Learnable Pooling in Graph Convolutional Networks for Brain Surface Analysis

Karthik Gopinath* Christian Desrosiers Herve Lombaert,
for the Alzheimer's Disease Neuroimaging Initiative†

***Abstract*—Brain surface analysis is essential to neuroscience, however, the complex geometry of the brain cortex hinders computational methods for this task. The difficulty arises from a discrepancy between 3D imaging data, which is represented in *Euclidean* space, and the *non-Euclidean* geometry of the highly-convoluted brain surface. Recent advances in machine learning have enabled the use of neural networks for non-Euclidean spaces. These facilitate the learning of surface data, yet pooling strategies often remain constrained to a single fixed-graph. This paper proposes a new learnable graph pooling method for processing multiple surface-valued data to output subject-based information. The proposed method innovates by learning an intrinsic aggregation of graph nodes based on graph spectral embedding. We illustrate the advantages of our approach with in-depth experiments on two large-scale benchmark datasets. The ablation study in the paper illustrates the impact of various factors affecting our learnable pooling method. The flexibility of the pooling strategy is evaluated on four different prediction tasks, namely, subject-sex classification, regression of cortical region sizes, classification of Alzheimer's disease stages, and brain age regression. Our experiments demonstrate the superiority of our learnable pooling approach compared to other pooling techniques for graph convolutional networks, with results improving the state-of-the-art in brain surface analysis.**

***Index Terms*—Learnable pooling, Graph Convolutional Networks, Brain surface analysis, Alzheimer classification.**

I. Introduction

Brain surface analysis plays a crucial role in understanding the mechanisms of perception and cognition in humans [1]. However, the complex geometry of the brain surface, comprised of intricate folding patterns, poses considerable challenges in neuroscience. Notably, brain imaging data, for instance acquired by magnetic resonance imaging, typically comes in 3D, a *Euclidean* space, while its analysis often focuses on the thin surface of the brain, a *non-Euclidean* space. This fundamental difference between the domains of acquisition and analysis, coupled with the geometrical complexity of brain surfaces, severely hinders computational approaches for brain surface analysis. As an illustration, neighboring 3D voxels in a neuroimage may in fact represent points that are far apart on the brain surface, as shown on Fig. 1. To alleviate this problem, popular surface-based methods [2], [3] often simplify the geometry of the brain, for instance, by mapping the surface to a sphere. This process is, however, computationally expensive. For example, the widely-used surface analysis pipeline of FreeSurfer [2] requires several hours to inflate the cortical surface to a sphere, match it to an atlas and finally perform a cortical analysis. The geometry of brain surfaces similarly complicates other conventional approaches for brain analysis, such as those based on diffeomorphic transformations [4] or on spherical harmonics [5].

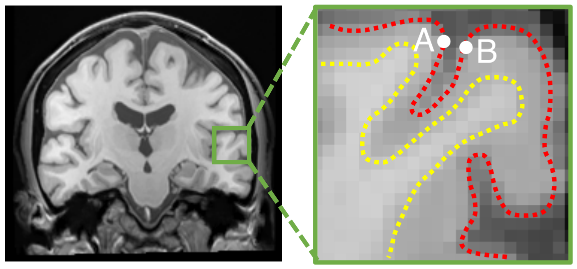


Fig. 1. Complex geometry of the cerebral cortex. As illustrated, two nearby points in the volume may in fact be far apart on the cortical surface.

A key application of brain surface analysis is detecting and tracking the progress of neurodegenerative disorders, such as Alzheimer's disease, which often result in a severe atrophy of brain tissues. Analyzing the geometrical changes of the brain can thus aid in the early diagnosis of such conditions. Initial work has focused on *Euclidean* 3D data based for instance on the texture of magnetic resonance images [6], [7], in order to differentiate Alzheimer's disease from normal aging. While volumetric approaches have shown usefulness in detecting global changes in a Euclidean space [1], surface-based methods [2], [3], [4], [5] are more adequate for analyzing data on brain surfaces. For example, the analysis of shape abnormalities on brain surfaces has improved the prediction of Alzheimer's disease [8] or the identification of stages in this progressive disorder [9]. Nevertheless, all these studies has focused on pre-established measurements of brain surface information. In this paper, we propose to learn and exploit the organizational structure of surface data in order to improve prediction tasks that use data on highly-complex surfaces.

A. Related work

Current machine learning approaches have achieved state-of-the-art performance in a broad range of computer vision and

†Data used in preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. A complete listing of ADNI investigators can be found at: http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf.

All authors are with the Computer and Software Engineering department of ETS Montreal, Canada. *Corresponding author: K. Gopinath. **Email:** karthik.gopinath.1@etsmtl.net.

medical imaging applications. In particular, deep learning architectures such as convolutional neural networks (CNNs) [10] offer higher accuracy and speed over traditional approaches for image analysis. In neuroimaging, CNNs are now widely used for various segmentation [11] and classification [12] problems, with architectures tailored for the target task and the available imaging data. For example, various architectures have been proposed to exploit volumetric data [13], [14], [15], [16]. A fundamental limitation of these models, however, is their restriction to data lying on a fixed Euclidean grid representing pixels or voxels. This restricted representation induces ambiguity when exploiting complex geometries such as in brain surfaces, impeding the application of these Euclidean models for brain surface analysis.

Geometric deep learning [17] generalizes deep learning models to operate on non-Euclidean domains such as graphs and manifolds. Recent advances in this field, particularly in graph convolutional networks (GCNs), have enabled convolution operations over graphs by exploiting spectral analysis, where convolutions translate into multiplications in a Fourier space [18], [19], [20], [21]. In such models, convolutions are manipulated with eigenfunctions of graph Laplacian operators [22], which can be approximated with Chebyshev [20] or Cayley polynomials [23]. These learned convolution filters can be expressed in terms of mixtures of Gaussians [21] or splines [24]. Despite their advantages over standard CNNs, these models are, however, limited to a fixed graph structure and thereby not suitable for brain imaging applications involving a population of subjects. Indeed, brain surfaces have varying geometries with a different number of nodes and a distinct connectivity across meshes. This variability poses computational challenges, for example, arising from the fact that the values of a Laplacian eigenfunction can drastically differ between brains with distinct surface geometries [25]. To this effect, a learned synchronization can correct for differences in eigenfunctions [26]. An alignment of eigenbases [27] similarly provides a common parameterization of brain surfaces. Such aligned eigenbases enabled the direct learning of surface data across multiple brain geometries [28]. Nevertheless, these types of GCNs are limited to a *fixed* graph structure, for instance, with the same number of nodes.

Standard pooling strategies rely in fact on such consistency of graph structures. Currently, heuristics are often used to mimic a max-pooling strategy in GCNs [18], [20], [29]. They include varying the number of feature dimensions across layers [18] while retaining fixed layer sizes, or relying on partition methods, for instance, based binary trees [20] or Graclus clustering [29] to coarsen the initial graph. However, these strategies are mainly used for point-wise operations in fixed-size graphs [21], such as node classification [30], and do not apply to the task of subject classification when the geometry varies across subjects. A few recent studies [31], [32] have attempted to tackle the problem of graph classification in GCNs by incorporating adaptive pooling modules in the network. For instance, [31] performs a hierarchical clustering of nodes using their spectral coordinates, with a subsequent pooling of node features within each cluster. While this approach handles varying graph structures, clusters are defined based only on node proximity in the embedding space, without considering node features. Consequently, this unsupervised pooling strategy may not be optimal for the classification or regression task at hand. More recently, a differential pooling technique [32] splits the network in two separate paths, one for computing latent features for each node of the input graph and another for predicting the node clusters by which features are aggregated. Similarly, [33] proposes to use a top-k graph pooling layer in order to down-sample the input graph. This method selects the top-k nodes for the downsampled graph based on a learned projection vector. However, these approaches ignore the intrinsic localization of nodes within the graph, which is sought when the geometry is highly curved such as in brain surfaces.

B. Contributions

This paper proposes a novel method based on GCNs for classification and regression of surface graphs. Our method includes a learnable pooling strategy which predicts optimal node clusters for each input graph, and thus can handle graphs with varying number of nodes or connectivity. This adaptive pooling technique is applied recursively to obtain a fixed-size representation, which is then used for predicting a target classification or regression value. Our method also leverages spectral embedding techniques for surface graphs [27], offering a more powerful representation of complex surfaces like the brain cortex. This contrasts with the differential pooling approach in [32] or [33], where nodes lack intrinsic localization within the graph.

We illustrate our approach on the challenging tasks of brain surface classification and regression using the well-known Mindboggle [34] and ADNI datasets [35]. We first consider the problem of subject-sex[1] classification and evaluate the impact of our learnable pooling method's hyper-parameters, including the type of pseudo-coordinates, number of clusters, number of eigenvectors, number of neighbors, graph convolution kernel, and input graph size. In an ablation study, we also assess the importance of alignment and regularization for this prediction task. To evaluate the usefulness of our learnable pooling strategy, we compare it against recently-proposed pooling techniques for GCNs.

We show the ability of our pooling strategy to learn important node clusters in a supervised manner by comparing the relationship between these clusters and prominent anatomical regions. To further validate the regions learned by our network, we use it to predict the size of cortical regions as defined by a standard parcellation atlas. Our model is also tested on cortical surface data from the ADNI dataset to (*i*) discriminate between control subjects and subjects suffering from different stages of Alzheimer's, and (*ii*) regress the brain age of subjects. We choose the ADNI dataset [35] as it provides manual labels of the subject age and three stages of Alzheimer's disease. Our method achieves a similar performance to the state-of-the-art on the ADNI dataset [35], while using only simple cortical measurements such as thickness and sulcal depth.

[1]As in most studies, we use the term *sex* instead of *gender* to designate biological differences between male and female subjects.

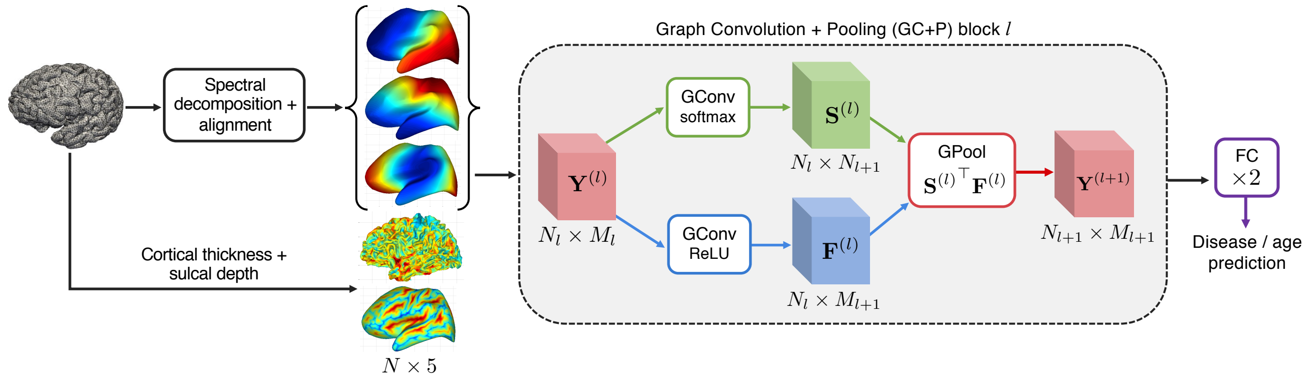


Fig. 2. **An overview of the proposed graph convolutional network:**The brain surface graph are mapped to a low-dimensional subspace using spectral decomposition. The spectral bases of the input brain are then aligned to a common reference. Aligned spectral coordinates and cortical surface features are fed as input to the network, composed of sequential Graph Convolution + Pooling (GC+P) blocks and two fully-connected (FC) layers. Each GC+P block processes input node features $\mathbf{Y}^{(l)}$ in two separate paths based on geometric convolutions, one (bottom) deriving a new set of features for each graph node $\mathbf{F}^{(l)}$ and the other (top) computing a soft assignment $\mathbf{S}^{(l)}$ of nodes to clusters representing nodes of the reduced output graph. A pooling layer then obtains reduced graph features $\mathbf{Y}^{(l+1)}$ by aggregating $\mathbf{F}^{(l)}$ in each predicted cluster of $\mathbf{S}^{(l)}$.

In summary, the major contributions of our work are as follows:

- A general model for classifying and regressing graphs with varying geometry, which combines a learnable, supervised pooling strategy with the intrinsic (non-Euclidean) localization of nodes via graph spectral embedding.
- A first fully-learned model for brain surface analysis contrasting with previous approaches based on predefined cortical features;
- An in-depth experimental evaluation on two large-scale benchmark datasets (i.e., Mindboggle and ADNI) and four different prediction tasks (i.e., subject-sex classification, cortical region size regression, Alzheimer's disease classification, and brain age regression). Our extensive experiments evaluate the impact of the main components and hyper-parameters of our learnable pooling method, and compares our method against four recently-proposed pooling strategies for GCN;
- State-of-the-art performance for ADNI stages classification and brain age prediction using cortical surface data.

This paper represents a significant extension of our previous work in [36]. Beyond giving a deeper motivation of our work and a more detailed description of the methodology, we thoroughly evaluate our method on a large multi-site dataset, i.e. Mindboggle, as well as on two additional prediction tasks, i.e. subject-sex classification and cortical region-size regression. Added experiments also provide a more comprehensive study of the main hyper-parameters and components of our pooling method and demonstrate its advantage over state-of-art graph pooling techniques relying on unsupervised spectral clustering [31], differentiable pooling approaches in Euclidean space [32] and a recent top-k pooling method [33]. Moreover, results of new experiments highlight the relationship between the learned clusters for these tasks and known cortical regions, and show the robustness of our method to surface mesh variability in terms of number of nodes and connectivity.

II. Method

We first describe a general formulation that extends standard convolutions to non-rigid geometries, such as surfaces. We then detail our strategy based on graph spectral embedding to model the intrinsic localization of mesh nodes and align them across multiple surfaces. Subsequently, we present our end-to-end learnable pooling strategy for the adaptive clustering of graph nodes. Finally, we provide detailed information on the overall network architecture and training procedure.

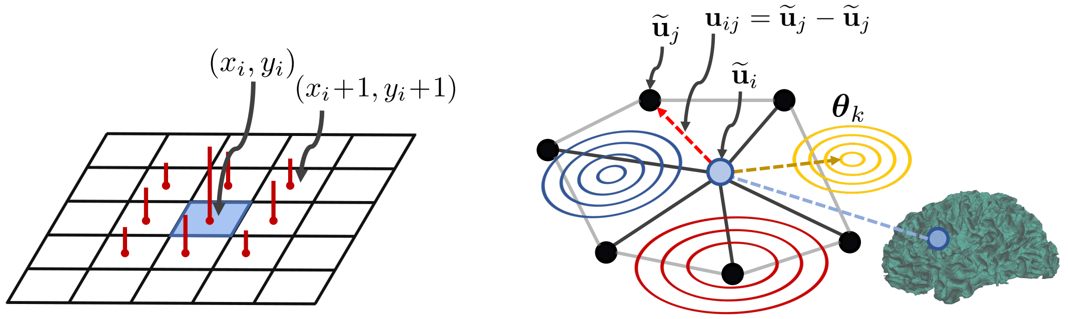
A. *Convolutions on non-rigid geometries*

In a standard CNN, the input is typically provided as a set of features observed over a regular grid of points like 2D pixels or 3D voxels. This information is then processed using a sequence of layers composed of a convolution operation followed by a non-linear activation function like the ReLU. Let $\mathbf{Y}^{(l)} \in \mathbb{R}^{N_l \times M_l}$ be the input feature map at convolution layer l, such that $y_{iq}^{(l)}$ is the q-th feature of the i-th input node. The feature map consists of N_l input nodes with M_l dimensions each. Assuming a 1D grid for simplicity, the output of layer l obtained by a convolution kernel of size K_l is given by $y_{ip}^{(l+1)} = f(z_{ip}^{(l)})$, where

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

Here, $w_{pqk}^{(l)}$ are the convolution kernel weights, $b_p^{(l)}$ the weights of the layer, and f the activation function.

For a general surface, points are not necessarily defined on a regular grid and can lie anywhere in a 3D Euclidean space. Such surface can conveniently be represented as a mesh graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where $\mathcal{V}$ is the set of nodes corresponding to points and $\mathcal{E}$ is the set of edges between the graph nodes. Given a node $i \in \mathcal{V}$, we denote as $\mathcal{N}_i = \{j \,|\, (i,j) \in \mathcal{E}\}$ the set of nodes connected to i, called neighbors. We extend



(x_i, y_i)
$(x_i\!+\!1, y_i\!+\!1)$
$\widetilde{\mathbf{u}}_j$
$\mathbf{u}_{ij} = \widetilde{\mathbf{u}}_j - \widetilde{\mathbf{u}}_j$
$\widetilde{\mathbf{u}}_i$
$\boldsymbol{\theta}_k$

Fig. 3. Illustration of standard grid-based 2D convolutions (left) and geometric graph convolution (right). The challenge is to exploit kernels on arbitrary graph structures, and to add pooling operations over convolutional layers of graph nodes.

the concept of convolution to arbitrary graphs using the more general definition of geometric convolution [21], [28], [24]:

$$z_{ip}^{(l)} \;=\; \sum_{j\in\mathcal{N}_i}\sum_{q=1}^{M_l}\sum_{k=1}^{K_l} w_{pqk}^{(l)} \cdot y_{jq}^{(l)} \cdot \varphi_{ij}(\boldsymbol{\theta}_k^{(l)}) \;+\; b_p^{(l)}, \qquad (2)$$

In this extended formulation, φ_{ij} is a symmetric kernel parameterized by $\boldsymbol{\theta}_k$, which encodes the relative position of neighbor nodes j to a node i when computing the convolution at node i. The pseudo-coordinates $\mathbf{u}_{ij}$ of i relative to j are usually defined based on Cartesian or polar coordinates. In this work, we explore two types of kernels for geometric convolutions: the Gaussian kernel [21] and B-spline kernel [24]. The Gaussian kernel, which has learnable parameters $\boldsymbol{\theta}_k = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$ corresponding to a mean vector and covariance matrix, computes the response as

$$\varphi_{ij}(\boldsymbol{\theta}_k) \;=\; \exp\big(-\tfrac{1}{2}(\mathbf{u}_{ij}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\mathbf{u}_{ij}-\boldsymbol{\mu}_k)\big). \qquad (3)$$

As shown in Fig. 3, standard convolutions (left) can be seen as a special case of geometric convolutions with Gaussian kernels (right) where nodes are placed on a regular grid and kernels are unit impulses (i.e., spherical Gaussian kernels with zero variance) placed at the grid position of neighbor nodes. On the other hand, B-spline kernels obtain the response as the product of M_l B-spline basis functions of degree m based on uniform knot vectors. Compared to Gaussian kernels, this kernel has the advantage of making computation time independent from the kernel size, thereby improving computational efficiency and scalability.

B. Spectral embedding of multiple surface graphs

A significant limitation of the above geometric convolutional model is its inability to process differently-aligned surfaces. Thus, since local coordinates $\mathbf{u}_{ij}$ are determined using a fixed coordinate system, any rotation or scaling of the surface mesh will produce a different response for a given set of kernels. Moreover, as shown in Fig. 1, geometric convolutions in Euclidean space are poorly-suited for complex surfaces like the highly-convoluted brain cortex.

We address these issues using a graph spectral embedding approach. Specifically, we map a surface graph $\mathcal{G}$ to a low-dimensional subspace using the eigencomponents of its normalized Laplacian $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$, where $\mathbf{A}$ is the weighted adjacency matrix and $\mathbf{D}$ is the diagonal degree matrix with $d_{ii} = \sum_j a_{ij}$. Although binary adjacency values could be used in $\mathbf{A}$, we instead define the weight between two adjacent nodes as the inverse of their Euclidean distance: $a_{ij} = \big(\|\mathbf{x}_i - \mathbf{x}_j\|_2 + \epsilon\big)^{-1}$ where ϵ is a small constant to avoid a zero-division. Denoting as $\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$ the eigendecomposition of $\mathbf{L}$, where $\boldsymbol{\Lambda}$ is the diagonal matrix of real, non-negative eigenvalues, we then compute the normalized spectral coordinates of nodes as the rows of matrix $\widehat{\mathbf{U}} = \mathbf{U}\boldsymbol{\Lambda}^{-\frac{1}{2}}$. Here, normalized components are scaled proportionally to the *inverse* of their eigenvalues since components with smaller eigenvalues encode more relevant characteristics of the embedded graph [37]. Based on the same principle and as in [38], we limit the decomposition to the $d=3$ first smallest non-zero eigenvalues of $\mathbf{L}$. This allows capturing the important variability of surfaces, while also limiting computational complexity.

We must align the spectral projection of different surface graphs to a common reference $\widehat{\mathbf{U}}^{\mathrm{ref}}$ because the spectral embedding of $\mathbf{L}$ is only defined up to an orthogonal transformation (i.e., rotation or flip). The spectral embedding of a random brain surface in the dataset is chosen as the common reference $\widehat{\mathbf{U}}^{\mathrm{ref}}$. To perform alignment, we find a node correspondence by using an iterative closest point (ICP) approach [27], where each node $i \in \mathcal{V}$ is mapped to its nearest reference node $\pi(i) \in \mathcal{V}^{\mathrm{ref}}$ in the embedding space. Denoting as $\hat{\mathbf{u}}_i$ the normalized spectral coordinates of node i, the alignment task can be expressed as

$$\operatorname*{arg\,min}_{\pi,\mathbf{R}} \; \sum_{i=1}^{N} \big\| \hat{\mathbf{u}}_i \mathbf{R} \,-\, \hat{\mathbf{u}}_{\pi(i)}^{\mathrm{ref}} \big\|_2^2. \qquad (4)$$

Let $\widehat{\mathbf{U}}_\pi^{\mathrm{ref}}$ be the matrix whose i-th row is $\hat{\mathbf{u}}_{\pi(i)}^{\mathrm{ref}}$. The transformation between corresponding nodes is approximated as

$$\mathbf{R} \;=\; \big(\widehat{\mathbf{U}}^\top \widehat{\mathbf{U}}\big)^{-1} \widehat{\mathbf{U}}^\top \widehat{\mathbf{U}}_\pi^{\mathrm{ref}} \;=\; \boldsymbol{\Lambda}^{\frac{1}{2}} \mathbf{U}^\top \widehat{\mathbf{U}}_\pi^{\mathrm{ref}}. \qquad (5)$$

We use the aligned spectral embedding $\widetilde{\mathbf{U}} = \widehat{\mathbf{U}}\mathbf{R}$ to define the local coordinates corresponding to an edge $(i,j) \in \mathcal{E}$: $\mathbf{u}_{ij} = \tilde{\mathbf{u}}_j - \tilde{\mathbf{u}}_i$. As illustrated in Fig. 3 (right), and based on Eq. (2), the convolution at node i therefore considers kernel responses $\varphi_{ij}(\boldsymbol{\theta}_k^{(l)})$ for neighbor nodes j, *relative to* the spectral coordinates of i.

C. Learnable pooling for graph convolutional networks

Pooling in standard CNNs is typically carried out by aggregating values inside non-overlapping regions of features maps. In graph convolutional networks [18], [19], [20], [21], however, this approach is not applicable for the following reasons. First, nodes are not laid out on a regular grid, which prevents aggregation of features in predefined regions. Second, the density of points may spatially vary in the embedding space; hence regions of fixed size or shape are not suitable for graphs with different geometries. Last, and more importantly, input surface graphs may have a different number of nodes, while the output may have a fixed size. This is the case when predicting a fixed number of class probabilities from different brain geometries.

We propose an end-to-end learnable pooling strategy for the subject-specific aggregation of cortical features, inspired by the differential pooling technique of Ying et al. [32]. Our strategy, shown in Fig. 2, produces a sequence of convolutional feature maps $\{\mathbf{Y}^{(1)},\ldots,\mathbf{Y}^{(l)},\ldots,\mathbf{Y}^{(L)}\}$, with $\mathbf{Y}^{(l)} \in \mathbb{R}^{N_l \times M_l}$, by the repeated application of a Graph Convolution + Pooling (GC+P) block. Each GC+P block takes as input a feature map $\mathbf{Y}^{(l)}$ on a N_l node graph, and processes it in two separate paths: the first one computing latent features for each node of the input graph and the second predicting the node clusters by which the features are aggregated. The feature encoding path applies a sequence of geometric convolutions as in Eq. (2) to generate a new feature map $\mathbf{F}^{(l)} \in \mathbb{R}^{N_l \times M_{l+1}}$ on the block's input graph. The clustering path also consists of sequential geometric convolutions, however the activation function of the last convolution is replaced by a node-wise softmax. The output of this last convolution, $\mathbf{S}^{(l)} \in [0,1]^{N_l \times N_{l+1}}$, gives for each node i the probability s_{ic} that i belongs to cluster $c \in \{1,\ldots,N_{l+1}\}$.

Pooled features $\mathbf{Y}^{(l+1)} \in \mathbb{R}^{N_{l+1} \times M_l}$ are computed as the expected sum of convolutional features in each cluster c, i.e.

$$\begin{aligned} y_{cp}^{(l+1)} &= \sum_{i=1}^{N_l} s_{ic}^{(l)} \cdot f_{ip}^{(l)} \\ \mathbf{Y}^{(l+1)} &= \mathbf{S}^{(l)\top}\mathbf{F}^{(l)}. \end{aligned} \tag{6}$$

The processing of aggregated node features, downstream the pooling operation, requires computing a new adjacency matrix $\mathbf{A}^{(l+1)}$ and spectral coordinates $\widetilde{\mathbf{U}}^{(l+1)}$ for the node clusters which become the nodes of the block's reduced-size output graph. Here, we define the adjacency weights between pooling clusters c and d as

$$\begin{aligned} a_{cd}^{(l+1)} &= \sum_{i=1}^{N_l}\sum_{j=1}^{N_l} s_{ic}^{(l)} \cdot s_{jd}^{(l)} \cdot a_{ij}^{(l)} \\ \mathbf{A}^{(l+1)} &= \mathbf{S}^{(l)\top}\mathbf{A}^{(l)}\mathbf{S}^{(l)}. \end{aligned} \tag{7}$$

Intuitively, $a_{cd}^{(l+1)}$ is the expected number of connected nodes between clusters c and d. Likewise, the spectral coordinates of cluster c is computed as the mean coordinates (i.e., centroid) of all nodes assigned to c:

$$\begin{aligned} \widetilde{u}_{cp}^{(l+1)} &= \sum_{i=1}^{N_l} s_{ic}^{(l)} \cdot \widetilde{u}_{ip}^{(l)} \\ \widetilde{\mathbf{U}}^{(l+1)} &= \mathbf{S}^{(l)\top}\widetilde{\mathbf{U}}^{(l)}. \end{aligned} \tag{8}$$

The bilinear formulation of Eq. (6) faces a challenging optimization problem with several local minima. For instance, the same output $\mathbf{Y}^{(l+1)}$ in Eq (6) can be obtained by modifying either $\mathbf{S}^{(l)}$ or $\mathbf{F}^{(l)}$. To alleviate this problem and obtain spatially-smooth clusters, we add a Laplacian regularization term to the loss function:

$$\begin{aligned} \mathcal{L}_{\mathrm{reg}}(\mathbf{S}^{(l)}) &= \sum_{i=1}^{N_l}\sum_{j=1}^{N_l} a_{ij}^{(l)} \cdot \big\|\mathbf{s}_i^{(l)} - \mathbf{s}_j^{(l)}\big\|_2^2 \\ &= \mathrm{tr}\big(\mathbf{S}^{(l)}\mathbf{L}^{(l)}\mathbf{S}^{(l)\top}\big), \end{aligned} \tag{9}$$

where $\mathbf{s}_i^{(l)}$ denotes the cluster probability vector of node i (i.e., the i-th row of $\mathbf{S}^{(l)}$). This well-known regularization approach [39] penalizes connected nodes to be mapped to different clusters, with a penalty proportional to the connection strength.

D. Architecture details

Figure 2 presents the overall architecture of our graph convolutional network. As input, we give to the network the cortical surface features $\mathbf{x}_i$ and aligned spectral coordinates $\widetilde{\mathbf{u}}_i$ of each node i. For computing graph convolutions as in Eq. (2), we define the neighbors $\mathcal{N}_i$ of node i as the $k=5$ nodes nearest to i in the spectral embedding (i.e., the distance between node i and j corresponds to $\|\widetilde{\mathbf{u}}_i - \widetilde{\mathbf{u}}_j\|_2$) plus node i itself. While various features could be considered to model the local geometry of the cortical surface [2], we considered sulcal depth and cortical thickness in this work, since the first one helps delineate anatomical brain regions [40] and the latter is related to ageing [41] and neurodegenerative diseases such as Alzheimer's [42].

The network comprises two cascaded GC+P blocks, followed by two fully-connected (FC) layers. The first block generates an $N\times 8$ feature map and an $N\times 16$ cluster assignment matrix, in two separate paths, and combines them using the pooling formulation of Eq. (6) to obtain a pooled feature map of 16×8. In the second block, pooled features are used to produce a 16×16 map of features, pooled in a single cluster. Hence, the second pooling step acts as an attention module selecting the features of most relevant clusters. The resulting 1×16 representation is converted to a 1×8 vector using the first FC layer, and then to a $1\times$ *NumOutputs* vector with the second FC layer, where the number of outputs *NumOutputs* depends on the prediction task.

Except for the cluster probabilities and network output, all layers employ the Leaky ReLU [43] as activation function: $y_{ip}^{(l)} = \max(0.01 z_{ip}^{(l)}, z_{ip}^{(l)})$. In the default setting of our pooling method, for the graph convolution kernel φ_{ij} of Eq. (2), we used the B-spline kernel proposed by Fey et al.

TABLE I
IMPACT OF OUR HYPER-PARAMETERS ON OUR LEARNABLE POOLING METHOD. MEAN AND STANDARD DEVIATION WERE COMPUTED ON 5 SEPARATE RUNS USING A DIFFERENT RANDOM 50K NODE SUB-SAMPLING OF EACH GRAPH. FOR EVERY HYPER-PARAMETER, THE DEFAULT SETTING OF OUR METHOD IS HIGHLIGHTED IN BOLD FONT.

Experiments	Parameters	Mean $\pm$ Std.
Pseudo-coordinates	Cartesian	80.40 $\pm$ 4.21
	Polar	83.15 $\pm$ 2.10
	Ours - Spectral	**84.21 $\pm$ 3.72**
Number of clusters	4	73.68 $\pm$ 5.76
	8	76.84 $\pm$ 7.87
	16	**84.21 $\pm$ 3.72**
	32	77.89 $\pm$ 2.10
Number of eigenvectors	Only cortical features	70.52 $\pm$ 5.36
	1	75.78 $\pm$ 7.13
	3	**84.21 $\pm$ 3.72**
	5	77.89 $\pm$ 2.10
	10	74.73 $\pm$ 8.40
Number of neighbors	2	81.05 $\pm$ 2.57
	3	82.10 $\pm$ 2.57
	5	**84.21 $\pm$ 3.72**
	10	84.21 $\pm$ 3.93
Graph convolution kernel	Gaussian [21]	83.15 $\pm$ 2.15
	B-Spline [24]	**84.21 $\pm$ 3.72**
Ablation study	W/o Alignment	69.47 $\pm$ 8.42
	W/o Regularization	74.73 $\pm$ 5.15

[24]. However, we also test the Gaussian kernel [21] in our experiments.

For training, the loss function combines the output prediction loss and cluster regularization loss on the first GC+P block:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\mathrm{out}}(\boldsymbol{\theta}) + \alpha \mathcal{L}_{\mathrm{reg}}\big(\mathbf{S}^{(1)}\big), \tag{10}$$

where α is a parameter controlling the amount of regularization. For classification tasks (i.e., disease prediction), $\mathcal{L}_{\mathrm{out}}$ is set as the cross-entropy between one-hot encoded ground-truth labels and output class probabilities. In the case of regression (i.e., brain age prediction), we use mean squared error (MSE) for this loss. Network parameters are optimized with stochastic gradient descent (SGD) using the Adam optimizer. Experiments were carried out on an i7 desktop computer with 16GB of RAM and an Nvidia Titan X GPU. The model takes less than a second for disease classification or age regression.

III. EXPERIMENTS AND RESULTS

We validate our method on two large-scale, publicly-available datasets: Mindboggle-101 [34] and ADNI1 [35]. The first one contains T1-weighted MRI from 101 healthy subjects (males: n=57, females: n=44, age: 20–61 years) collected from 9 different sites. We use this dataset for the tasks of subject-sex classification and cortical region size regression since both subject-sex labels and manual annotations for 32 cortical parcels are provided with imaging data. The ADNI1 dataset [35] is comprised of multi-sequence MRI data from 400 subjects diagnosed with mild cognitive impairment (MCI), 200 subjects with early Alzheimer's disease (AD) and 200 elderly control subjects (NC), obtained from 55 participating sites. Both datasets contain brain surface meshes with pointwise cortical thickness and sulcal depth measurements, generated by FreeSurfer[2]. Cortical meshes in these datasets vary from 102K to 185K nodes. The code for our work is available at the following URL: https://github.com/kharitz/learnpool.git.

In the first series of experiments, we evaluate the effects of hyper-parameters influencing the performance of our pooling method. Next, an ablation study is presented to assess the effect of our spectral alignment and our Laplacian regularization. Different pooling strategies for our graph convolutional network are thereafter compared on the subject-sex classification problem, while also evaluating the impact of input graph size on prediction accuracy. We then illustrate our network's ability to learn meaningful node clusters by predicting the size of cortical parcels from an anatomical atlas. Finally, we highlight the advantages of working in the spectral domain on the problems of disease classification (NC *vs* AD, MCI *vs* AD, and NC *vs* MCI) and brain age regression.

A. *Impact of hyper-parameters*

Our learnable pooling method requires the selection of several hyper-parameters: the type of pseudo-coordinates, the number of clusters, the number of eigenvectors, the number of neighbors, and the type of graph convolutions. In the next series of experiments, we assess the impact of each of these hyper-parameters on the task of subject-sex classification with the MindBoggle dataset, using a 70-10-20 split for training, validation, and testing. To have a measure of variance, keeping the same split, we generated 5 different subsets by randomly sub-sampling 50K nodes in each training, validation and testing graph, and used the sub-sampled graphs as input to our model. Performance (mean and standard deviation) is measured across 5 runs, each one carried out on a different subset. The same architecture, shown in Fig. 2, is used across the following experiments.

1) Pseudo-coordinates: We first evaluate the benefit of using spectral information when computing the pseudo-coordinates of nodes in the graph convolution kernel, by comparing it against conventional Cartesian and polar coordinates. The same architecture of Fig. 2, based on B-spline kernels, is used for all three settings. As reported in Table I accuracy improvements of 3.81% and 1.06% are obtained over Cartesian and polar coordinates, respectively, showing the ability of spectral pseudo-coordinates to better capture the local geometry of a complex surface. Note that, to have a fair comparison, spectral node coordinates were used as input to the network in all three settings, hence the models using Cartesian and polar pseudo-coordinate also leverage spectral information. Comparing Cartesian and polar pseudo-coordinates together, we find that polar ones provide a higher accuracy. While both encode similar information, polar coordinates offer a more direct description of distance and direction between two points, which could help to learn their relation. This may explain why polar pseudo-coordinates were preferred in earlier work [21].

2) Number of clusters: Next, we train our GCN network using different numbers of clusters for the pooling operation of the network's first GC+P block. As presented in Table I,

[2]https://surfer.nmr.mgh.harvard.edu/

four settings are tested: 4, 8, 16 and 32 clusters. We see a regular increase in accuracy from 73.64% to 84.21% when going from 4 to 16 clusters. This reflects the fact that sex-related differences are present in various cortical regions, which can be learned by the network. However, the accuracy drops significantly when further increasing the number of clusters to 32. This could be due to the creation of near-empty clusters that add no useful information to the training while increasing the number of parameters to learn.

3) Number of eigenvectors: The inputs of our GCN are the aligned spectral components (the Laplacian matrix eigenvectors) and two cortical features, i.e., sulcal depth and cortical thickness, corresponding to each mesh node. In the next experiment, we vary the number of spectral components given as input, testing five different settings: 0 (only cortical features), 1, 3, 5, and 10. For all settings, three eigenvectors are used to compute pseudo-coordinates in the graph convolutions, and the same 70-10-20 split as the previous experiments is employed. Results presented in Table I demonstrate the importance of including spectral information as input, with an accuracy improvement of 5.26% when adding the first component (i.e., eigenvector with smallest non-zero eigenvalue) to cortical features. The best performance of 84.21% is achieved when considering the first three eigenvectors in addition to cortical features. A possible explanation for this result is that three is the minimal number of eigenvectors required to uniquely locate a point on a 3D surface [27]. As suggested by the decreasing accuracy for 5 and 10 spectral components, higher-order eigenvectors may capture highly-varying and subject-specific patterns of sulcal and gyral geometry, which is not predictive of subject sex.

4) Number of neighbors: The number of neighbors k directly impacts the computation of convolutions in Eq. (2). To better assess the effect of this hyper-parameter, the performance of a classification task is evaluated while increasing the number of neighbors within randomly sub-sampled graphs of 50K nodes. More precisely, for every node i in the graph, the k nearest neighbors are defined in the spectral embedding space. The smoothness of the Laplacian matrix eigenvectors ensures that neighbors are locally close to each other on the brain surface. Performance is then evaluated using a classification model that is trained on sub-sampled graphs with $k=2$, 3, 5, and 10 neighbors.

Table I shows a higher classification accuracy when the number of neighbors increases. From a classification accuracy of 81.05% for $k=2$ neighbors, the performance improves gradually to 84.21% with $k=5$ and $k=10$ neighbors. However, the computational overhead of employing larger neighborhoods must also be taken into account. For instance, runtime increases by a factor of 1.7, from 93.6 ms to 158.5 ms, when going from $k=5$ to $k=10$. For this reason, a neighborhood size of $k=5$ is used in the default setting of our method.

5) Graph convolution kernel: Last, we compare the B-spline convolution of our default architecture with the Gaussian kernel of [21] with diagonal covariance matrix. As reported in Table I, we observe a small improvement of 1.06% when employing B-spline kernels. Interestingly, the number of parameters is almost the same for both kernel types (2,257 for Gaussian compared to 2,233 for B-spline) and, thus, performance differences are not due to overfitting. Note that similar improvements of B-spline kernels compared to Gaussian kernels were also observed in [24] for various analysis tasks.

B. Ablation study on alignment and regularization

In this section, we perform an ablation study to evaluate the contribution of spectral alignment and Laplacian regularization on our method's performance for the task of predicting the sex of Mindboggle subjects. To assess the usefulness of spectral alignment, we first train a model with unaligned spectral coordinates and with cortical features. The results in Table I show the unaligned components to yield a low accuracy of 69.47%, demonstrating the importance of this alignment for learning across different surfaces. Results also highlight the role of Laplacian regularization in learning, with a 9.48% drop in accuracy when removing the corresponding term in the loss of Eq. (10), i.e. using $\alpha=0$ in the loss. As explained before, this regularization term is necessary to avoid getting stuck in a local minimum caused by the bi-linear formulation of the pooling operation. Laplacian regularization also provides spatially-smoother clusters that better reflect the underlying anatomy of the brain.

C. Comparison of different pooling methods

We compared our learnable pooling strategy against four other pooling techniques applicable to graph convolutional networks: 1) taking the global average of feature maps, 2) pooling feature maps in fixed regions computed from a cortical parcel atlas, 3) pooling the same features in regions obtained by applying k-means clustering on the spectral embedding, 4) the top-k pooling approach proposed in [33] for downsampling. For all tested methods, we used a network composed of two graph convolution layers followed by two fully-connected layers, as described in Section II-D. In the case of global average pooling and fixed parcellation pooling, a single pooling operation is applied after the second graph convolution. For spectral clustering pooling, nodes are grouped after each of the two convolution layers as in our learnable pooling. However, the pooling path of the network is replaced by a static node clustering. Likewise, for top-k pooling, we employ the same architecture as presented in Section II-D, but replace our pooling path with the top-k pooling after the graph convolution operation. We train and test all methods on subject-sex classification using the MindBoggle dataset with a 70-10-20 split for training, validation, and testing. Once again, we perform 5 separate runs with a different random sub-sampling of 50K nodes for each graph.

Table II summarizes the results of this experiment. We see that global average pooling yields the poorest performance with a mean accuracy of 60.76%. Using atlas-defined cortical parcels to aggregate features improves accuracy slightly to 64.59%, suggesting that these parcels are informative for identifying subject sex. Moreover, applying unsupervised clustering on the spectral embedding further increases mean accuracy to 67.94%, which indicates the benefits of having a

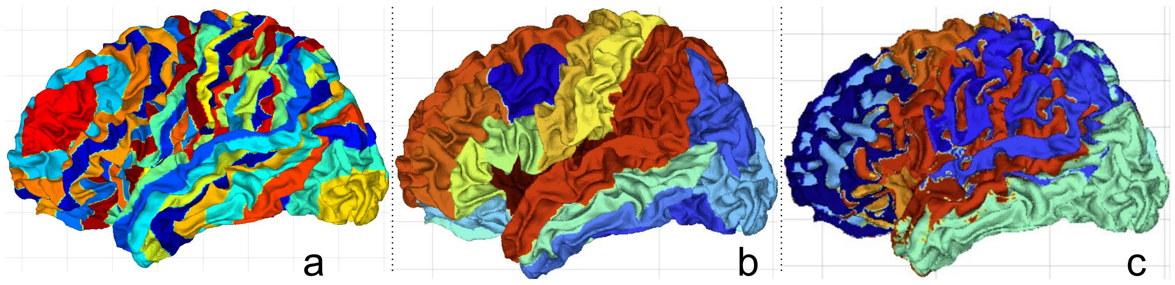


Fig. 4. **Clusters of different pooling methods**: (a) Clusters obtained by spectral k-means clustering. (b) Fixed clusters computed from a cortical parcel atlas. (c) Clusters learned by our learnable pooling method. Colors on the brain surface represent different regions.

TABLE II
BASELINE METHODS COMPARISON. MEAN AND STANDARD DEVIATION WERE COMPUTED ON 5 SEPARATE RUNS USING A DIFFERENT RANDOM 50K NODE SUB-SAMPLING OF EACH GRAPH.

Pooling method	**Mean $\pm$ Std.**
Global Average Pooling	60.76 $\pm$ 3.62
Fixed Parcellation Pooling	64.59 $\pm$ 7.84
Spectral Clustering Pooling [31]	67.94 $\pm$ 4.97
Top-k pooling [33]	78.94 $\pm$ 3.32
Learnable Pooling (ours)	**84.21 $\pm$ 3.72**

hierarchy of non-fixed clusters. The advantage of a learnable top-k pooling over fixed pooling methods can be seen with a classification accuracy of 78.92%. However, by learning clusters in a supervised manner from spectral embeddings, our method achieves the highest accuracy of 81.33%, an improvement of 13.39% over spectral clustering and a 5.3% gain over top-k pooling.

Figure 4 gives examples of clusters for the different pooling strategies (except global average pooling, which considers all nodes as part of a single cluster and top-k pooling as it selects nodes to drop for downsampling). While spectral clustering yields spatially-regular clusters, the distribution of these clusters is arbitrary and does not seem to match known parcels of the cortex (shown in Fig. 4b). In contrast, the clusters predicted by our pooling strategy are larger and better aligned with these known parcels.

D. Impact of input graph size

In the next experiment, we investigate whether our method is robust to variability in the size of the surface mesh. Toward this goal, we use the same split of the MindBoggle dataset as in the first experiment, and randomly sub-sample the original mesh to 100, 1K, 5K, 10K, 25K, 50K and 75K nodes. Because convolutions at each node use information from its $k = 5$ nearest neighbors, as described in Eq. (2), testing multiple sub-sampling with the same number of nodes also assesses the robustness of our model to variations in graph connectivity. We train our model on each of these reduced graph datasets to predict the sex of MindBoggle subjects.

Table III-D gives the classification accuracy for different sizes of training graphs when testing on sub-sampled graphs of the same size, or on the original full-sized graph. The first case evaluates whether the same accuracy can be achieved with less information at the input of the network, whereas the second case tests if the convolution parameters learned by the network generalize to larger graphs. As expected, classification performance decreases when reducing the size of input graphs, both when testing on sub-sampled graphs and full-sized graphs. When testing on sub-sampled graphs, accuracy drops from 94.73% while training with full graphs to 55.02% for graphs with only 100 nodes. However, high accuracy values of 84.21% and 85.26% can be achieved when training graphs of 50K and 75K nodes, respectively, about half the size of the original graphs. Furthermore, we see that our model trained with moderately-reduced graphs can still perform well on full-sized ones. For instance, the model trained with graphs of 50K nodes and 75K nodes achieves an accuracy of 78.94% and 84.21% respectively, when tested on original graphs with twice the number of nodes.

TABLE III
SUBJECT-SEX CLASSIFICATION PERFORMANCE OF OUR POOLING APPROACH ON DIFFERENT SUB-GRAPHS: MEAN CLASSIFICATION ACCURACY (%) WITH STANDARD DEVIATION OVER TEST SET FROM THE MINDBOGGLE DATASET.

Num. of nodes	**Testing on Sub-sampled graphs**	**Testing on Full graphs**
100	55.02 $\pm$ 13.18	52.63 $\pm$ –
1k	55.98 $\pm$ 4.25	52.63 $\pm$ –
5k	64.11 $\pm$ 1.58	47.36 $\pm$ –
10k	67.94 $\pm$ 5.98	52.63 $\pm$ –
25k	71.77 $\pm$ 4.86	73.68 $\pm$ –
50k	84.21 $\pm$ 3.72	78.94 $\pm$ –
75k	85.26 $\pm$ 3.93	84.21 $\pm$ –
Full graph	94.73 $\pm$ –	94.73 $\pm$ –

E. Task-specific pooling regions

In this section, we qualitatively and quantitatively evaluate the predicted clusters and feature maps learned by our network. Once more, we consider the task of classifying males *vs.* females subjects from the Mindboggle dataset with the architecture depicted in Fig. 2.

Figure 5 shows examples of features and clusters learned by our graph pooling model for a male and a female subject. The first and third columns give the distribution of four different activation maps learned by the network for the two subjects. The mean activation in each predicted cluster for the same subjects is illustrated in the second and fourth columns of

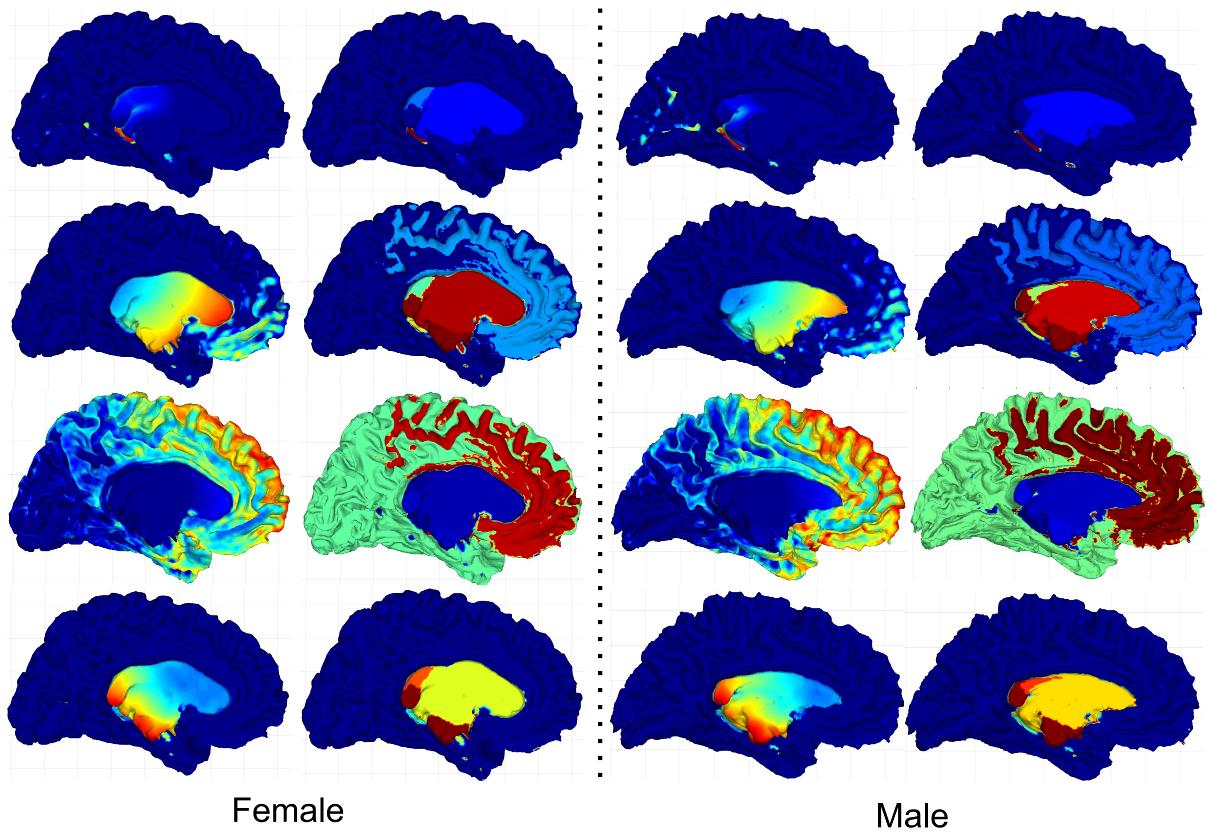


Fig. 5. **Feature maps and predicted clusters for the task of subject-sex classification**: The first column shows examples of activation maps computed by the embedding path of our network for a female subject. The second column gives the average activation in each predicted cluster for the same subject and feature maps. Coloring indicates output of the ReLU activation with minimum value indicated by blue and maximum value indicated by red. Third and fourth columns depict the same information for a male subject.

the figure. We observe the diversity of depicted clusters, spawning different regions of the brain both on the cortex and around regions of the basal ganglia. Interestingly, several of the learned clusters focus on sub-cortical regions like the hippocampus (first row) and amygdala (last row), which have been linked to sex-related differences in the literature [44]. This illustrates the benefit of learning task-specific clusters in a supervised manner. Additionally, we see that predicted feature maps and clusters in both subjects are similar, demonstrating that the model can adapt to the specific brain geometry of individual subjects.

We further evaluate the relevance of learned clusters by training the same model to predict the size of 32 anatomical parcels of each brain surface, using labeled data from Mindboggle. For this experiment, we hypothesize that the network should learn clusters that are related to the predefined parcels. To do so, we modify the last layer of the architecture in Fig. 2 to have 32 outputs, one for the size of each parcel, and change the loss function to mean square error. Adjusted mutual information (AMI) is used to measure the similarity between learned clusters and ground-truth parcels. AMI values range from 0 to 1, a score of 0 corresponding to random clusters and a score of 1 for clusters identical to ground-truth.

Figure 7 gives the mean AMI obtained at each training epoch, and examples of predicted clusters at four different epochs are shown in Fig. 6. In the initial stages of training, the model predicts a small number of clusters corresponding mainly to the components of the spectral embedding (see the network input in Fig. 2). In the first 500 epochs, the AMI score between predicted clusters and ground-truth parcels drops. Then, as training progresses, we observe increasing AMI values and progressively more defined clusters. At the end of training (2500 epochs), the model achieves an AMI score of 0.39. Obtained clusters appear to be a combination of different ground-truth parcels, suggesting that fully-connected layers further help regressing parcel sizes.

F. *Disease classification*

In the following experiment, we evaluate our method on the task of classifying subjects from the ADNI dataset as normal control (NC), mild cognitive impairment (MCI) or Alzheimer's disease (AD). Specifically, we consider three different binary classification problems: NC *vs* AD, MCI *vs* AD and NC *vs* MCI. We compare our method against the random forest approach in [45], which also considers surface-based information from the ADNI dataset. To measure the contribution of the spectral embedding in our method, we also evaluate our model trained with only cortical thickness and sulcal depth as input. The same random split of 70-10-20 is employed for all three models.

The classification performance of tested models is reported in Table IV. We see that our method outperforms the random

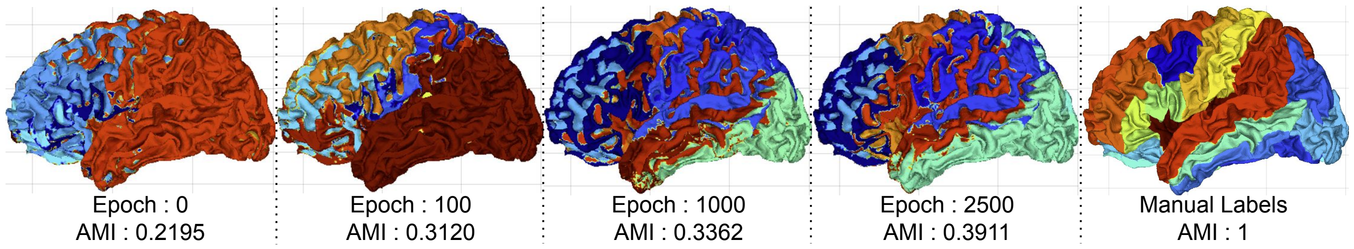


Fig. 6. **Pooling regions learned during training:** The pooling regions are learned for the model training to regress the size of cortical regions. During initial epochs, random regions are clustered together to aggregate feature maps. A low AMI score indicates this random clustering compared to the ground-truth. After training, the model finally learns to group multiple parcels (cyan) into on cluster pooling region. AMI score increases over epochs, indicating task-dependent learning by our model. The last figure shows manual parcels with AMI score of 1 for reference.

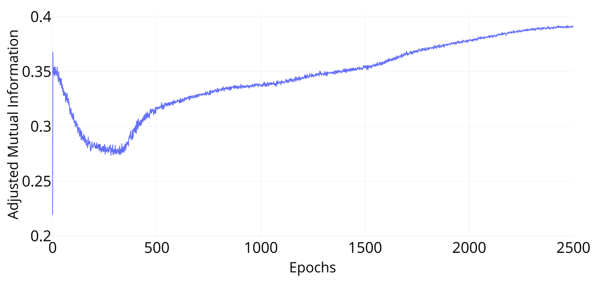


Fig. 7. **Evolution of AMI score**: The adjusted mutual information score between the pooling regions and the manual parcels over multiple epochs is shown. A random overlap between learned pooling regions and parcels is observed at initial epochs. After training, the AMI score increases with the pooling regions corresponding to ground-truth parcels.

forest approach of [45] on all three classification problems. Relative to this approach, the proposed method yields mean accuracy improvements between 7.79% and 11.92%. A significant gain in performance is also observed when comparing the same method trained without spectral node coordinates. This is particularly notable for NC *vs* MCI, where adding spectral coordinates increases the mean accuracy by 13.33%. Note that we have also tried giving the network original (x, y, z) coordinates of mesh nodes. However, this led to worse results. This illustrates the advantage of using intrinsic node localization when processing surface data.

TABLE IV
EVALUATION OF THE PROPOSED WORK: AVERAGE ACCURACY OF DISEASE CLASSIFICATION (%), WITH STANDARD DEVIATION OVER THE COMPLETE ADNI DATASET. FIRST ROW IS A RANDOM FOREST (RF) WITH MULTIPLE CORTICAL-BASED FEATURES [45]. SECOND ROW IS OUR GRAPH CONVOLUTIONAL MODEL WITHOUT GEOMETRICAL INFORMATION (SPECTRAL NODE COORDINATES $\widetilde{\mathrm{U}}$). LAST ROW IS WITH THIS INFORMATION.

Input	**NC *vs* AD**	**MCI *vs* AD**	**NC *vs* MCI**
RF [45]	80 ± 5	65 ± 6	63 ± 4
Ours w/o $\widetilde{\mathbf{U}}$	76.00 ± 6.06	74.03 ± 8.63	63.71 ± 5.72
Ours with $\widetilde{\mathbf{U}}$	89.33 ± 4.30	76.92 ± 4.78	70.79 ± 6.40

G. Brain age prediction

The last experiment demonstrates our method in a regression problem where the age of NC subjects of the ADNI dataset is predicted using pointwise surface-based measurements. In this case, the network outputs a single value, and MSE is used as loss function. Once more, we test our method trained with or without spectral node coordinates as input. Moreover, to evaluate brain age prediction as a potential imaging biomarker for Alzheimer's, we also measure the prediction accuracy of our model on AD test subjects.

The results of this experiment are summarized in Fig. 8, which gives the distribution of mean absolute error (MAE) and prediction bias (predicted age minus real age) for NC subjects and AD subjects. When testing on NC subjects, our method achieves an MAE of 4.35 ± 3.19 years, which is comparable with results in the literature. As expected, a higher MAE of 6.80 ± 6 years is obtained for AD subjects, since the symptoms of early Alzheimer's are similar to premature brain aging. The brain age, calculated as the predicted age minus the real age, shows a statistically significant difference with a p-value of 0.0032. This value suggests the potential application of brain age prediction as a biomarker for AD.

IV. CONCLUSION

We presented a novel strategy that enables pooling operations on arbitrary graph structures. The performance of our learnable pooling scheme was evaluated in seven experiments.

The first series of experiments evaluated the impact of hyper-parameters: the type of pseudo-coordinates of nodes in graph convolution kernels, showing improvement when employing our spectral-based coordinates instead of Cartesian or polar-based ones; the number of clusters in pooling operations, with a regular increase of performance up to 16 clusters; the number of eigenvectors, suggesting that a minimal number of three Laplacian eigenvectors is necessary for optimal accuracy; the number of neighbors, revealing a compromise between accuracy and computation time; the type of graph convolution kernel, showing an improvement of accuracy when using B-spline convolution kernels in our default architecture instead of Gaussian kernels.

A second experiment provided an ablation study validating the positive effects of spectral alignment and Laplacian regularization in our method. Results showed a significant performance gain when using both techniques, compared to employing only one of them.

A third experiment compared different pooling techniques for graph convolutional networks on the subject-sex classification task. A simple global average pooling failed to capture

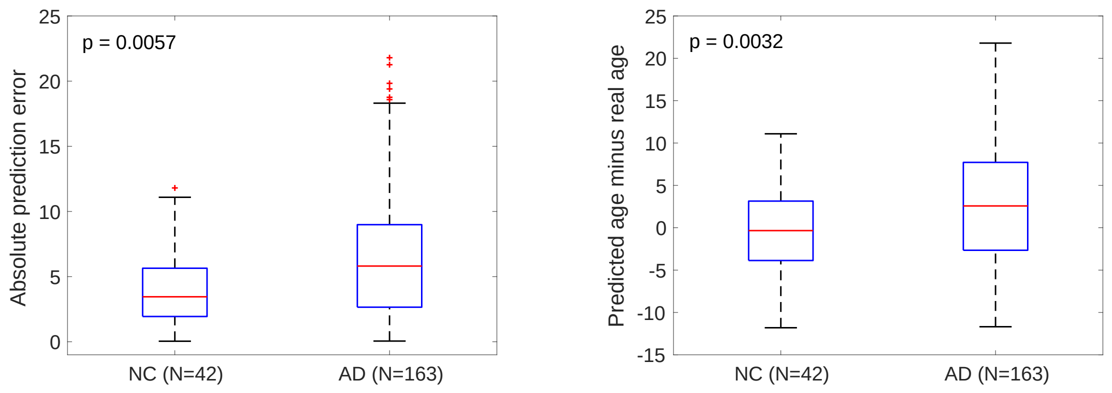


Fig. 8. **Distribution of absolute prediction error** (left) and predicted minus real age (right), for NC and AD test subjects. Our learnable pooling strategy yielded graph models that could correctly capture age discrepancies between real and geometry-based ages, as expected between subjects with NC and AD.

geometric information from consecutive layers, yielding a low performance of 60%. In comparison to employing fixed pooling regions, learning these regions with unsupervised clustering, or applying the top-k pooling strategy to select nodes from a learned projection vector, our learnable pooling strategy offers significantly higher accuracy.

A fourth experiment evaluated the effect of the graph size on the performance of subject-sex classification. The results showed that small graphs lack information to capture the complete geometry of surfaces. However, reducing the size of the graph by 25% up to 75K still yields a high accuracy, while improving memory and computational requirements.

The fifth experiment explored the relationship between learned features and anatomy. The visualization of activation maps and clusters in the network revealed diversity in terms of brain regions. Several learned clusters highlighted essential regions of the basal ganglia, such as the hippocampus and amygdala, which are associated with sex-related differences in the literature. We further evaluated this result with an experiment to regress the size of cortical parcels. As expected, the trained model learns pooling regions similar to the manually-annotated parcels.

The sixth experiment focused on predicted stages of Alzheimer's disease from surface data, including cortical thickness and sulcal depth. Our results showed that pointwise surface values could be efficiently aggregated into a fixed number of class probabilities using the proposed network architecture. Compared to another approach exploiting surface-based features [45], our method achieved significant improvements ranging from 7% to 11%. This performance gain is mainly due to including spectral coordinates of graph nodes as input to the network, demonstrating the importance of intrinsic node localization.

In a final experiment, the age of ADNI subjects was predicted using pointwise surface data. Results showed that our method provides an accuracy comparable to previous approaches in the literature, while using only surface-based information. As expected, subjects with Alzheimer's have higher discrepancies than subjects with normal cognition (Fig. 8). The potential of the proposed method as an imaging biomarker for AD could be evaluated in a future study.

To summarize, the proposed pooling strategy enables the exploration of a new family of architectures for graph convolutional networks. Our method exploits the spectral embeddings of graph nodes in order to learn spatially-representative pooling patterns across different layers. However, this requires having datasets of comparable brain geometry, since the eigen-decomposition of the graph Laplacian matrix assumes that shapes are topologically equivalent. Differences in the meshing procedure as well as the presence of holes or cuts in the mesh, for instance caused by ablated tumors, might therefore impact the performance of our method. In future work, we plan to investigate domain adaptation techniques, for example based on adversarial learning, to learn an internal representation which is robust to such differences. Moreover, by incorporating unpooling operations in the proposed model, we could also explore applications requiring node-level outputs like regressing cortical thickness over time.

Acknowledgment – This work is supported by the Canada Research Chair on Shape Analysis in Medical Imaging, the Research Council of Canada (NSERC), the Fonds de recherche du Québec - Nature et technologies (FRQNT), and NVIDIA Corporation with the donation of a Titan Xp GPU. Part of the data were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database.

References

[1] Arbabshirani, M.R., Plis, S., Sui, J., Calhoun, V.D.: Single subject prediction of brain disorders in neuroimaging: Promises and pitfalls. NeuroImage (2017)
[2] Fischl, B., van der Kouwe, A., Destrieux, C., Halgren, E., Segonne, F., Salat, D.H., Busa, E., Seidman, L.J., Goldstein, J., Kennedy, D., Caviness, V., Makris, N., Rosen, B., Dale, A.M.: Automatically parcellating the cortex. Cereb. Cortex (2004)
[3] Yeo, B.T., Sabuncu, M.R., Vercauteren, T., Ayache, N., Fischl, B., Golland, P.: Spherical demons: Fast diffeomorphic surface registration. TMI (2010)
[4] Glaunes, J., Trouve, A., Younes, L.: Diffeomorphic matching of distributions: a new approach for unlabelled point-sets and sub-manifolds matching. In: CVPR. (2004)
[5] Styner, M., Oguz, I., Xu, S., Brechbühler, C., Pantazis, D., Levitt, J.J., Shenton, M.E., Gerig, G.: Framework for the statistical shape analysis of brain structures using SPHARM-PDM. Insight journal (2006)
[6] Vemuri, P., Gunter, J.L., Senjem, M.L., Whitwell, J.L., Kantarci, K., Knopman, D.S., Boeve, B.F., Petersen, R.C., Jack Jr, C.R.: Alzheimer's disease diagnosis in individual subjects using structural mr images: validation studies. Neuroimage (2008)

[7] Freeborough, P.A., Fox, N.C.: Mr image texture analysis applied to the diagnosis and tracking of alzheimer's disease. TMI (1998)

[8] Tang, X., Holland, D., Dale, A.M., Younes, L., Miller, M.I., Initiative, A.D.N.: Shape abnormalities of subcortical and ventricular structures in mild cognitive impairment and alzheimer's disease: detecting, quantifying, and predicting. Human brain mapping (2014)

[9] Oliveira Jr, P.P.d.M., Nitrini, R., Busatto, G., Buchpiguel, C., Sato, J.R., Amaro Jr, E.: Use of svm methods with surface-based cortical and volumetric subcortical measurements to detect alzheimer's disease. Journal of Alzheimer's Disease (2010)

[10] Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. ISP (1998)

[11] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI. (2015)

[12] Wachinger, C., Reuter, M., Klein, T.: DeepNAT: Deep convolutional neural network for segmenting neuroanatomy. NeuroImage (2017)

[13] Zhang, T., Davatzikos, C.: ODVBA: Optimally-discriminative voxel-based analysis. TMI (2011)

[14] Hua, X., Hibar, D.P., Ching, C.R., Boyle, C.P., Rajagopalan, P., Gutman, B.A., Leow, A.D., Toga, A.W., Jack, C.R., Harvey, D., Weiner, M.W., Thompson, P.M., Alzheimer's Disease Neuroimaging Initiative: Unbiased tensor-based morphometry: Improved robustness and sample size estimates for Alzheimer's disease clinical trials. NeuroImage (2013)

[15] Dolz, J., Desrosiers, C., Ben Ayed, I.: 3D fully convolutional networks for subcortical segmentation in MRI: A large-scale study. NeuroImage (2017)

[16] Kamnitsas, K., Ledig, C., Newcombe, V.F.J., Simpson, J.P., Kane, A.D., Menon, D.K., Rueckert, D., Glocker, B.: Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. MedIA (2017)

[17] Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: Going beyond Euclidean data. TSP (2017)

[18] Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and locally connected networks on graphs. In: ICLR. (2014)

[19] Kipf, T.N., Welling, M.: Semi-Supervised classification with graph convolutional networks. In: ICLR. (2017)

[20] Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: NIPS. (2016)

[21] Monti, F., Boscaini, D., Masci, J., Rodolà, E., Svoboda, J., Bronstein, M.M.: Geometric Deep Learning on Graphs and Manifolds Using CNNs. In: CVPR. (2017)

[22] Xu, Y., Fan, T., Xu, M., Zeng, L., Qia, Y.: SpiderCNN: Deep learning on point sets with parameterized convolutional filters. In: ECCV. (2018)

[23] Levie, R., Monti, F., Bresson, X., Bronstein, M.M.: CayleyNets: Graph convolutional neural networks with complex rational spectral filters. In: ICLR. (2018)

[24] Fey, M., Lenssen, J.E., Weichert, F., Müller, H.: SplineCNN: Fast geometric deep learning with continuous B-Spline kernels. In: CVPR. (2018)

[25] Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., Guibas, L.: Functional maps: A flexible representation of maps between shapes. In: SIGGRAPH. (2012)

[26] Yi, L., Su, H., Guo, X., Guibas, L.J.: SyncSpecCNN: Synchronized spectral CNN for 3D shape segmentation. In: CVPR. (2017)

[27] Lombaert, H., Arcaro, M., Ayache, N.: Brain Transfer: Spectral Analysis of Cortical Surfaces and Functional Maps. In: IPMI. (2015)

[28] Gopinath, K., Desrosiers, C., Lombaert, H.: Graph convolutions on spectral embeddings for cortical surface parcellation. MedIA (2019)

[29] Dhillon, I.S., Guan, Y., Kulis, B.: Weighted graph cuts without eigenvectors a multilevel approach. PAMI (2007)

[30] Parisot, S., Ktena, S.I., Ferrante, E., Lee, M., Guerrero Moreno, R., Glocker, B., Rueckert, D.: Spectral Graph Convolutions for Population-Based Disease Prediction. In: MICCAI. (2017)

[31] Wang, C., Samari, B., Siddiqi, K.: Local spectral graph convolution for point set feature learning. In: ECCV. (2018)

[32] Ying, R., You, J., Morris, C., Ren, X., Hamilton, W.L., Leskovec, J.: Hierarchical graph representation learning with differentiable pooling. In: NeurIPS. (2018)

[33] Gao, H., Ji, S.: Graph U-Nets. In: ICML. (2019)

[34] Klein, A., Ghosh, S.S., Bao, F.S., Giard, J., Häme, Y., Stavsky, E., Lee, N., Rossa, B., Reuter, M., Chaibub Neto, E., Keshavan, A.: Mindboggling morphometry of human brains. PLOS Computational Biology (2017)

[35] Jack Jr, C.R., Bernstein, M.A., Fox, N.C., Thompson, P., Alexander, G., Harvey, D., Borowski, B., Britson, P.J., L. Whitwell, J., Ward, C., et al.: ADNI: MRI methods. JMRI (2008)

[36] Gopinath, K., Desrosiers, C., Lombaert, H.: Adaptive graph convolution pooling for brain surface analysis. In: International Conference on Information Processing in Medical Imaging, Springer (2019)

[37] Chung, F.: Spectral Graph Theory. AMS (1997)

[38] Lombaert, H., Criminisi, A., Ayache, N.: Spectral forests: Learning of surface data, application to cortical parcellation. In: MICCAI. (2015)

[39] Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. JMLR (2006)

[40] Destrieux, C., Fischl, B., Dale, A., Halgren, E.: A sulcal depth parcellation of the cortex. NeuroImage (2009)

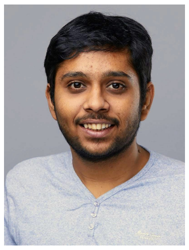
[41] Sowell, E.R., Thompson, P.M., Leonard, C.M., Welcome, S.E., Kan, E., Toga, A.W.: Longitudinal mapping of cortical thickness and brain growth in normal children. Journal of Neuroscience (2004)

[42] Lerch, J.P., Pruessner, J.C., Zijdenbos, A., Hampel, H., Teipel, S.J., Evans, A.C.: Focal decline of cortical thickness in alzheimer's disease identified by computational neuroanatomy. Cereb. cortex (2004)

[43] Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: ICML. (2010)

[44] Murphy, D.G., DeCarli, C., Mclntosh, A.R., Daly, E., Mentis, M.J., Pietrini, P., Szczepanik, J., Schapiro, M.B., Grady, C.L., Horwitz, B., et al.: Sex differences in human brain morphometry and metabolism: an in vivo quantitative magnetic resonance imaging and positron emission tomography study on the effect of aging. Archives of general psychiatry (1996)

[45] Ledig, C., Guerrero, R., Tong, T., Gray, K., Makropoulos, A., Heckemann, R., Rueckert, D.: Alzheimer's state classification using volumetry, thickness and intensity. In: MICCAI. (2014)



Karthik Gopinath received his BE from BNM Institute of technology, Bangalore, India and MSc from the International Institute of Information Technology Hyderabad, Hyderabad, India in Electronics and Communication engineering. As part of his MSc thesis, he proposed an automated Optical Coherence Tomography analysis system. He is now a PhD student in the Department of Software and IT Engineering of ÉTS Montreal. His main research topic is medical image analysis using geometric deep learning.

Christian Desrosiers received a Ph.D. in computer engineering from Polytechnique Montreal and was a post-doctoral researcher at the University of Minnesota on the topic of machine learning. In 2009, he joined the Department of Software and IT Engineering of ÉTS as professor. He is currently co-director of the Laboratoire d'Imagerie, de Vision et d'Intelligence Artificielle (LIVIA) of ÉTS, and is a member of the REPARTI research network. His main research interests focus on machine learning, image processing, computer vision and medical imaging.

Hervé Lombaert is Associate Professor of Computer Engineering at ETS Montreal, Canada, and holds a Canada Research Chair in Shape Analysis in Medical Imaging. He earned his PhD in 2012 and had the chance to work in multiple centers, including Microsoft Research (UK), Inria (France), McGill University (Canada), Siemens Corporate Research (USA), and the University of Montreal (Canada). His research focuses on the statistics and analysis of shapes in the context of medical imaging. His work on graph analysis has impacted the performance of several applications in medical imaging, from the early image segmentation techniques with graph cuts to recent surface analysis with spectral graph theory. His research has received several awards including the Erbsmann Prize.