Adaptive Graph Convolution Pooling for Brain Surface Analysis

Karthik Gopinath*, Christian Desrosiers, and Herve Lombaert

ETS Montreal, Canada

Abstract. Learning surface data is fundamental to neuroscience. Recent advances has enabled the use of graph convolution filters directly within neural network frameworks. These filters are, however, constrained to a single fixed-graph structure. A pooling strategy remains yet to be defined for learning graph-node data in non-predefined graph structures. This lack of flexibility in graph convolutional architectures currently limits applications on brain surfaces. Graph structures and number of mesh nodes, indeed, highly vary across brain geometries. This paper proposes a new general graph-based pooling method for processing full-sized surface-valued data, as input layers of graph neural networks, towards predicting subject-based variables, as output information. This novel method learns an intrinsic aggregation of input graph nodes based on the geometry of the input graph. This is leveraged using recent advances in spectral graph alignment where the surface parameterization becomes common across multiple brain geometries. These novel adaptive intrinsic pooling layers enable the exploration of entirely new architectures of graph neural networks, which were previously constrained to one single fixed structure in a dataset. We demonstrate the flexibility of the new pooling strategy in two proof-of-concept applications, namely, the classification of disease stages and regression of subject's ages using directly the surface data from varying mesh geometries.

1 Introduction

The analysis of brain surface data is essential for understanding the underlying mechanisms of cognition and perception. This surface has, however, a complex geometry. Its complex folding notably hinders current computational approaches for analyzing brain imaging data. Existing methods [1] are either volumetric or surface-based. On the one hand, volumetric approaches [2] operates over the whole brain, which is ideal for studying the brain fiber structure within the brain. The volumetric representation of imaging data, however, mostly ignore the geometry of the brain surface. Neighboring voxels in a volume may be quite far apart on the surface of a brain (see Fig. 1), posing a challenge for analyzing surface data. On the other hand, surface-based approaches [3,4] often over simplify the brain geometry to a sphere. Although topologically equivalent, the

* Corresponding author: K. Gopinath. **Email:** karthik.gopinath.1@etsmtl.net

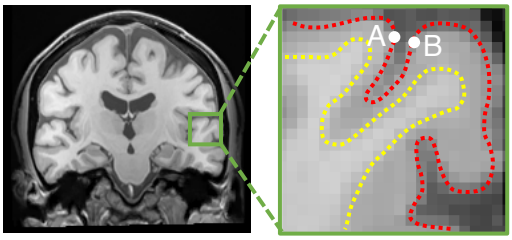

Fig. 1. Complex geometry of the cerebral cortex. As illustrated, two nearby points in the volume may in fact be far apart on the cortical surface.

important metric distortions, between a brain surface and a sphere, severely burden computations. For instance, FreeSurfer [3], a widely popular framework for brain analysis, requires several hours to inflate a brain into a sphere and parcellate its surface. Spherical harmonics approaches [5] also fundamentally rely on a spherical simplification of the brain.

Machine learning approaches has made recent breakthroughs in computer vision. In particular, convolutional neural networks [6] offer a computational advantage in terms of speed and accuracy over conventional approaches. They are, for instance, used in brain imaging for image segmentation [7,8]. However, they were limited to grid-structured data, such as images or volumes organized in a lattice. Recent advances [9,10,11,12] enable convolution operations over graphs by exploiting spectral analysis where convolutions translates into multiplications in a Fourier space. Convolutions are manipulated with eigenfunctions of graph Laplacian operators [13], approximated with Chebyshev [11] or Cayley polynomials [14]. These learned convolution filters are expressed in terms of mixtures of Gaussians [12] or splines [15]. These methods are, however, limited to a fixed graph structure, inadequate for brain imaging. Brain surfaces have, indeed, varying geometries with non-fixed degrees of nodes and edges across meshes. These variabilities pose a geometrical challenge [16] since the values of a Laplacian eigenfunction can drastically differ between brains with different surface geometries. To this effect, a learned synchronization [17] corrects for changes in eigenfunctions. An alignment of eigenbases [18] similarly provides a common parameterization of brain surfaces. Such aligned eigenbases enabled the direct learning of surface data across multiple brain geometries [19]. The architecture of these graph convolutional neural networks are, however, limited to use fixed sizes across hidden layers since pooling strategies remain to be defined on graph neural networks. Currently, heuristics are often used to mimic max pooling strategies [9,11,20]. They include varying the number of feature dimensions across layers [9] while retaining fixed sizes of layers, or relying on binary trees [11] or Graclus clustering methods [20] to coarsen the initial graph. They usually constrain choices of sizes in hidden layers or, more generally, architectural flexibility in pooling operations. This limits graph convolutional networks to fixed architectures or point-wise operations [12], such as node classification [21]. This

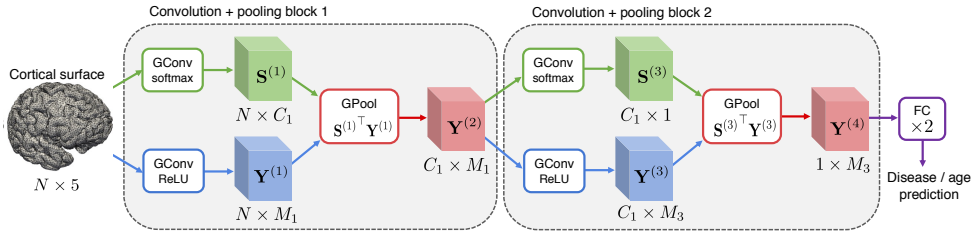

Fig. 2. An overview of the proposed graph convolutional network for subject-specific cortical surface analysis.

limitation prevents, for instance, whole subject classification of diseases under graphs of varying sizes.

This paper proposes a new geometric, adaptive pooling strategy for graph convolutional networks. Its flexibility enables arbitrary architectures in graph neural networks for extracting global subject-wise information from node-wise surface data in varying brain geometries. To do so, the novel pooling strategy learns node associations between layers from the surface data. The leverage of spectral node coordinates [18] enables a precise localization of prominent associations between hidden layers. This, for instance, contrasts with hierarchical approaches [22] where nodes lack intrinsic localization within a graph. The flexibility of the new pooling strategy is demonstrated in two proof-of-concept applications, with the classification of disease stages and the regression of subject ages directly from node-wise surface data. Both are shown to improve recent state-of-the-art on the ADNI dataset [23]. This is, to the best of our knowledge, the first application of graph convolutional networks with pooling layers for classifying or regressing subject-wise information from full-sized surface-valued data using varying brain geometries.

2 Method

We start by explaining how standard convolutions can be extended to non-rigid geometries such as surfaces. We describe next our end-to-end learnable pooling strategy which provides subject-specific aggregation of cortical features. Subsequently, we present how our proposed graph convolutional neural network operates with pooling layers to predict subject-based information such as stage of disease or brain age, directly from surface-valued data.

2.1 Geometric convolutions on surfaces

In a standard CNN, the input of the network is given as a set of features observed over a regular grid of points, such as pixels in 2D or voxels in 3D. The network processes information from input to output predictions with a cascade of convolutional layers, typically composed of a convolution operation followed

by a non-linear activation function, typically a sigmoid or ReLU. This can be formalized as follows. Let $\mathbf{Y}^{(l)} \in \mathbb{R}^{N \times M_l}$ 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 input feature map consists of N input nodes, each with M_l dimensions. Assuming a 1D grid for simplicity, the output feature map of layer l, convoluted with one kernel of size K_l, is given by $y_{ip}^{(l+1)} = f(z_{ip}^{(l)})$, where

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

In this formulation, $w_{pqk}^{(l)}$ are the convolution kernel weights; $b_p^{(l)}$, the bias weights of the layer; and f, the activation function.

In the case of 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 the concept of convolution to arbitrary graphs using the more general definition of geometric convolution [12,19,15]:

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

Here, φ_{ij} is a symmetric kernel parameterized by $\boldsymbol{\theta}_k$, relating the relative position of neighboring nodes j to nodes i when computing the convolutions. In [12], φ_{ij} is defined as a Gaussian kernel with learnable parameters $\boldsymbol{\theta}_k = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$ on the local polar coordinate $\mathbf{u}_{ij} = (\phi_{ij}, \theta_{ij})$ from node i to j:

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

Fig. 3 illustrates the relationship between conventional and geometrical convolutions. The standard convolution (left) can in fact be seen as a special case of a geometric convolution (right), for which nodes are placed on a regular grid and kernels are unit impulses placed at the grid position of neighbor nodes, effectively spherical Gaussian kernels with zero variance.

2.2 Extension to multiple complex surfaces

An important limitation of the geometric convolution model presented above is its inability to process surfaces which are aligned differently. 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. Additionally, as illustrated in Figure 1, geometric convolutions in a Euclidean space is not well suited for complex surfaces such as the highly folded brain,

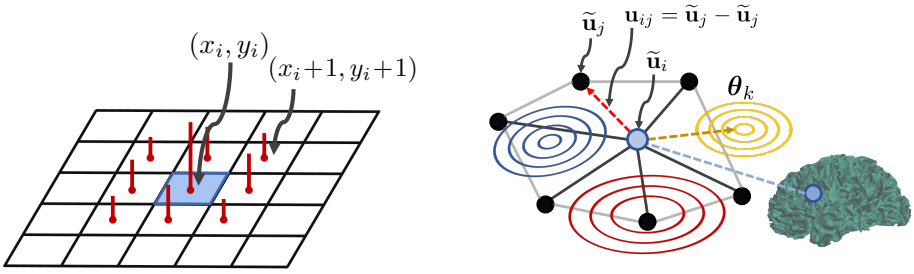

Fig. 3. Illustration of standard grid-based 2D convolutions (left) and geometric graph convolution (right). The challenge is to exploit kernels on arbitrary graph structures, and to add pooling operations over convolutional layers of graph nodes.

where nearby points in space may actually be far apart on the surface in terms of geodesic distance.

We address the issues of inter-surface alignment and intra-surface distance 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 $\mathbf{D}_{ii} = \sum_j \mathbf{D}_{ij}$. While 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. Denoting as $\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$ the eigen-decomposition of $\mathbf{L}$, we compute the normalized spectral coordinates of nodes as the rows of matrix $\widehat{\mathbf{U}} = \boldsymbol{\Lambda}^{-\frac{1}{2}}\mathbf{U}$. Because the most relevant characteristics of the embedded graph are captured by the principal spectral components of $\mathbf{L}$, as in [24], we limit the decomposition to the $d = 3$ first smallest non-zero eigenvalues of $\mathbf{L}$. The use of a low number of main spectral components is also computationally efficient.

Since the spectral embedding of $\mathbf{L}$ is only defined up to an orthogonal transformation, we must align spectral representations of different surface graphs to a common reference. Let $\widehat{\mathbf{U}}^{(0)}$ be the normalized spectral embedding of this reference. We align an embedding $\widehat{\mathbf{U}}$ to $\widehat{\mathbf{U}}^{(0)}$ with an iterative closest point (ICP) method [18]. In this method, each node $i \in \mathcal{V}$ is mapped to its nearest reference node $\pi(i) \in \mathcal{V}^{(0)}$ in the embedding space. The transformation $\mathbf{R}$ between corresponding nodes is found by approximating $\mathbf{R} = (\widehat{\mathbf{U}}^\top\widehat{\mathbf{U}})^{-1}(\widehat{\mathbf{U}}^\top\widehat{\mathbf{U}}^{(0)})$. Denote as $\widehat{\mathbf{u}}_i$ the normalized spectral coordinates of node i, the overall alignment process can be expressed as

$$\underset{\pi,\mathbf{R}}{\arg\min} \sum_{i=1}^{N} \left\| \widehat{\mathbf{u}}_i\,\mathbf{R} - \widehat{\mathbf{u}}^{(0)}_{\pi(i)} \right\|_2^2. \tag{4}$$

This optimization is solved by updating the node correspondence mapping π and the transformation matrix $\mathbf{R}$ as described above, until convergence [18].

We use the aligned spectral embedding $\widetilde{\mathbf{U}} = \widehat{\mathbf{U}}\,\mathbf{R}$ to define the local coordinates corresponding to an edge $(i, j) \in \mathcal{E}$: $\mathbf{u}_{ij} = \widetilde{\mathbf{u}}_j - \widetilde{\mathbf{u}}_i$. As on 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, *with respect to* the spectral coordinates of i.

2.3 Adaptive graph convolution pooling

In standard CNNs, pooling is typically carried out by aggregating values inside non-overlapping sub-regions of features maps. In graph convolutional networks [9,10,11,12], this approach is not applicable for several reasons. First, nodes are not laid out on a regular grid. This prevents aggregation of features in pre-defined regions. Second, the density of points may spatially vary in the embedding space. Pooling regions of fixed size or fixed shape are, therefore, not suitable for graphs with different geometries. Lastly, 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.

The limitations of traditional pooling techniques for graph convolutional networks can be addressed using different strategies. A first strategy is to aggregate features across all nodes in a *global* pooling step, typically after the last convolutional layer. A major problem with this strategy is the loss of all geometric and structural information during pooling. Another strategy, proposed by Wang et al. [25], performs a hierarchical clustering of nodes using their spectral coordinates, with a subsequent pooling of node features within each cluster. While this approach considers the graph structure, it is restricted by the chosen number of clusters. Furthermore, clusters are defined based only on node proximity in the embedding space, and the values to predict are ignored. Consequently, this unsupervised pooling strategy may not be optimal for the classification or regression task at hand.

In this work, we propose an end-to-end learnable pooling strategy for the subject-specific aggregation of cortical features. Inspired by the recently-proposed differential pooling technique of Ying et al. [22], this method 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 the features are aggregated. This two-path architecture is shown in Fig. 2. The feature encoding path is similar to a conventional CNN, and produces a sequence of convolutional feature maps $\{\mathbf{Y}^{(1)}, \dots, \mathbf{Y}^{(l)}\}$ with $\mathbf{Y}^{(l)} \in \mathbb{R}^{N \times M_l}$. The clustering path consists of sequential convolutional blocks, but replaces the activation function of the last block with a node-wise softmax. The output of this last block, $\mathbf{S} \in [0,1]^{N \times C}$, gives for each node i the probability s_{ic} that i belongs to cluster c. Pooled features $\mathbf{Y}^{\text{pool}} \in \mathbb{R}^{C \times M_l}$ are computed as the expected sum of convolutional features in each cluster:

$$y_{cp}^{\text{pool}} = \sum_{i=1}^{N} s_{ic} \cdot y_{ip}^{(l)}, \qquad \mathbf{Y}^{\text{pool}} = \mathbf{S}^{\top}\mathbf{Y}^{(l)} \tag{5}$$

At this stage, nodes are now replaced by clusters. The convolutions of node features, downstream the pooling operation, requires computing the adjacency

matrix of node clusters, $\mathbf{A}^{\text{pool}}$. The adjacency weights between pooling clusters c and d are defined as

$$a_{cd}^{\text{pool}} = \sum_{i=1}^{N}\sum_{j=1}^{N} s_{ic} \cdot s_{jd} \cdot a_{ij}, \quad \mathbf{A}^{\text{pool}} = \mathbf{S}^{\top}\mathbf{A}\mathbf{S}. \tag{6}$$

As described in [22], the bilinear formulation of Eq. (5) faces a challenging optimization problem with several local minima. To facilitate the learning process and obtain spatially smooth clusters, our approach adds the following regularization term:

$$\mathcal{L}_{\text{reg}}(\mathbf{S}) = \sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij} \cdot \|\mathbf{s}_i - \mathbf{s}_j\|^2 = \text{tr}(\mathbf{S}\mathbf{L}\mathbf{S}^{\top}), \tag{7}$$

where $\mathbf{s}_i$ denotes the cluster probability vector of node i.

2.4 Architecture details

The overall architecture of the proposed graph convolution network is shown in Fig. 2. As input to the network, we give the cortical surface features $\mathbf{x}_i$ and aligned spectral coordinates $\widetilde{\mathbf{u}}_i$ of each node i. Although various features could be considered to model the local geometry of the cortical surface [3], we used sulcal depth and cortical thickness, since the first one helps delineate anatomical brain regions [26] and the latter is related to ageing [27] and neurodegenerative diseases such as Alzheimer's [28].

The network is composed of two cascaded convolution-pooling 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. (5) 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 $1 \times$ nb.outputs vector with the second FC layer.

Except for the cluster probabilities and network output, all layers employ the Leaky ReLU as activation function: $y_{ip}^{(l)} = \max(0.01 z_{ip}^{(l)}, z_{ip}^{(l)})$. Moreover, for the graph convolution kernel φ_{ij} of Eq. (2), we used the B-spline kernels proposed by Fey et al. [15]. Compare to Gaussian kernels [12], this kernel has the advantage of making computation time independent from the kernel size.

For training, the loss function combines the output prediction loss and cluster regularization loss on the convolution-pooling block:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\text{out}}(\boldsymbol{\theta}) + \alpha \mathcal{L}_{\text{reg}}\big(\mathbf{S}^{(1)}(\boldsymbol{\theta})\big), \tag{8}$$

where α is a parameter controlling the amount of regularization. For classification tasks (i.e., disease prediction), $\mathcal{L}_{\text{out}}$ is set as the cross-entropy between one-hot encoded ground-truth labels and output class probabilities. In the case of

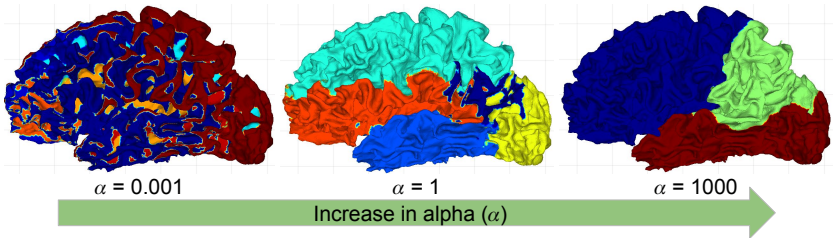

Fig. 4. Effect of α on clustering: The clusters learned with different strength of α. Smaller values of α produces multiple (10) spatially inconsistent clusters (left). Anatomically meaningful regions (middle) are learned with $\alpha = 1$ with 5 different clusters. $\alpha = 1000$ (right) results mimicking global pooling with 3 clusters. Nodes are color-coded to highlight the clustering.

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 a Nvidia Titan X GPU. The model takes less than a second for disease classification or age regression.

3 Results

We now validate our adaptive pooling approach. As a benchmark, we perform a disease classification and a brain age prediction using the ADNI dataset [29]. We use all available 731 brain surfaces, generated by FreeSurfer and manually labeled as normal cognition (NC), mild cognitive impairment (MCI), and Alzheimer's disease (AD). Each surface includes pointwise cortical thickness and sulcal depth. Meshes have a varying number of vertices and different triangulation. In a first experiment, we evaluate the influence of the regularization parameters α on the clustering of our learning framework. In a second experiment, we highlight the advantages of working in the spectral domain for disease classification (NC *vs* AD, MCI *vs* AD, and NC *vs* MCI). Finally, learning performance is measured when regressing the brain age operating directly in a spectral domain.

3.1 Effect of regularization on clustering

In practice, we use the Laplacian regularization $\mathcal{L}_{\text{reg}}$ within the loss function to avoid early spurious local minima when training for clustering. We have a hyper-parameter in our formulation α controlling regularization. We randomly split the ADNI dataset into a 70-10-20% ratio for training, validation and testing.

To evaluate the effect of α, we first set $C_1 = 16$ in the pooling path as a maximum number of possible clusters in a pooling layer. The goal of this experiment is to study how the spatial consistency of clusters varies with an increasing regularization. Fig. 4 shows the change in cluster assignments when increasing α in the NC *vs* AD classification task.

The spatial consistency of the clusters obtained with lower $\alpha = 0.0001$, enforcing a lower regularization, results in highly scattered regions, as seen on the left of Fig. 4. Equal weighting of $\mathcal{L}_{\text{out}}$ and $\mathcal{L}_{\text{reg}}$, with $\alpha = 1$, results in consistent clustered regions in the pooling layer. The middle of Fig. 4 shows clustered regions corresponding to prominent anatomical regions on the surface of the brain. A strong regularization, with $\alpha \geq 10$, results in higher level pooled regions, as seen on the right of Fig. 4. There are fewer clusters, with consistent regions that may relate with anatomical brain lobes.

3.2 Disease classification

Our method is now evaluated in a classification problem. To do so, we validate the performance of our algorithm on the ADNI dataset for NC *vs* AD, MCI *vs* AD and NC *vs* MCI binary classification problem, using all available 731 brain surfaces, generated by FreeSurfer. We compare our method with a random forest-based approach [30]. This baseline processes similar surface-based information, such as cortical thickness and sulcal depth, using the same dataset.

One of our contributions is to provide pooling operations in a geometry-aware domain. This is enabled by aligning spectral embeddings of brain surfaces across various mesh geometries. We first illustrate the current limitations of learning pooling operations without local spectral features. Models are trained models with only cortical thickness and sulcal depth. We also evaluate the improvement of our model performance by adding geometric information as spectral coordinates to the graph learning framework. We train three independents models to classify NC *vs* AD, MCI *vs* AD, or NC *vs* MCI. We use the same random split using the same architecture described earlier for each of our three models.

The performance on classification task is reported in Table 1. The accuracy for NC *vs* MCI is 76% without the use of geometric information. Our graph convolution network with spectral information indicates an accuracy of 89.33% for the same task. This is a 13.33% improvement. This gain in performance illustrates the advantage of using geometric information when processing surface data. Improvements are also observed when classifying MCI *vs* AD and NC *vs* MCI, with an increase of 2.89% and 6.20% respectively.

Table 1. Evaluation of the proposed work: Average accuracy of disease classification, in %, with standard deviation over the complete ADNI dataset. First row shows performance of a random forest with multiple cortical-based features [30]. Second row shows performance of our graph convolutional model without geometrical information (spectral). Last row indicates the results of our model with spectral shape information.

Input	**NC *vs* AD**	**MCI *vs* AD**	**NC *vs* MCI**
Random forest (Cortical-based) [30]	80 ± 5	65 ± 6	63 ± 4
Ours (Thick. + Depth)	76.00 ± 6.06	74.03 ± 8.63	63.71 ± 5.72
Ours (Spectral + Thick. + Depth)	89.33 ± 4.30	76.92 ± 4.78	70.79 ± 6.40

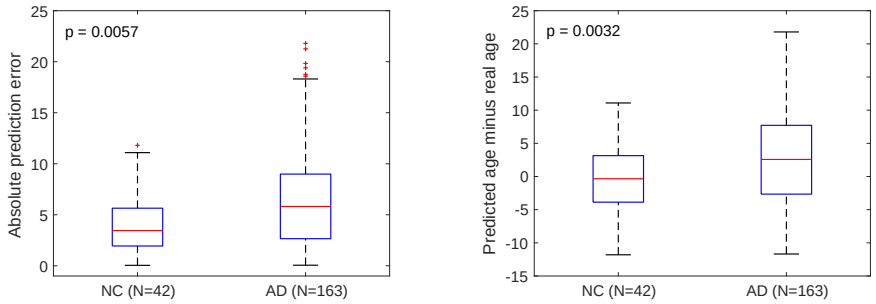

Fig. 5. Distribution of absolute prediction error (left) and predicted minus real age (right), for NC and AD test subjects. Our adaptive pooling strategy yielded graph models that could correctly capture age discrepancies between real and geometry-based ages, as expected between subjects with NC and AD.

3.3 Brain age prediction

In this experiment, our method is demonstrated in a regression problem where the brain age is predicted using pointwise surface-based measurements. We train our model on NC brains to regress brain ages. A model is learned with mean square error between real and predicted age with $\mathcal{L}_{\text{out}}$ as regularization. The graph convolution model uses cortical thickness, sulcal depth and spectral information as input. The model trained only on NC brain surfaces is then used to predict the real age of NC and AD subjects. Finally, we assess improvement due to the use of spectral coordinates, by comparing a model trained with and without geometric information.

The network architecture is illustrated in Fig. 2. The mean absolute error for the model predicting the real age on the NC subject is 4.35 ± 3.19 years. However, when the model is tested on the AD subjects, the prediction mean absolute error increases to 6.80 ± 6 years. The brain age calculated as the difference between prediction of our model and real age for NC and AD subjects indicate a statistically significance with a p-value of 0.0032. The real versus predicted ages over NC and AD is shown in Fig. 5.

4 Conclusion

We presented a novel strategy that enables pooling operations on graph convolutional networks of arbitrary graph structures. The ability to learn pooling patterns among graph nodes offers the possibility of exploring new graph-based neural network architectures. This new flexibility in designing network architectures is highly relevant for brain surface analysis. Subject-based prediction is, indeed, often drawn from surface-based values that resides on heterogeneous geometries.

Our experiments explore two different applications. In a first evaluation, the stage of Alzheimer's disease is learned from surface data, including cortical thick-

ness and sulcal depth. Our results show that point-wise surface values can be efficiently aggregated into a fixed number of class probabilities using a simple network architecture. The classification accuracy in recent state-of-the-art approaches exploiting directly *surface-based* features, such as cortical areas, thickness and other measurements [30], indicates that our graph pooling strategy provides an increase in accuracy from 63–80% to 70–89% on the ADNI dataset (Table 1). This is an 11% improvement. A closer evaluation also reveals that most of the performance gain occurs when spectral localization of graph nodes is used in the learning of pooling patterns. This indicates that node localization is essential to learn pooling strategies. Our method enables, therefore, on the one hand, new graph neural network architectures, via our proposed spectral pooling strategy, and on the other hand, a novel spatially varying learning of pooling patterns, via the spectral localization of probable graph patterns. In a second evaluation, the age of subjects are predicted using the geometry of their brains with point-wise surface data. Whole subject-based values, in this case, the subject's age, is regressed using our flexible pooling strategy. The architecture of the graph convolutional neural network can combine pooled layers of decreasing sizes, from full-sized cortical feature vectors to a single output for the predicted age. This experiment indicates that such new architectures can yield graph-based regressor of subject's characteristics directly from surface-based features lying on diverse brain geometries. The results shows that our graph networks could correctly capture the age discrepancies between the real age of a subject and its predicted geometry-based age. As expected, subjects with Alzheimer's have higher discrepancies than subjects with normal cognition (Fig. 5).

To summarize, our pooling strategy enables the exploration of a new family of architectures for graph convolutional neural networks. However, the proposed method depends on having datasets of comparable brain geometries. The spectral decomposition of graph Laplacian, indeed, assumes that shapes are topologically equivalent. Heterogeneity in holes and cuts in datasets of surfaces remains challenging to exploit since they may produce incompatible sets of Laplacian eigenvectors. This method is consequently inadequate for applications where major geometrical changes exist, such as when tumors are ablated. Nevertheless, our proposed pooling strategy remains highly relevant for a wide range of applications where surface data needs to be pooled sequentially in layers from full-size surface-valued vectors to single whole-subject characteristics.

Acknowledgment – This work is supported by the Research Council of Canada (NSERC), NVIDIA Corp. with the donation of a Titan Xp GPU. 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. Hua, X., et al.: Unbiased tensor-based morphometry: Improved robustness and sample size estimates for Alzheimer's disease clinical trials. NeuroImage (2013)

3. Fischl, B., et al.: Automatically parcellating the cortex. Cereb. Cortex (2004)
4. Yeo, B.T., Sabuncu, M.R., Vercauteren, T., Ayache, N., Fischl, B., Golland, P.: Spherical demons: Fast diffeomorphic surface registration. TMI (2010)
5. Styner, M., et al.: Framework for the statistical shape analysis of brain structures using SPHARM-PDM. Insight journal (2006)
6. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. ISP (1998)
7. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI. (2015)
8. Kamnitsas, K., et al.: Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. MedIA (2017)
9. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and locally connected networks on graphs. In: ICLR. (2014)
10. Kipf, T.N., Welling, M.: Semi-Supervised classification with graph convolutional networks. In: ICLR. (2017)
11. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: NIPS. (2016)
12. 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)
13. Xu, Y., Fan, T., Xu, M., Zeng, L., Qia, Y.: SpiderCNN: Deep learning on point sets with parameterized convolutional filters. In: ECCV. (2018)
14. Levie, R., Monti, F., Bresson, X., Bronstein, M.M.: CayleyNets: Graph convolutional neural networks with complex rational spectral filters. In: ICLR. (2018)
15. Fey, M., Lenssen, J.E., Weichert, F., Müller, H.: SplineCNN: Fast geometric deep learning with continuous B-Spline kernels. In: CVPR. (2018)
16. Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., Guibas, L.: Functional maps: A flexible representation of maps between shapes. In: SIGGRAPH. (2012)
17. Yi, L., Su, H., Guo, X., Guibas, L.J.: SyncSpecCNN: Synchronized spectral CNN for 3D shape segmentation. In: CVPR. (2017)
18. Lombaert, H., Arcaro, M., Ayache, N.: Brain transfer: Spectral analysis of cortical surfaces and functional maps. In: IPMI. (2015)
19. Gopinath, K., Desrosiers, C., Lombaert, H.: Graph convolutions on spectral embeddings: Learning of cortical surface data. In: arXiv preprint 1803.10336. (2018)
20. Dhillon, I.S., Guan, Y., Kulis, B.: Weighted graph cuts without eigenvectors a multilevel approach. PAMI (2007)
21. Parisot, S., et al.: Spectral Graph Convolutions for Population-Based Disease Prediction. In: MICCAI. (2017)
22. Ying, R., et al.: Hierarchical graph representation learning with differentiable pooling. In: arXiv 1806.08804. (2018)
23. Bron, E., et al.: The CADDementia challenge. Neuroimage (2015)
24. Lombaert, H., Criminisi, A., Ayache, N.: Spectral forests: Learning of surface data, application to cortical parcellation. In: MICCAI. (2015)
25. Wang, C., Samari, B., Siddiqi, K.: Local spectral graph convolution for point set feature learning. In: ECCV. (2018)
26. Destrieux, C., et al.: A sulcal depth parcellation of the cortex. NeuroImage (2009)
27. Sowell, E.R., et al.: Longitudinal mapping of cortical thickness and brain growth in normal children. Journal of Neuroscience (2004)
28. Lerch, J.P., et al.: Focal decline of cortical thickness in alzheimer's disease identified by computational neuroanatomy. Cereb. cortex (2004)
29. Jack, C.R., et al.: ADNI: MRI methods. JMRI (2008)
30. Ledig, C., et al.: Alzheimers state classification using volumetry, thickness and intensity. In: MICCAI. (2014)