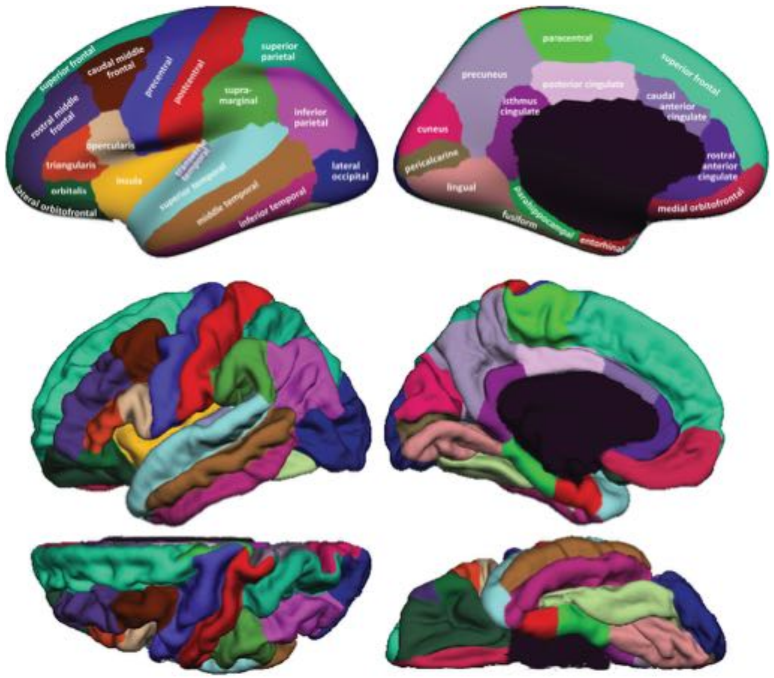
Regularization loss
to obtain spatially regular clusters

$$\mathcal{L}_{\text{reg}}(\mathbf{S}) \;=\; \sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij} \cdot \|\mathbf{s}_i - \mathbf{s}_j\|^2 \;=\; \operatorname{tr}(\mathbf{S}\mathbf{L}\mathbf{S}^\top)$$

Avoids issues of [Ying *et al,* 2018]:

- **Hard training** of pooling path,
- Spurious **local minima**

Experiments and Results



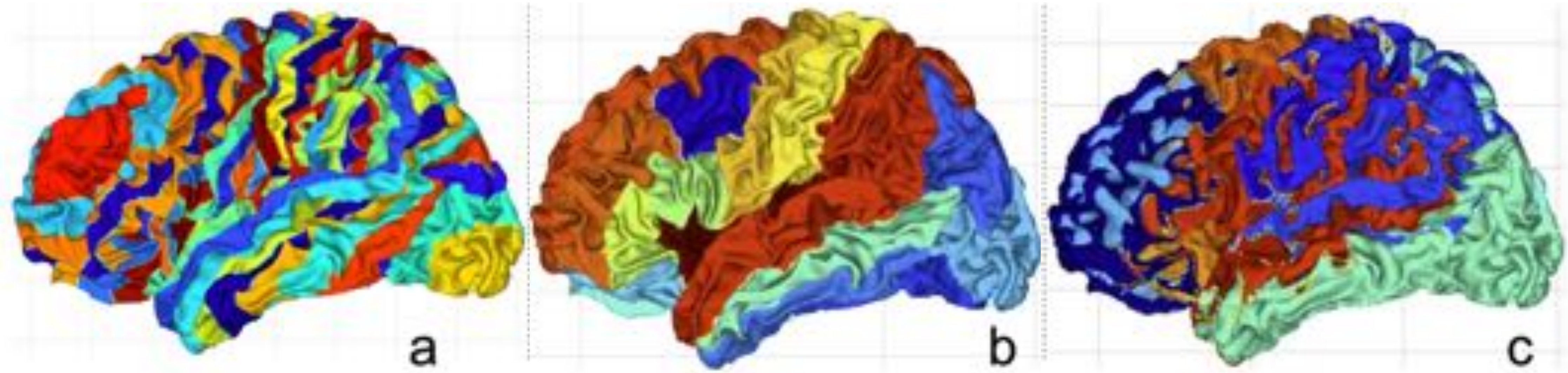
<u>Datasets:</u>
- ADNI: 731 brains
- MindBoggle: 101 brains

<u>Experiments:</u>
- Pooling comparison
- Disease classification
- Age prediction

Klein *et al*, PLOS 2017
Jack *et al*, MRI 2008

Comparison of Different Pooling Methods

- **Pooled Clusters** from Subject-sex Classification



Spectral **k-means** clustering | Fixed **parcel** clusters | **Learned clusters** from our method

Gopinath *et al*, IPMI 2019, PAMI 2021
Wang *et al*, ECCV 2018

Comparison of Different Pooling Methods

- **Pooled Clusters** from Subject-sex Classification

Pooling method	Mean ± Std.
Global Average Pooling	60.76 ± 3.62
Fixed Parcellation Pooling	64.59 ± 7.84
Spectral Clustering Pooling	67.94 ± 4.97
Top-k pooling	78.94 ± 3.32
Learnable Pooling (ours)	**84.21 ± 3.72**

Geometry-based **Pooling improves** Sex **Classification**

Gopinath *et al*, IPMI 2019, PAMI 2021
Wang *et al*, ECCV 2018
Gao, Ji, ICML 2019

Learnable Pooling – Results for Disease Classification

- **Dataset**: 731 FreeSurfer Brain Surfaces from ADNI

Learnable Pooling

↓

Classification

↓

Normal *vs* MCI *vs* Alzheimer's

Average accuracy for disease classification

	Baseline*	**Ours without spectral features**	**Ours with spectral features**
Features	Cortical thickness + Sulcal Depth	Cortical thickness + Sulcal Depth	**Spectral +** Cortical thickness + Sulcal Depth
NC vs MCI	63 ± 4	63.71 ± 5.72	**70.79 ± 6.40**
MCI vs AD	65 ± 6	74.03 ± 8.63	**76.92 ± 4.78**
NC vs AD	80 ± 5	76.00 ± 6.06	**89.33 ± 4.30**

*C. Ledig *et al*, 2014
Pointwise information,
No neighborhood

Learnable Graph Pooling,
No geometrical information

Learnable Graph Pooling,
With geometrical information

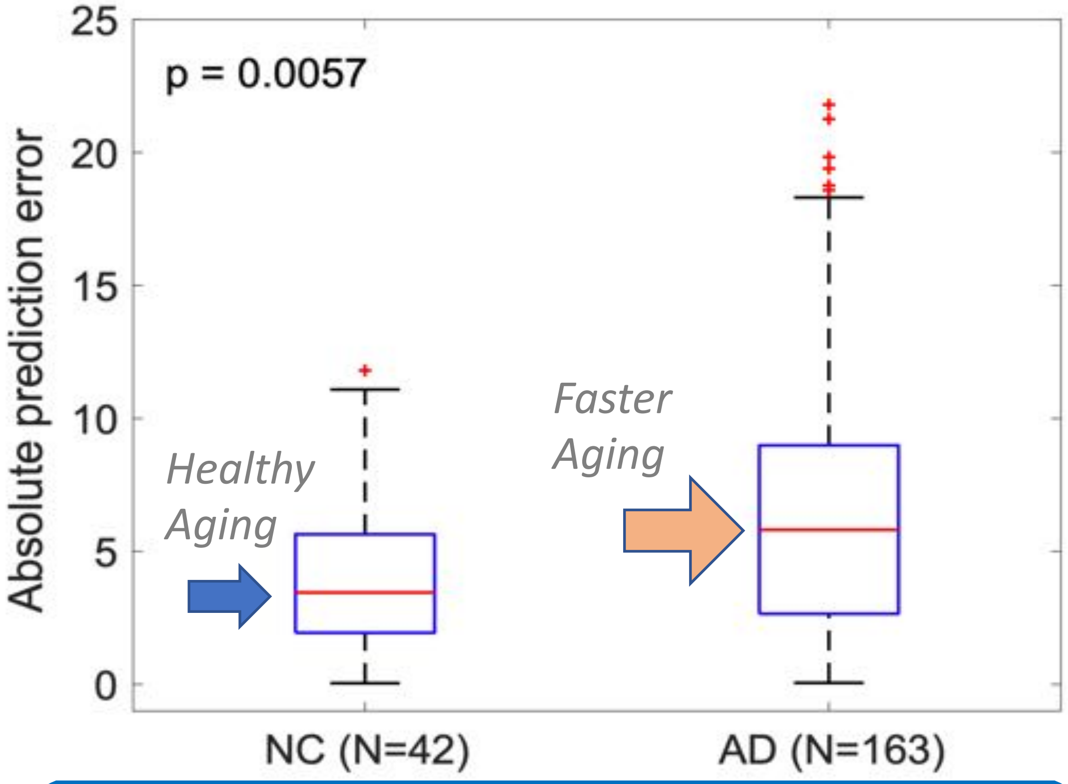
Geometry-based **Pooling**
improves Alzheimer's **Classification**

Learnable Pooling – Results for Brain Age Prediction

- **Assumption:** Can our model be used as a biomarker for AD?
- **Prediction** of Alzheimer's age (or **Geometry age**) **differs** from Healthy

Learnable pooling → **Age Regression** → Train on NC → Predict on NC and AD

Train on Healthy (NC)
Predict Age *from Geometry*
↪ if age is good -> **Healthy**
↪ if **faster aging** -> **Alzheimer's**

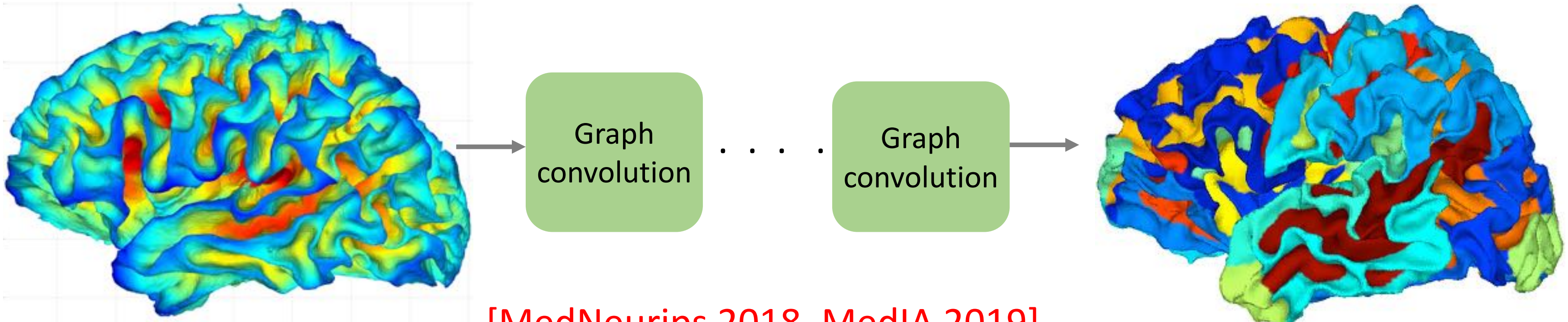


Geometry-age of Alzheimer's Subjects
Deviates from Normal Aging

Contributions: Graph Conv + Pooling

Graph Convolutions on Spectral Embeddings for Cortical Surface Parcellation

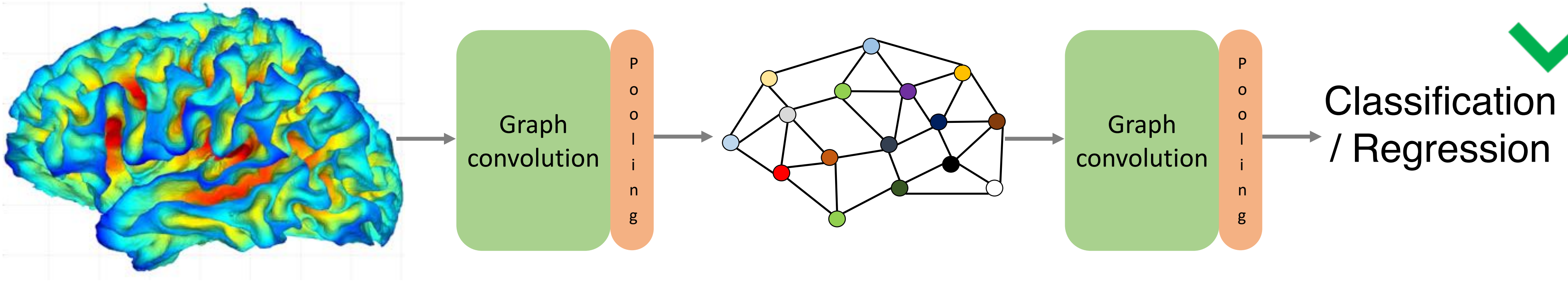
(1)



[MedNeurips 2018, MedIA 2019]

Learnable Pooling in Graph Convolutional Networks for Brain Surface Analysis

(2)



[IPMI 2019, TPAMI 2021]

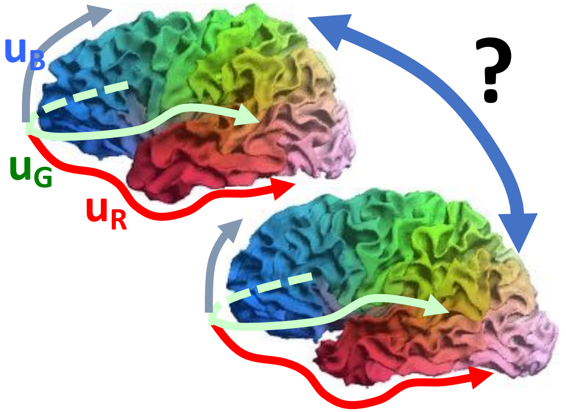
1 2 3 4 5 6 **7**

Conclusion: Rethinking Learning on Surfaces

Use Spectral Shape Embeddings

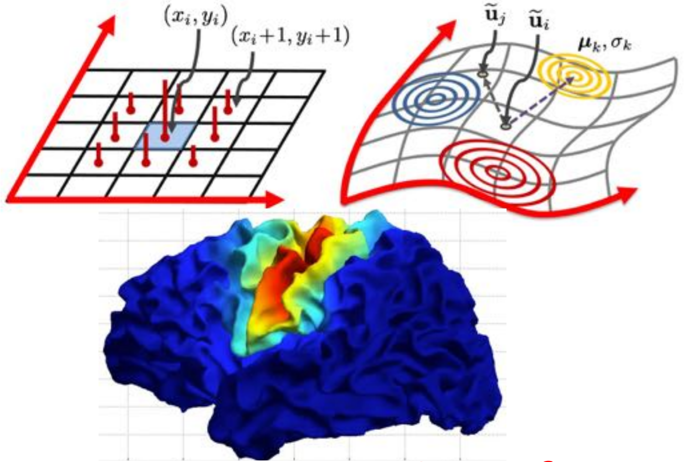
Conclusions

1 *Spectral* Parameterization



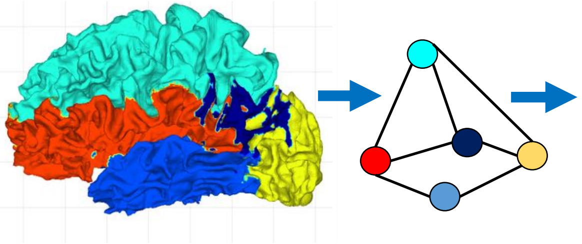
***Intrinsic* Shape Coordinates**

2 *Spectral Graph* Convolution



Conv Nets on Brain Surfaces

3 *Spectral Graph* Pooling



Classification / Regression

Geometry-based Pooling

Dr. Karthik Gopinath

Prof. Christian Desrosiers

Take Home Message

- **Graph Convolution + Pooling on Surfaces,**
  - **↪ Easier with Spectral Shapes**
    - **Simple – Fast** Operations on Surfaces
    - **Direct** Learning on Surface
- Limitations: Same Mesh Topology

PAPERS

[PAMI 2021] Learnable **Pooling** in Graphs
[IPMI 2019] Graph Convolution **Pooling**
[MedIA 2018] **Graph Convolutions**

Canada Research Chairs — Chaires de recherche du Canada
NSERC CRSNG
Fonds de recherche sur la nature et les technologies Québec
McGill
Inria informatiques mathématiques
Microsoft Research

Acknowledgments