Geometric Deep Learning in Medical Imaging

Prof. Hervé Lombaert, ETS Montreal

Summer School on Deep Learning for Medical Imaging 2021
Geometry & Machine Learning

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

Correspondence?

Parcellation?

Classification?

Cerebral Cortex

Key role in cognition, planning & perception
Segmentation on Medical Images

• One Example – Finding Lesions on Brain MRIs

Kamnitsas et al, DeepMedic, MedIA 2017
Segmentation on Medical Images

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

  ![Diagram of Convolutional Neural Network](image)

  **A Conv Filter Scans the Image**

  **Algorithm Learns the Filter Params**

  **In High Resolution**

  **In Low Resolution**

  ![Image courtesy: Ben Glocker, DeepMedic]

**Many works on CNN Segmentations**

- Zikic et al, 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

**One issue**

- Kamnitsas et al, DeepMedic, MedIA 2017
- Havaei et al, 2015
- Pereira et al, 2015
- Menze et al, 2015
- Prasson et al, 2013
- Li et al, 2014
- Roth et al, 2014
- Brebisson and Montana, 2015
- Ciresan et al, 2013
- Ronneberger et al, 2015
- Zhao et al, 2018
- Litjens et al, 2017 – and more
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?**

[Image courtesy: Freesurfer]
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 ($\mu$'s and $\sigma$'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)
Qiu, Bitouk, Miller, TMI (2006)
Shi, Lai, Wang, Pelletier, Mohr, Sicotte, Toga, TMI (2014)
Germanaud, Lefevre, Toro, Fischer, Dubois, Hertz, Mangin, Neuroimage (2012)
Challenge – Anatomical Variability

Complex Shapes, Highly variable

How to find point correspondence?
Challenge – Anatomical Variability

One Related Problem –

Matching Points between Brains

Flowing Surfaces
• Costly (CPU, mesh size)

Sphere Inflations
• Costly (3 to 4 hours)

Spectral Matching
✓ Fast (Few seconds)
✓ Accurate (as FreeSurfer)

Proposal → Fast, as Accurate

Dense Point Correspondence
300k+ meshes

(LDDMM and variants)

\[ E(v) = \int_0^t |v_t|^2 dt + \int_{\Omega} \| \phi_t^{-1}(y) - f(y) \|^2 dy \]

(FreeSurfer, Spherical Demons)

\[ \phi(t) = c_t \circ \exp(v(t)) \text{ on } S^2 \]

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)
Background on Spectral Shape Analysis
How to Represent and Exploit Surfaces?
Spectral Signature

Shape Vibration $\rightarrow$ Unique intrinsic Shape Signature

Spectral Decomposition

Spectral Signature

---

Eigenmode 1  |  Eigenmode 2  |  Eigenmode 3  |  Eigenmode 4  |  Eigenmode 5

Good: Equivalent Points $\rightarrow$ Same (shape) Spectral Coordinates

- Umeyama, PAMI (1988)
- Shapiro & Brady, IVC (1992)
- Mateus, CVPR (2008)
- Jain & Zhang, ICSMA (2006)
- Reuter et al., MICCAI (2007), CAD (2009)
- Ovsjanikov et al., SIGGRAPH (2012)
- Shi, Dinov, Toga, TMI (2014)
Method – Spectral Shapes

Surfaces in Euclidean Space

\[ L = U^T \Lambda U \]

\( u^{(1)} \)
\( u^{(2)} \)
\( u^{(3)} \)

Graph Laplacian

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

Surface \#1

Surface \#2

Surfaces in Spectral Space

Energy:

\[ \phi(v_i) = \arg\min_{\phi(v_i) \in B} \| D_A(v_i) - D_B(\phi(v_i)) \|^2 \]

Data Term

Spatial Term

Nearest Neighbor Search between vectors \([D_A(v_i)] \) & \([D_B(\phi(v_i))]\)

Add Data

Sulcal depth

Easier Matching

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**

**Surfaces in Euclidean Space**

- **Before (FreeSurfer):** CPU (3-4hrs)
- **After (Spectral Matching):** < 1 min

**Surfaces in Spectral Space**

- **Easier Matching**

---

**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

Spherical Demons

- ✔ Average Dice = 0.85 (±0.07)
- × 2hrs + 3mins for one matching

Spectral Matching

- Average Dice = 0.83 (±0.08)
- <1min for one matching

References:

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 ($\mu$’s and $\sigma$’s)
- Requires Graph Neighborhoods

Intrinsic Shape Parameterization
Intrinsic Surface Parameterization

• Spectral Coordinates
  • an Intrinsic Surface Parameterization

Spectral Coordinates
Equivalent Points → Similar Shape Characteristics

Same Spectral Coordinates (Same RGB)
Approach: Learning on Surfaces

Input

| vertex #1 | +0.2cm |
| vertex #2 | -0.7cm |
| ...       |        |
| vertex #N | 1.4cm  |

Output

How to Represent Space?

Test1: >0cm ?
Yes - No
Test2: frontal?
Yes - No
Test3: ...

Final Answer:
# calcarine

e.g., Random Forests

Classifier

Data + Ground Truth

Criminisi, Shotton, Springer (2013)
Amit, Geman, Neural Comp. (1997), Breiman, ML (2001)
Learning on **Surfaces**

**Standard Euclidean Forests**
- Based on Euclidean Coordinates
  - \( v = \{\text{Data}, X, Y, Z\} \)

**Spectral Forests**
- Spectral Coordinates is Geometry Aware
  - \( v = \{\text{Data}, U_1, U_2, U_3\} \)

**Reason**: Ignore Complex Geometry

**Similarity**: 54.5%

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

**Similarity**: 92.1%

**Lombaert, Criminisi, Ayache, MICCAI (2015)**
**Application: Learning on Surfaces**

![Diagram](image)

**Standard Forests**
Spatial Representation is extrinsic

**Spectral Forests**
Learning is *Directly* on the surface

---

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

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

---

**Random Forests**

---

**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)*
Background on Geometric Deep Learning

How to Learn on Graph Node Data?
Neural Network on Images

Problem if image content moves
X No invariance to translation

$\sigma\left(\sum_{j=1}^{N} h^{(l)}_{j}(x) \cdot w_{ij}\right)$
Convolutions on Images

One Solution: Let’s move along the image

√ Invariance to translation

Convolution

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 $\leftrightarrow$ Multiplication in Fourier space

$$\mathcal{F}\{f \ast g\} = \mathcal{F}\{f\} \cdot \mathcal{F}\{g\}$$
Spectral Convolutions on Graphs

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

\[ f * g = \mathcal{F}^{-1}\{\mathcal{F}\{f\} \cdot \mathcal{F}\{g\}\}\]

\[ = \Phi (\Phi^T g) \odot (\Phi^T f) \]

\[ = \Phi \text{diag}(\mathcal{F}\{g\}) \Phi^T f \]

\[ \text{diag}(\mathcal{F}\{g\}) \text{ expressed in terms of } \lambda \]

\[ = \Phi \text{diag}(\mathcal{F}\{g(\lambda)\}) \Phi^T f \]

\[ \mathcal{F}\{g(\lambda)\} \text{ approximated with Chebyshev Polynomials:} \]

\[ \approx \Phi \text{diag} \left( \sum_{k=0}^{K} \theta_k T_k(\lambda) \right) \Phi^T f \]

In Fourier Space, matrix notation

\[ \mathcal{F}\{g(\lambda)\} = \sum_{k=0}^{K} \theta_k T_k(\lambda) \]

simplification with First-order Chebyshev

\[ \approx \theta_0 f - \theta_1 D^{-\frac{1}{2}} AD^{-\frac{1}{2}} f \]

simplification with single parameter

\[ \theta = \theta_0 = -\theta_1 \]

\[ \approx \theta \left( I + D^{-\frac{1}{2}} AD^{-\frac{1}{2}} \right) f \]

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

\[
\hat{f} \ast \hat{g} = \sum_{k=0}^{K} \theta_k T_k(L) \hat{f}
\]

\[
\approx \theta_0 \hat{f} - \theta_1 D^{-1/2} A D^{-1/2} \hat{f} \\
\approx \theta \left( I + D^{-1/2} A D^{-1/2} \right) \hat{f}
\]

**Easy Convolution on Graphs**

**Simple Convolution Layer**

\[
h^{(l+1)}_i(x) = \sigma \left( \sum_{j=1}^{N} \left( h^{(l)}_j \ast w_{ij} \right)(x) \right)
\]

\[
h^{(l+1)}_i(x) = \sigma \left( \theta h^{(l)}_i(x) + \theta \frac{1}{\sqrt{d_i}} \sum_{j \in N_i} a_{ij} \frac{1}{\sqrt{d_j}} h^{(l)}_j(x) \right)
\]

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**

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

[Image Courtesy: Kipf, GCN, 2016]

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?

Learn Filter Weights in Polar Coordinates

Patch Operator $D(x)f$

Image Courtesy: Monti, MoNet, 2017

Image Courtesy: Masci, Geodesic CNN, 2015

Kokkinos et al, CVPR 2012
Masci et al, 3dRR 2015
Boscaïni 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**

Learn Kernel Parameters

\[ h_i^{(l+1)}(x) = \sigma \left( \sum_{j=1}^{N_i} h_j^{(l)} \ast w_{ij}(x) \right) \]

Patch Orientation? Patch Construction?

Richer Kernels on Shapes

Patch Operator

\[ D(x)f \]

Learn Kernel Parameters

\[ h_i^{(l+1)}(x) = \sigma \left( \sum_{j=1}^{N_i} h_j^{(l)} \ast w_{ij}(x) \right) \]

Richer Kernels on Shapes

**Geodesic CNN**

[Bosciaini et al, NeurIPS 2016]

\[ w(u) = \exp \left( -\frac{1}{2} (u - \mu_{\rho,\theta})^T \left( \sigma_{\rho}^{-1} \sigma_{\theta}^{-1} \right) (u - \mu_{\rho,\theta}) \right) \]

**Anisotropic GCNN**

[Bosciaini et al, NeurIPS 2016]

\[ w(u) = \exp \left( -tu^T R \alpha \left( \begin{array}{c} 1 \\ R^T u \end{array} \right) \right) \]

**Mixture of Gaussians GCNN**

[Monti et al, CVPR 2017]

\[ w(u) = \exp \left( -\frac{1}{2} (u - \mu_{\rho,\theta})^T \Sigma_{\rho,\theta}^{-1} (u - \mu_{\rho,\theta}) \right) \]

[Image Courtesy: Monti, MoNet, 2017]
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
Limitations of Geometric Deep Learning

What is preventing Generalization to Arbitrary Surfaces?
Challenges in Medical Imaging

• **Geometrical Complexity** of Surfaces

[Image courtesy: Freesurfer]

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? \((x, y, z); (\rho, \theta)\) inadequate in Euclidean Space

Mesh Coordinates **Inadequate in Euclidean Space**

*Image courtesy: Freesurfer*
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

Conv Filters on Surfaces

Algorithm:
– Learns the Filter params ($\mu$'s and $\sigma$'s)
– Requires Graph Neighborhoods

The Graph Structure can Change

How to Parameterize a Brain Surface?
Challenges in Medical Imaging

Convolution

Pooling
Graph Networks – Two Contributions

Graph Convolutions on Spectral Embeddings for Cortical Surface Parcellation

(1) Graph convolution . . . Graph convolution


Learnable Pooling in Graph Convolutional Networks for Brain Surface Analysis

(2) Graph convolution Pooling Graph convolution Pooling

Classification / Regression

[IPMI 2019, PAMI 2021]
One Contribution: Localized Graph Convolutions

How to Navigate Graph Convolutions on Arbitrary Surfaces?

Convolutions on Surfaces

• Convolutions on Spectral Embeddings

Convolution on an Image

Graph Convolution on a Brain Surface

Euclidean $\rightarrow$ Spectral Coordinates
Spatial Information as Spectral Encoding

Graph Laplacian

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

Spectral Decomposition

\[
L = U^T \Lambda U
\]

Problem: Spectral bases are ambiguous to rotation

Lombaert et al, IPMI 2015
Spectral Alignment

\[ L = U^T \Lambda U \]

\[ \tilde{U} = T^{1\rightarrow ref} U \]

\[ \tilde{U} = T^{2\rightarrow ref} U \]

\[ \tilde{U} = T^{N\rightarrow ref} U \]

Hervé Lombaert, Summer School on Deep Learning for Medical Imaging, 2021
Extension of 2D convolutions to irregular grids

Standard convolution on regular grid:

\[
(x_i, y_i) \rightarrow (x_i+1, y_i+1)
\]

Grid space

Geometric convolution for embedded graphs:

\[
l^{(l+1)} = f(z^{(l)})
\]

Non-linear activation (ReLU)

Convolutions kernel weights

Input feature map

Neighbor nodes on mesh

Parameters are learned

Extension of 2D convolutions to irregular grids

Standard convolution on regular grid:

\[
(x_i, y_i) \rightarrow (x_i+1, y_i+1)
\]

Grid space

Geometric convolution for embedded graphs:

\[
l^{(l+1)} = f(z^{(l)})
\]

Non-linear activation (ReLU)

Convolutions kernel weights

Input feature map

Neighbor nodes on mesh

Parameters are learned

\[
\varphi(\hat{u_i}, \hat{u_j}; \mu_k, \sigma_k) = \exp \left( -\sigma_k \parallel (\hat{u_j} - \hat{u_i}) - \mu_k \parallel^2 \right)
\]
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

**Learning is now Directly on the Surface**

Karthik Gopinath, PhD Student – Mostly his work

Gopinath et al, 2018
Spectral Graph Conv Net – Feature Maps

• The Spectral Network – *illustrated*

Input

- Spectral coordinates
- Sulcal depth

Layer 1

- Spectral features
- $y_1^{(1)}$
- $y_2^{(1)}$
- $y_3^{(1)}$

Layer 2

- Spectral features
- $y_1^{(2)}$
- $y_2^{(2)}$
- $y_3^{(2)}$

Layer $L$

- Spectral features
- $y_1^{(L)}$
- $y_2^{(L)}$
- $y_3^{(L)}$

Output

- Parcel probabilities
- $P_1$
- $P_2$
- $P_C$

Final parcellation

Graph convolutions of spectral filters

Spectral coordinates of nodes

Karthik Gopinath, PhD Student – *Mostly his work*

Gopinath et al, 2018

Gopinath, Desrosiers, Lombaert, Medical Image Analysis 2018
Spectral Graph Conv Net – Loss Function

Cross Entropy

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

\( \Theta = \{ w_{pqk}^{(l)}, b_p^{(l)}, \Theta_k^{(l)} \} \)

To Learn: Kernel weights, bias, parameters \((\mu,\sigma)\)
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%)

<table>
<thead>
<tr>
<th>Method</th>
<th>Dice overlap (%)</th>
<th>Accuracy (%)</th>
<th>Avg. Hausdorff (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euclidean forest</td>
<td>45.87 ± 8.74</td>
<td>49.26 ± 8.32</td>
<td>4.97 ± 1.11</td>
</tr>
<tr>
<td>GC on Euclidean</td>
<td>50.78 ± 10.78</td>
<td>54.24 ± 10.33</td>
<td>5.82 ± 1.66</td>
</tr>
<tr>
<td>Spectral alignment</td>
<td>77.67 ± 3.65</td>
<td>81.87 ± 3.39</td>
<td>2.87 ± 0.47</td>
</tr>
<tr>
<td>Spectral forest</td>
<td>79.89 ± 2.62</td>
<td>81.94 ± 2.54</td>
<td>1.97 ± 0.40</td>
</tr>
<tr>
<td>FreeSurfer</td>
<td>84.39 ± 1.91</td>
<td>85.19 ± 1.98</td>
<td>2.11 ± 0.29</td>
</tr>
<tr>
<td>Ours</td>
<td>85.37 ± 2.36</td>
<td>86.97 ± 2.43</td>
<td>1.75 ± 0.35</td>
</tr>
<tr>
<td>Ours + MRF</td>
<td><strong>86.61 ± 2.45</strong></td>
<td><strong>88.08 ± 2.47</strong></td>
<td><strong>1.66 ± 0.44</strong></td>
</tr>
</tbody>
</table>

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
Contributions: Graph Conv

Graph Convolutions on Spectral Embeddings for Cortical Surface Parcellation

(1)

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

Classification or Regression

Loss of shape information when pooling

Considers all nodes equally

$N \times F_1$
# Nodes x # Feature maps

$N \times F_2$
# Nodes x # Feature Maps

$1 \times F_3$
1 x # Feature Maps

$N \times F_3$
# Nodes x # Feature Maps

$N \times F_3$
# Nodes x # Feature Maps
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**

Clusters (e.g., probabilities)

New layer with C nodes, M features

Features

*Hervé Lombaert, Summer School on Deep Learning for Medical Imaging,* 56
Learnable Graph Pooling – Building Nodes

Cluster Assignment \[ S^T \] \[ \rightarrow \] Node features

\[ y_{cp}^{pool} = \sum_{i=1}^{N} s_{ic} y_{ip}^{(l)} \]

Expected convolution value over a cluster
Cluster Assignment \( T \) \( S^T \)

Node features

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

Expected convolution value over a cluster

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

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

\[ \mathcal{L}(\theta) = \mathcal{L}_{\text{out}}(\theta) + \alpha \mathcal{L}_{\text{reg}}(S(\theta)) \]

- **Cross-Entropy / Mean square error**
- **Regularization loss** to obtain spatially regular clusters

\[ \mathcal{L}_{\text{reg}}(S) = \sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij} \cdot \|s_i - s_j\|^2 = \text{tr}(SLS^T) \]

Avoids issues of [Ying et al, 2018]:
- **Hard training** of pooling path,
- Spurious **local minima**
Experiments and Results

Datasets:
- ADNI: 731 brains
- MindBoggle: 101 brains

Experiments:
- 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  Clusters learned 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

<table>
<thead>
<tr>
<th>Pooling method</th>
<th>Mean ± Std.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Average Pooling</td>
<td>60.76 ± 3.62</td>
</tr>
<tr>
<td>Fixed Parcellation Pooling</td>
<td>64.59 ± 7.84</td>
</tr>
<tr>
<td>Spectral Clustering Pooling</td>
<td>67.94 ± 4.97</td>
</tr>
<tr>
<td>Top-k pooling</td>
<td>78.94 ± 3.32</td>
</tr>
<tr>
<td>Learnable Pooling (ours)</td>
<td>84.21 ± 3.72</td>
</tr>
</tbody>
</table>

*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

### Average accuracy for disease classification

<table>
<thead>
<tr>
<th>Features</th>
<th>Baseline*</th>
<th>Ours without spectral features</th>
<th>Ours with spectral features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cortical thickness + Sulcal Depth</td>
<td>63 ± 4</td>
<td>63.71 ± 5.72</td>
<td>70.79 ± 6.40</td>
</tr>
<tr>
<td>NC vs MCI</td>
<td>65 ± 6</td>
<td>74.03 ± 8.63</td>
<td>76.92 ± 4.78</td>
</tr>
<tr>
<td>MCI vs AD</td>
<td>80 ± 5</td>
<td>76.00 ± 6.06</td>
<td>89.33 ± 4.30</td>
</tr>
</tbody>
</table>

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

**Learnable Graph Pooling, No geometrical information**

**Geometry-based Pooling improves Alzheimer’s Classification**
• **Assumption**: Can our model be used as a biomarker for AD?

• **Prediction** of Alzheimer’s age (or **Geometry age**) differs from Healthy

Learnable pooling

Regression

- Train on NC
- Predict on NC and AD

Train on Healthy (NC)
Predict Age \textit{from Geometry}
← if age is good → Healthy
← if faster aging → Alzheimer’s

\textbf{Geometry-age} of Alzheimer’s Subjects
Deviates from Normal Aging
Contributions: Graph Conv + Pooling

**Graph Convolutions on Spectral Embeddings** for Cortical Surface Parcellation

(1) Graph convolution → ... → Graph convolution → [MedNeurips 2018, MedIA 2019]

**Learnable Pooling** in Graph Convolutional Networks for Brain Surface Analysis

(2) Graph convolution → Pooling → Graph convolution → Classification / Regression [IPMI 2019, TPAMI 2021]
Conclusion: Rethinking **Learning on Surfaces**

Use Spectral Shape Embeddings
Conclusions

1. **Spectral Parameterization**
   - Intrinsic Shape Coordinates
   - \( u_R, u_G, u_B \)

2. **Spectral Graph Convolution**
   - Conv Nets on Brain Surfaces
   - \( (x_i, y_i) \)
   - \( \tilde{u}_i, \tilde{u}_j, f_k \)

3. **Spectral Graph Pooling**
   - Classification / Regression

**Take Home Message**

- **Graph Convolution + Pooling** on Surfaces,
  - \( \leftrightarrow \) Easier with **Spectral Shapes**
  - Simple – Fast Operations on Surfaces
  - Direct Learning on Surface

- Limitations: Meshes of Same Topology (no missing parts)

**Acknowledgments**

- Karthik Gopinath
  - MedIA 2018, IPMI 2019, PAMI 2021
- Prof. Christian Desrosiers
  - MedIA 2018, IPMI 2019, PAMI 2021

**Hervé Lombaert**, Summer School on Deep Learning for Medical Imaging,