Medical Image Analysis 54 (2019) 297–305

Contents lists available at ScienceDirect

Medical Image Analysis

journal homepage: www.elsevier.com/locate/media

Graph Convolutions on Spectral Embeddings for Cortical Surface Parcellation

Karthik Gopinath*, Christian Desrosiers, Herve Lombaert

ETS Montreal, Computer and Software Engineering, 1100 Notre Dame St. W., Montreal, QC H3C 1K3, Canada

ARTICLE INFO

Article history:
Received 31 July 2018
Revised 31 December 2018
Accepted 27 March 2019
Available online 30 March 2019

Keywords:
Graph convolution networks
Geometric deep learning
Spectral graph theory
Cortical parcellation

ABSTRACT

Neuronal cell bodies mostly reside in the cerebral cortex. The study of this thin and highly convoluted surface is essential for understanding how the brain works. The analysis of surface data is, however, challenging due to the high variability of the cortical geometry. This paper presents a novel approach for learning and exploiting surface data directly across *multiple* surface domains. Current approaches rely on geometrical simplifications, such as spherical inflations, a popular but costly process. For instance, the widely used FreeSurfer takes about 3 hours to parcellate brain surfaces on a standard machine. Direct learning of surface data via graph convolutions would provide a new family of fast algorithms for processing brain surfaces. However, the current limitation of existing state-of-the-art approaches is their inability to compare surface data across different surface domains. Surface bases are indeed incompatible between brain geometries. This paper leverages recent advances in spectral graph matching to transfer surface data across aligned spectral domains. This novel approach enables direct learning of surface data across compatible surface bases. It exploits spectral filters over intrinsic representations of surface neighborhoods. We illustrate the benefits of this approach with an application to brain parcellation. We validate the algorithm over 101 manually labeled brain surfaces. The results show a significant improvement in labeling accuracy over recent Euclidean approaches while gaining a drastic speed improvement over conventional methods.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Neuroimage analysis consists of studying functional and anatomical information over the brain geometry. Various aspects of the brain are investigated using different imaging modalities, such as magnetic resonance imaging (MRI) data. Structural MRI provides notably the geometry of the cortex. The thin outer layer of the brain cerebrum is of particular interest due to its vital role in cognition, vision, and perception. Statistical frameworks on surfaces are, therefore, highly sought for studying various aspects of the brain. For instance, variations in surface data can reveal new biomarkers as well as possible relations with disease processes (Arbabshirani et al., 2017). The challenge consists of learning surface data over highly complex and convoluted surfaces and across different subjects.

The goal of separating the cerebral cortex into distinct regions based on structure or function is known as parcellation. Initially, automated parcellation techniques used clustering based on local regional statistics (Craddock et al., 2012). For instance, a semi-supervised technique (Glasser et al., 2016) delineated the cortical boundary from sharp changes in multimodal MRI data. Most research works use a cortical surface based feature to find surface correspondence. BrainVisa (Cointepas et al., 2010; Rivière et al., 2003; Auzias et al., 2013) uses sulcal features defined by the cortical folding patterns to find correspondence between brain surfaces. Features like sulcal pits or sulcal lines (Lohmann et al., 2007; Auzias et al., 2015) are other existing features used for estimating surface correspondences. Conventional approaches typically rely on geometrical simplifications, such as spherical inflation and slow mesh deformations (Tustison et al., 2014; Styner et al., 2006; Yeo et al., 2010), a popular but costly process. For instance, the widely used FreeSurfer (Fischl et al., 2004) takes around 3 hours to parcellate brain surfaces by slowly deforming brain models towards labeled atlases.

Convolutional Neural Networks (CNNs) (Lecun et al., 1998) have the potential to offer a drastic speed advantage over traditional surface-based methods. CNNs are mostly used in neuroimage analysis for segmentation (Wachinger et al., 2017) or finding structural abnormalities (Valverde et al., 2017). The network architecture is either fixed for various segmentation applications (Ronneberger et al., 2015) or tailored to particular problems (Kamnitsas et al., 2017).

* Corresponding author.
E-mail address: karthik.gopinath.1@etsmtl.net (K. Gopinath).

https://doi.org/10.1016/j.media.2019.03.012
1361-8415/© 2019 Elsevier B.V. All rights reserved.

Fundamentally, current statistical frameworks exploit spatial information mostly derived from the Euclidean domain, for instance, based on vector fields, image or volumetric coordinates (Zhang and Davatzikos, 2011; Hua et al., 2013; Dolz et al., 2017; Kamnitsas et al., 2017). Such information is highly variable across brain geometries and severally hinders the training of modern machine learning algorithms.

Geometric deep learning (Bronstein et al., 2017) recently proposed to use convolutional filters on irregular graphs. To handle the neural network on a graph, Scarselli et al. (2009) proposes to map and learn graph data in a high-dimensional Euclidean space. Lately, Bruna et al. (2014) formulates the convolution theorem from Fourier space to spectral domains over graphs. Chebyshev polynomials are also used to avoid the explicit computation of graph Laplacian eigenvectors (Defferrard et al., 2016). The main concern of these methods is their inability to compare surface data across *different* surface domains (Bronstein et al., 2013; Kovnatsky et al., 2013; Ovsjanikov et al., 2012; Eynard et al., 2015). Laplacian eigenbases are indeed incompatible across multiple geometries. Alternatively, Masci et al. (2015) and Boscaini et al. (2016) proposed a graph convolution approach in the spatial domain. These approaches map local graph information onto geodesic patches and use conventional spatial convolution as template matching. For instance, Monti et al. (2017) obtains geodesic patches with local parametric constructions of tangent planes to the surface. Another prominent spatial approach Veličković et al. (2018) proposes to include self-attentional layers in which neighborhoods are used to avoid an explicit computation of a graph Laplacian. This attentional approach reduces to a particular formulation of Monti et al. (2017). A related work (Simonovsky and Komodakis, 2017) also conditions convolutional filter weights on specific edge labels over neighborhoods rather than on graph nodes. Applications of graph convolution networks in neuroimaging remain yet limited. Existing work includes the use of graph convolutions over population graphs for predicting brain disorders and learning distance metrics in functional brain networks (Parisot et al., 2018; Ktena et al., 2017). A recent work (Cucurull et al., 2018) proposes to parcellate the cerebral cortex into three parcels using an attention-based method (Veličković et al., 2018). Brain meshes are, however, constrained within a unique graph structure, limiting all meshes to use the same mesh geometry. Fundamentally, these methods either lack the capability to process multiple surface domains (Bronstein et al., 2013; Kovnatsky et al., 2013; Ovsjanikov et al., 2012; Eynard et al., 2015) or have spatial representations of surface data defined in a Euclidean space, which ignore the complex geometry of the surface. They rely, for instance, on polar representations of mesh vertices (Boscaini et al., 2016; Masci et al., 2015; Monti et al., 2017; Veličković et al., 2018).

This paper leverages recent advances in spectral graph matching to transfer surface data across aligned spectral domains (Lombaert et al., 2015a). The transfer of spectral coordinates across domains provides a robust formulation for spectral methods that naturally handles differences across Laplacian eigenvectors, including sign flips, ordering, and mixing of eigenvectors in higher frequencies. This spectral alignment strategy was exploited to learn surface data (Lombaert et al., 2015b) within the random forest framework. Spectral forests are operating in a spectral domain and use the first spectral coordinates as well as sulcal depth of each cortical point. This approach is, however, limited to only pointwise information, ignoring local patterns within surface neighborhoods. Our approach consists of leveraging spectral coordinates within graph convolutional networks to bridge a gap between learning algorithms and geometrical representations. To the best of our knowledge, this is the first attempt at intrinsically learning surface data via spectral graph convolutions in neuroimaging. This novel approach enables a *direct* learning of surface data across compatible surface bases by exploiting spectral filters over intrinsic representations of surface neighborhoods.

The main contributions of our work are:

- A novel *spectral* graph convolutional approach for cortical parcellation;
- A *direct* learning of surface data using trainable spectral filters over surface embeddings;
- The training of spectral filters across *multiple* mesh geometries of various graph structures;
- The leverage of spectral filters to exploit *local* patterns of data within surface neighborhoods;
- An evaluation on the *largest* publicly available dataset of manually labeled brain surfaces (Klein et al., 2017);
- An improved state-of-the-art performance for cortical surface parcellation with graph convolutions;

In this work, we propose a surface learning algorithm. We illustrate the learning capabilities of this approach with an application to brain parcellation. We choose cortical parcellation since it provides established benchmarks with publicly available datasets of manual labelings. The validation over the largest publicly available dataset of manually labeled brain surfaces (Klein et al., 2017), with 101 subjects, demonstrates a significant improvement in using spectral graph convolutions over Euclidean approaches. This change of paradigm indeed improves the parcellation accuracy when using graph convolutions, from a Dice score of 53% to 85%. Our approach is at least at par with the well established FreeSurfer algorithm (Fischl et al., 2004) when benchmarking over a large dataset (Klein et al., 2017), while gaining a drastic speed improvement in the order of seconds. The next section details the fundamentals of our spectral approach, followed by experiments evaluating the impact of our spectral strategy over standard Euclidean approaches for graph convolutions.

2. Method

An overview of the proposed method is shown in Fig. 1. In the first step, each cortical surface is modeled as a graph. Spectral decomposition is then applied on these graphs to capture the intrinsic geometry of brain surfaces and embed this information in a low-dimensional feature space (Lombaert et al., 2015a; Lefèvre et al., 2018). Subsequently, the transfer of surface data between spectral embeddings enables graph convolution networks to process cortical data across multiple mesh domains. This is implemented with a realignment of spectral embeddings. Finally, cortical parcellation is performed by learning spectral filters over realigned spectral coordinates and cortical features like the sulcal depth. Dense connections (Huang et al., 2017) improve convergence by propagating information from the initial layers to output layers. We therefore use dense connections among successive graph convolution layers. The overview of the network architecture is shown in Fig. 2.

2.1. Spectral embedding of brain graphs

Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ be a brain graph defined with node set $\mathcal{V}$, such that $N = |\mathcal{V}|$, and edge set $\mathcal{E}$. Each node i has a feature vector $\mathbf{v}_i = (\mathbf{x}_i, \mathbf{s}_i)$ composed of 3D spatial coordinates $\mathbf{x}_i$ and surface data features $\mathbf{s}_i$. Various features could be considered to model the local geometry of the cortical surface, including mean curvature, average convexity, and cortical thickness (Fischl et al., 2004; Li et al., 2015). In this work, we use sulcal depth since the boundaries of several regions in anatomical parcellation protocols typically follow such sulcal features (Destrieux et al., 2009).

We map $\mathcal{G}$ to a low-dimensional subspace using the eigencomponents of the normalized graph Laplacian operator

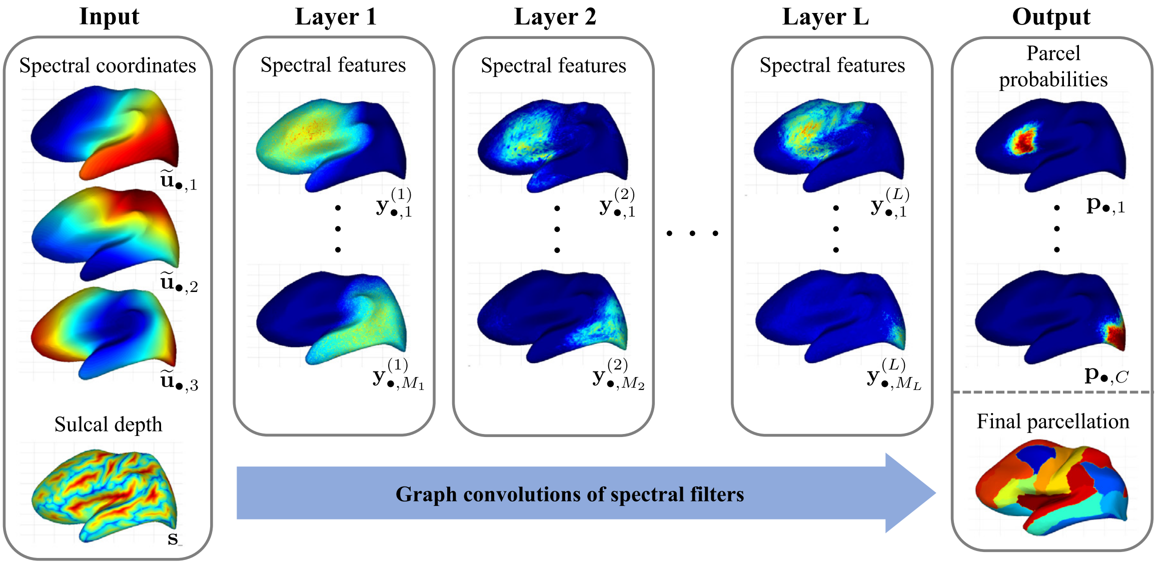

Input
Layer 1
Layer 2
Layer L
Output
Spectral coordinates
$\tilde{\mathbf{u}}_{\bullet,1}$
$\tilde{\mathbf{u}}_{\bullet,2}$
$\tilde{\mathbf{u}}_{\bullet,3}$
Sulcal depth
$\mathbf{s}$
Spectral features
$\mathbf{y}^{(1)}_{\bullet,1}$
$\mathbf{y}^{(1)}_{\bullet,M_1}$
Spectral features
$\mathbf{y}^{(2)}_{\bullet,1}$
$\mathbf{y}^{(2)}_{\bullet,M_2}$
Spectral features
$\mathbf{y}^{(L)}_{\bullet,1}$
$\mathbf{y}^{(L)}_{\bullet,M_L}$
Parcel probabilities
$\mathbf{p}_{\bullet,1}$
$\mathbf{p}_{\bullet,C}$
Final parcellation
Graph convolutions of spectral filters

Fig. 1. Overview of the algorithm – Graph convolutions of spectral filters are applied sequentially to process cortical surface data. On the left are inputs: Surface data, such as sulcal depth, *s*, and aligned spectral coordinates, $\tilde{\mathbf{u}}$. In the middle are the learned spectral features, **y**, found in each layer. On the right are: Predicted parcel probabilities, **p**, given by the softmax and the final surface parcellation. Coloring represents the pointwise value of respective maps from low (blue) to high (red) values. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

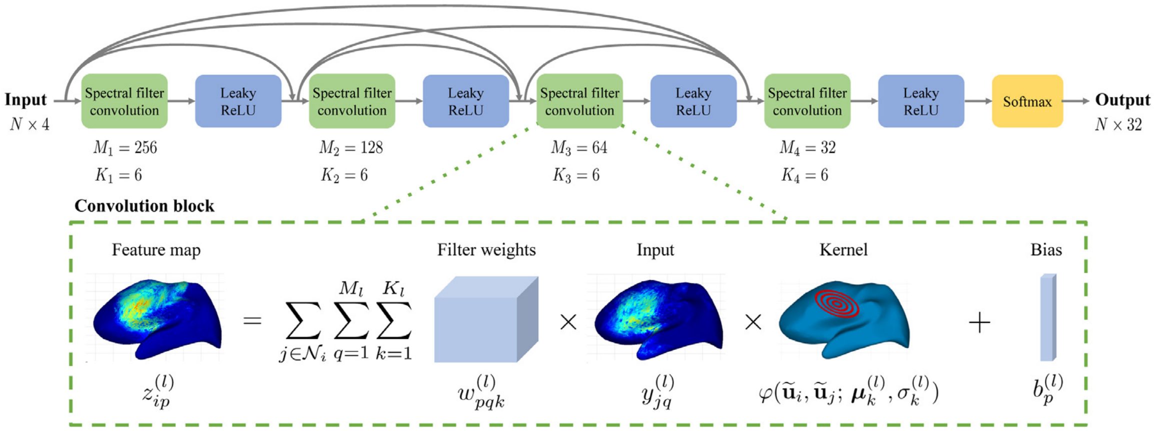

Input
$N \times 4$
Spectral filter convolution
Leaky ReLU
Spectral filter convolution
Leaky ReLU
Spectral filter convolution
Leaky ReLU
Spectral filter convolution
Leaky ReLU
Softmax
Output
$N \times 32$
$M_1 = 256$
$K_1 = 6$
$M_2 = 128$
$K_2 = 6$
$M_3 = 64$
$K_3 = 6$
$M_4 = 32$
$K_4 = 6$
Convolution block
Feature map
Filter weights
Input
Kernel
Bias
$z^{(l)}_{ip} = \sum_{j \in \mathcal{N}_i} \sum_{q=1}^{M_l} \sum_{k=1}^{K_l} w^{(l)}_{pqk} \times y^{(l)}_{jq} \times \varphi(\tilde{\mathbf{u}}_i, \tilde{\mathbf{u}}_j; \boldsymbol{\mu}^{(l)}_k, \sigma^{(l)}_k) + b^{(l)}_p$

Fig. 2. Overview of the network architecture – Dense connections are used among successive layers constituted with graph convolutions of learned spectral filters and leaky ReLU activations. Weights (w), biases (b), and parameters of our spectral filters (μ, σ) are learned via back-propagation. A final softmax function produces parcel probabilities (p) on the brain surface.

$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$, where **A** is the weighted adjacency matrix and **D** is the diagonal degree matrix (Chung, 1997). Here, we define the weight between two adjacent nodes in terms of node affinity (Grady and Polimeni, 2010), such as the inverse of their Euclidean distance: $a_{ij} = \left(\|\mathbf{x}_i - \mathbf{x}_j\|_2 + \epsilon\right)^{-1}$ where ϵ is a small constant to avoid a zero-division. Let $\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$ be the eigendecomposition of the normalized Laplacian matrix **L**. Since the most relevant characteristics of the embedded surface are captured by the principal spectral components of **L**, following Lombaert et al. (2015b), we limit the decomposition to the $d = 3$ first smallest non-zero eigenvalues of **L**. We then obtain the normalized spectral coordinates of nodes as the rows of matrix $\widehat{\mathbf{U}} = \boldsymbol{\Lambda}^{\frac{1}{2}}\mathbf{U}$.

Because the spectral embedding of **L** is only defined up to an orthogonal transformation, we need to align spectral representations of different brain graphs to an arbitrary reference. Denote as $\widehat{\mathbf{U}}^{(0)}$ the normalized spectral embedding of this reference, we align an embedding $\widehat{\mathbf{U}}$ to $\widehat{\mathbf{U}}^{(0)}$ with an iterative method based on the ICP algorithm (Lombaert et al., 2015a). 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 via a nearest neighbor search. The optimal transformation **R** between matched nodes is then obtained by solving a Procrustes analysis problem. Let $\widehat{\mathbf{u}}_i$ be the normalized spectral coordinates of node i, the overall alignment process can thus be formulated as:

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

We solve this problem by updating the node correspondence mapping π and the transformation **R** as described above, until convergence.

2.2. Graph convolution on surfaces

We start by presenting the standard CNN model for rigid grids and then explain how this model can be extended to an arbitrary geometry. Let $\mathbf{Y}^{(l)} \in \mathbb{R}^{N \times M_l}$ be the input feature map at convolution layer l of the network, such that $y^{(l)}_{iq}$ is the q-th feature of the i-th input node. The network input thus corresponds to $\mathbf{Y}^{(1)}$. Assuming a 1D grid, the output feature map of layer l is given by $y_{ip}^{(l+1)} = f(z_{ip}^{(l)})$ with:

$$z_{ip}^{(l)} = \sum_{q=1}^{M_l} \sum_{k=-K_l}^{K_l} w_{pqk}^{(l)} \cdot y_{i+k,q}^{(l)} + b_p^{(l)}. \tag{2}$$

Here, $w_{pqk}^{(l)}$ are the convolution kernel weights, $b_p^{(l)}$ the bias weights of the layer, and f is a non-linear activation function, for instance, the sigmoid or rectified linear unit (ReLU) functions.

To extend this fixed-grid formulation to a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, we denote as $\mathcal{N}_i = \{j \,|\, (i,j) \in \mathcal{E}\}$ the neighbors of node $i \in \mathcal{V}$. A generalized convolution operation can then be defined as:

$$z_{ip}^{(l)} = \sum_{j\in\mathcal{N}_i} \sum_{q=1}^{M_l} \sum_{k=1}^{K_l} w_{pqk}^{(l)} \cdot y_{jq}^{(l)} \cdot \varphi_{ij}(\boldsymbol{\Theta}_k^{(l)}) + b_p^{(l)}, \tag{3}$$

where $\varphi_{ij}(\boldsymbol{\Theta}_k)$ is a symmetric kernel with parameters $\boldsymbol{\Theta}_k$. In Monti et al. (2017), this kernel is defined on a tangent plane of a mesh at node i and is parameterized using polar coordinates. Learning is however constrained to a *single* graph structure, which hinders the application of convolutions across multiple graphs.

2.3. Learning across multiple mesh geometries

To learn surface data across multiple graphs, we leverage the spectral transfer of information across spectral embeddings. The transformation $\mathbf{R}$ of Eq. (1) is first used to obtain the aligned spectral coordinates with $\widetilde{\mathbf{U}} = \mathbf{R}\widehat{\mathbf{U}}$. Convolution kernels φ is then defined in the common spectral domain:

$$z_{ip}^{(l)} = \sum_{j\in\mathcal{N}_i} \sum_{q=1}^{M_l} \sum_{k=1}^{K_l} w_{pqk}^{(l)} \cdot y_{jq}^{(l)} \cdot \varphi(\widetilde{\mathbf{u}}_i, \widetilde{\mathbf{u}}_j;\ \boldsymbol{\Theta}_k^{(l)}) + b_p^{(l)}. \tag{4}$$

While any symmetric kernel can be used, in this work, we set φ as the Gaussian kernel with mean (or *offset*) $\boldsymbol{\mu}_k$ and variance (or *bandwidth*) σ_k:

$$\varphi(\widetilde{\mathbf{u}}_i, \widetilde{\mathbf{u}}_j;\ \boldsymbol{\mu}_k, \sigma_k) = \exp\left(- \|(\widetilde{\mathbf{u}}_j - \widetilde{\mathbf{u}}_i) - \boldsymbol{\mu}_k\|_2^2 \,/ 2\sigma_k^2 \right). \tag{5}$$

In an image domain, the neighborhood structure is regular, often organized in a lattice with fixed edge lengths between neighboring pixels. However, in a graph embedding, neighborhoods can have arbitrary structures with different edge lengths across the embedding. The continuous spectral domain embeds the geometric information of the graph. We define learnable kernels in a spectral domain relative to node i in order to capture the neighborhood information. This is thus the reason for subtracting $\widetilde{\mathbf{u}}_i$ in Eq. (5). Keeping kernels relative to node i allows the application of convolutions over a continuous space and reusing the same kernel parameters across different nodes. This fundamental change from Euclidean space to the spectral domain enforces the learning process to be geometry-aware.

The difference between the proposed graph convolution and standard convolutions over a fixed grid is illustrated in Fig. 3. Grid-based convolutions can be seen as a special case of our graph convolutions, for which the kernel offsets $\boldsymbol{\mu}_k$ are positioned regularly on the grid and the bandwidth $\sigma_k \to 0$. In addition to extending standard convolutions to irregular grids in continuous space, the proposed formulation can also model filter responses at different scales by varying σ_k. Moreover, since kernel parameters $\boldsymbol{\mu}_k$ and σ_k are learned directly from training data, instead of being defined during architecture design, our formulation can better adapt to the task at hand. For instance, kernels with small bandwidth can be learned to recognize thinner cortical structures, while large bandwidth kernels can be learned to model broader regions. Finally, the proposed strategy avoids the use of tangent planes, polar pseudo-coordinates (Monti et al., 2017) or the costly computation of geodesic distances (Masci et al., 2015; Boscaini et al., 2016).

Using the formulation of Eq. (4), we define a fully-convolutional network whose input at node i is given by $\mathbf{y}_i^{(0)} = (\widetilde{\mathbf{u}}_i, s_i)$, where $\widetilde{\mathbf{u}}_i$ is the aligned spectral coordinates of i and s_i is the sulcal depth at this node. The output layer of the network has a size corresponding to the number of parcels to be segmented, 32 in our case. Leaky ReLU is applied after each layer to obtain filter responses: $y_{ip}^{(l)} = \max(0.01 z_{ip}^{(l)}, z_{ip}^{(l)})$. Since the parcels to segment are mutually exclusive, we use a softmax operation after the last graph convolution layer to obtain the parcel probabilities of each node. The softmax function is given by $\frac{exp(y_{ip}^{(l)})}{\sum_q \exp(y_{iq}^{(l)})}$. Finally, the weighted cross-entropy is employed as output loss function:

$$E(\boldsymbol{\Theta}) = -\sum_{i=1}^{N} \sum_{c=1}^{C} \omega_c \cdot s_{ic} \cdot \log p_{ic}(\boldsymbol{\Theta}), \tag{6}$$

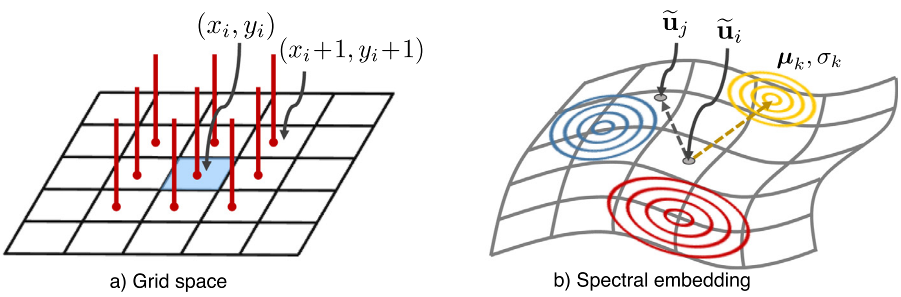
where $\boldsymbol{\Theta} = \{w_{pqk}^{(l)}, b_p^{(l)}, \boldsymbol{\Theta}_k^{(l)}\}$ are the trainable network parameters, $p_{ic}(\boldsymbol{\Theta})$ is the output probability for node i and parcel label c, and s_{ic} is a one-hot encoding of the reference segmentation. The weights ω_c compensate for the size difference of parcels, and are set inversely proportional to their surface areas such that smaller and larger parcels have similar importance. This loss is minimized by back-propagating the error using standard gradient descent optimization.

3. Results

We now validate our spectral learning approach. To do so, we benchmark our performance using the largest publicly available dataset of manually labeled brain surfaces, Mindboggle (Klein et al., 2017). It contains 101 subjects collected from different sites, with cortical meshes varying from 102K to 185K vertices. Each brain surface includes 32 manually labeled parcels. The experiments are carried out on an i7 desktop computer with 16GB of RAM and a Nvidia Titan X GPU. First, we evaluate the influence of different parameters in our learning framework. Second, we highlight the effect and advantages of spectral alignment in this framework. Finally, we assess the improvement in accuracy of learning frameworks when directly operating in a spectral domain rather than a conventional Euclidean domain.

3.1. Model selection

The hyper-parameters in our formulation are the number of graph convolution layers, L, in the fully-convolutional network and the number of Gaussian kernels, K, in each layer. To evaluate the effect of these parameters, we first set the size of the output layer to be equal to the number of parcels, 32 in our case. We measure the change in performance when increasing the number of layers from 1 to 4. Dense connections are removed in order to benchmark performance in a same controlled experimental setting. Along the layers of our architecture, the size of feature maps is reduced by two between two consecutive layers. This is to focus information towards the final parcellation and to limit memory usage. We also evaluate the performance when changing the number of kernels from 1 to 7. Our goal is to study how performance improves with an increasing number of layers and kernels. The total number of trainable parameters in a layer l that has a feature map of size M_l and K kernels is given by $K\left(M_{l-1} + M_{l-1} \times M_l\right) + M_l$. This indicates that the number of trainable parameters grows with the size of our network architecture. The model becomes computationally expensive in terms of memory usage for architecture beyond 4 layers and 7 kernels. Using relatively large sized brain meshes of


(x_i, y_i)
$(x_i\!+\!1, y_i\!+\!1)$
$\widetilde{\mathbf{u}}_j$
$\widetilde{\mathbf{u}}_i$
$\boldsymbol{\mu}_k, \sigma_k$

a) Grid space

b) Spectral embedding

Fig. 3. Illustration of the convolution operation – (a) Standard convolution on grids. (b) Geometric convolutions using three Gaussian kernels on a spectral embedding. Operations on grids can be shown to be a special case of operations on surfaces.

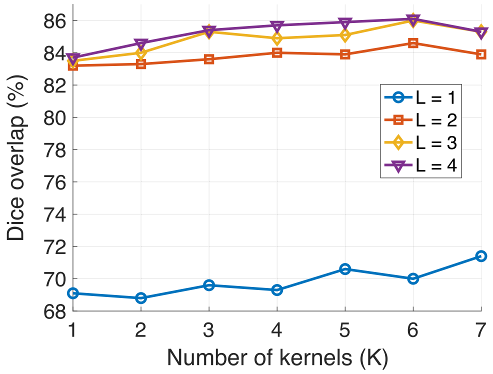


Fig. 4. Model selection - Each line indicates the segmentation accuracy in terms of Dice scores on the test split for different architectural models. It is observed that performance improves when the number of layers, *L*, increases, but quickly reaches a plateau. Performance also increases with the number of kernels, *K*. A peak is observed with 6 kernels and 4 layers. This configuration requires about 10GB of RAM. Increasing the model complexity would unnecessarily burden memory usage.

the Mindboggle dataset, we use random splits for training, validation, and testing in a 70–10–20% ratio. Each evaluated architecture is trained for 250 epochs on the train split. We observe an increase in segmentation accuracy in terms of Dice score when the number of kernels, *K*, increases. However, for the same number of kernels, adding more layers burdens the model complexity, while accuracy increases but stagnates from 2 to 4 layers. This is shown in Fig. 4.

This experiment indicates that $L = 4$ with $K = 6$ would be the optimal hyper-parameter values. The model has a total number of 264,768 trainable parameters and takes up to 10GB of GPU memory. The trainable parameters for our dense version doubles and takes up to 11GB of shared GPU memory. We similarly train our dense model for 250 epochs. The best performing model on the validation set is used for testing. Fig. 5 illustrates the evolution of the learning algorithm over iterations when classifying one parcel. We use this dense model for the rest of our experiments.

3.2. Spectral alignment

Our contribution is to operate in a geometry-aware spectral domain. This is enabled by aligning spectral embeddings across various mesh domains. We now evaluate the effect of a spectral alignment when learning graph convolution kernels. We align the spectral representations of different brain graphs to an arbitrary reference from the dataset.

First, we verify independence of our method with respect to the choice of a reference for alignment. To do so, we train and test our algorithm with 5 different reference brains, where the whole dataset is aligned to a reference. The evaluation of our algorithm on these 5 different references shows a performance having an average dice score of 86.4% and a standard deviation of 0.17% (Table 1). This indicates that our learning algorithm is robust to the choice of reference.

Second, we evaluate the impact of aligning spectral embeddings in learning graph convolution kernels. When both training and testing sets are aligned towards one same reference, the trained model yields an accuracy of 86.62% in terms of Dice overlap. However, when training and testing sets are both aligned towards differing references, $\text{Ref}_{\text{training}}$ and $\text{Ref}_{\text{testing}}$, the performance drops to 79.73%. This may be expected since both training and testing sets are expressed using differing spectral domains. To solve this discrepancy, our methodology consists of realigning the testing set towards one unique spectral domain, for instance, using $\text{Ref}_{\text{training}}$. In such case, the performance on realigned embeddings is improved to 84.7%. To evaluate the effect of varying references between training and testing sets, we iterate over all possible combinations, summarized in Fig. 6. It is observed that if spectral alignment is not present between the training and testing set, the classification accuracy degrades, while a spectral realignment of the testing set brings back the accuracy to initial scores.

3.3. Comparison with the state-of-the-art

We now compare our method with state-of-the-art approaches in learning graph-structured methods. First, we show the

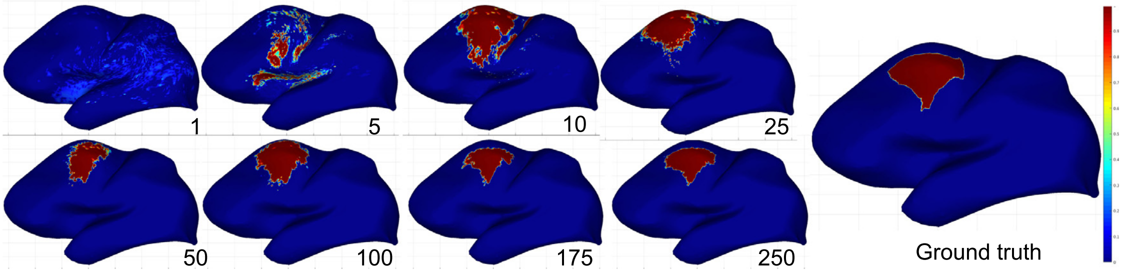
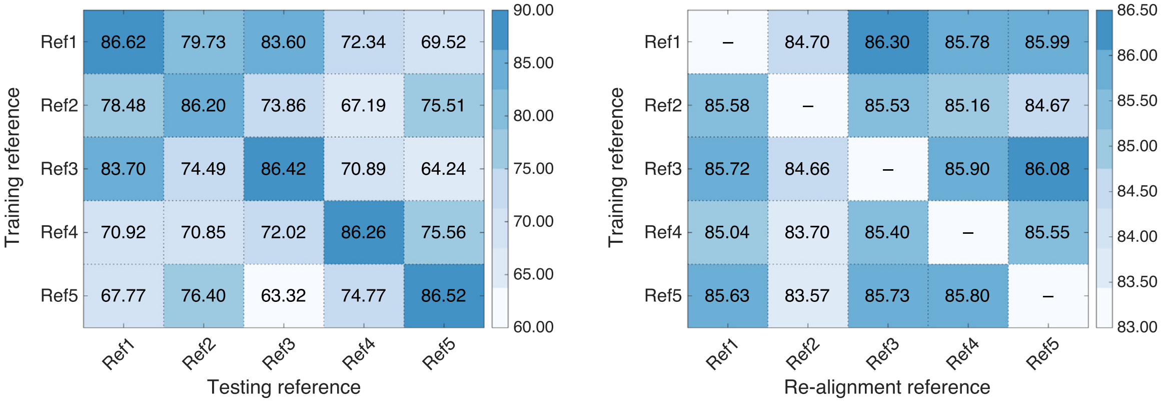


Fig. 5. Evolution of learning algorithm - The prediction of a particular parcel over multiple epochs is shown. A coarse to fine refinement of the parcel region is observed. After training, the predicted parcel probability corresponds to the ground truth parcel, shown on the right.

Table 1
Robustness to reference across all parcels – The average dice percentage obtained after separate training and testing with 5 references. The last column provides the mean and standard deviation of the results across all 35 parcels tested with all 5 references.

Ref_1	Ref_2	Ref_3	Ref_4	Ref_5	Mean
86.62% ± 1.72	86.20% ± 1.56	86.42% ± 1.73	86.26% ± 1.67	86.52% ± 1.75	**86.40%** ±**0.17**



	Ref1	Ref2	Ref3	Ref4	Ref5
Ref1	86.62	79.73	83.60	72.34	69.52
Ref2	78.48	86.20	73.86	67.19	75.51
Ref3	83.70	74.49	86.42	70.89	64.24
Ref4	70.92	70.85	72.02	86.26	75.56
Ref5	67.77	76.40	63.32	74.77	86.52

	Ref1	Ref2	Ref3	Ref4	Ref5
Ref1	–	84.70	86.30	85.78	85.99
Ref2	85.58	–	85.53	85.16	84.67
Ref3	85.72	84.66	–	85.90	86.08
Ref4	85.04	83.70	85.40	–	85.55
Ref5	85.63	83.57	85.73	85.80	–

Fig. 6. **Robustness to the choice of reference** – (Left) Dice score performance of two spectral models trained and tested using different aligned references. Row-column (*i*, *j*) provides the score of a using model trained via reference *i*, and tested via reference *j*. If references differ, scores degrade. This illustrates the current limitation of current graph convolutional approaches. (Right) Dice score performance of models when references are aligned. Row-column (*i*, *j*) indicates that difference references are used during trained and testing. The higher scores in off-diagonal experiments indicates that a realignment of spectral embeddings is essential to exploit multiple mesh domains.

limitations of working in the Euclidean domain with the Random Forest method (Lombaert et al., 2015b) as well as the latest approaches of graph convolution networks (Monti et al., 2017). Second, we show the advantage of changing the paradigm in graph learning frameworks from operating in a conventional Euclidean domain towards a spectral domain. This is enabled by our transfer of spectral embeddings across brain surface domains. Finally, we assess the improvement of exploiting neighborhoods of surface data versus the pointwise Spectral Forest method (Lombaert et al., 2015b), as well as a comparison with the established FreeSurfer algorithm (Fischl et al., 2004).

We train and test all methods on the entire dataset with a 5-fold cross-validation. We train a random forest with 50 trees on 3D spatial location and sulcal depth, which we name Euclidean Forest, as in Lombaert et al. (2015b). For comparison, we also train a graph convolution network similarly to Monti et al. (2017) in the Euclidean domain, with 3D spatial location and sulcal depth. The architecture remains the same as described earlier, as in Fig. 2. The average Dice overlap across all parcels in our dataset is 45.87% with Euclidean Forests. Graph convolution networks in the Euclidean space has an average Dice overlap of 50.78%, which is an increase of 4.9%.

To put in perspective, a pure spectral alignment of brain surfaces yields aligned parcellations that have an average Dice overlap of 77.67% over all pairs of possible brains. This pure alignment process does not involve learning of surface data while producing scores 16.9% superior to the previous Euclidean learning approaches. This indicates the benefits of operating in a spectral domain instead of the conventional Euclidean domain.

The effect of learning over spectral domains is assessed using, first, pointwise information in the Random Forest framework and in graph convolutional networks. Input for forests consists of sulcal depth and the first three spectral coordinates. This is referred to the Spectral Forest, similarly to Lombaert et al. (2015b), and yields an average Dice overlap of 79.89%. It is important to note that Spectral Forests learn over pointwise information only. Our graph convolutional network benefits from exploiting neighborhood information of surface data. The trained kernels on spectral embeddings yield an average Dice overlap of 85.37%. These results are summarized in Table 2. Fig. 7 shows that indeed learning using spectral method produces an improved parcellation quantitatively. The qualitative results from our algorithm look similar to the manual parcellation.

As an illustration of further refinement, we use a Markov Random Field (MRF) regularization for our method. We apply a standard graph cut algorithm (Boykov and Kolmogorov, 2004) with minus-log parcel probabilities as unary potentials and the Potts model for defining binary potentials. MRF regularization further improves the overall classification accuracy from 85.37% to 86.62.1%. The Spectral Forest parcellates the brain with an average Dice overlap of 79.8%. With an MRF as post-processing over the prediction of Spectral Forest, the Dice overlap improves to 83.7%, still lower than our approach without MRFs. The improvement of 3.9% in Dice overlap is seen with the use of neighborhood information from MRF. We see an increase of 1.3% in terms of Dice overlap when we use MRF with our approach. We can observe a similar improvement regarding average Hausdorff distances, with a reduction of distance error from 2.1 mm to 1.75 mm (Table 2). However, our graph convolution based approach has higher performance of 2.7% average Dice overlap over the Spectral Forest + MRF. A closer look at the performance scores for each parcel (Fig. 8) also reveals a general improvement when exploiting

Table 2
Comparisons with graph learning approaches – Average dice overlaps (in %) over 32 parcels of 101 subjects are shown along classification accuracy (in %), and average Hausdorff distances (in millimeters).

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

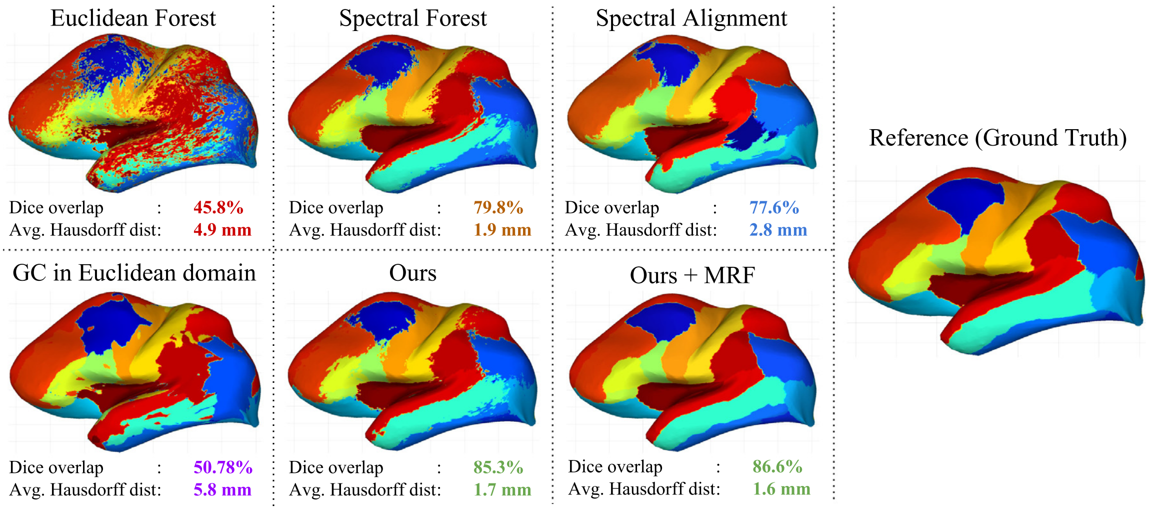


Fig. 7. Cortical parcellation – (*First column, Left*) Learning with Euclidean coordinates: yields low Dice score (45.8% with Random Forests, 50.8% with graph convolutions) and inconsistent boundaries (Hausdorff distance of 4.9–5.8 mm). (*Second column, Middle*) Learning with Spectral coordinates: improves Dice score (79.8% with Spectral Forests, 85.3% with our Spectral convolutions) and boundary regularity (1.9–1.7 mm). (*Third column, Top*) A pure spectral alignment without learning yields a Dice score of 77.6%. This is used as a benchmark to assess improvement in learning strategies. (*Third column, Bottom*) The parcel probability maps generated with our spectral filters could be further refined with an MRF regularization, leading to an improvement in boundary regularity (1.6 mm) and Dice score (86.6%). (*Right*) Reference ground truth for comparison purposes. Brain surfaces are inflated for visualization.

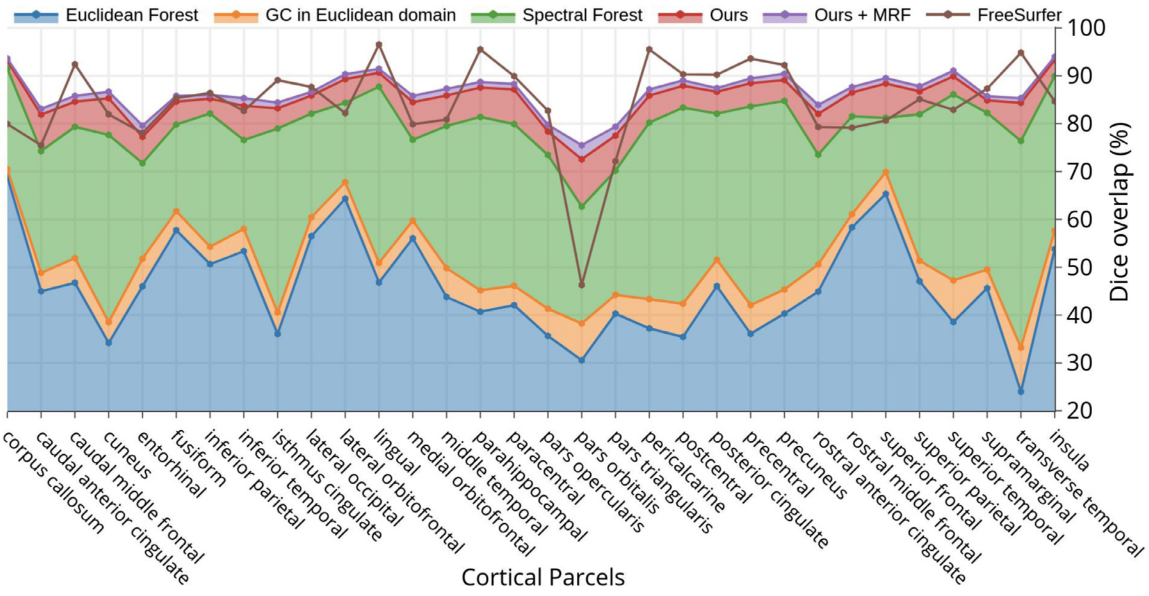


Fig. 8. Performance evaluation – Dice scores for all 32 cortical parcels across the dataset when learning with: (*Blue*) Euclidean Forest, (*Orange*) Graph convolutions in the Euclidean domain, (*Green*) Spectral Forest with pointwise information, (*Red/Ours*) Graph convolutions in a spectral domain, exploiting neighborhood information, (*Purple/Ours+MRF*) Final MRF refinement of our spectral maps, and (*Brown*) FreeSurfer provided for comparison. Improvements are consistent across all 32 parcels. The first leap in accuracy (*Orange area*, +11%) corresponds to an improvement from using convolutional networks over random forests. The second leap (*Green area*, +58%) corresponds to an improvement from learning in a spectral domain rather than Euclidean. The third leap (*Red area*, +7%) corresponds to the extra improvement of exploiting spectral neighborhoods when learning spectral convolutional filters. The fourth leap (*Purple area*, +1%) indicates the effect of regularizing final parcel probability maps with MRFs. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

neighborhoods (Our method) over pointwise surface data (Spectral Forest). This is a 34.59% improvement (Table 2) over learning in the standard Euclidean domain.

4. Discussion and conclusion

This paper presented a novel framework for learning surface data via graph convolutions of spectral filters. The algorithm leverages recent advances in spectral matching to enable the comparison of surface data across multiple surface domains. Our experiments illustrated the benefits of our approach with an application to cortical surface parcellation. This is a particularly challenging problem where current graph convolutional approaches remain limited by the inability to compare surface data across brain geometries. This typically results in spatial irregularities of parcel boundaries as illustrated in Fig. 7.

Shifting graph convolutions into a spectral domain endows the learning process with a geometry-aware representation of surface data. This strategy reveals that the use of spectral features improves a classification from a 50% Dice score in a conventional Euclidean domain to an 85% Dice score in a spectral domain. A performance gain is also noted when using a graph convolutional network instead of a standard random forest, from 45% to 50% when learning from a conventional representation of spatial information. Our experiments further indicate that an extra improvement is also gained by exploiting spectral neighborhoods. Fig. 8 indeed exhibits a major performance leap when leveraging spectral features over Euclidean features. This corresponds to the green area in Fig. 8, from 50% to 79% – a 29% improvement. The next leap in the graphic indicates an improvement due to exploiting spectral neighborhoods, from pointwise Spectral Forests (Lombaert et al., 2015b), where no neighborhood can be exploited with forests, to our graph convolutional approach exploiting spectral neighborhood. It is also worth noting that our graph convolutional approach uses only the first three principal components whereas the Spectral Forests (Lombaert et al., 2015b) considers five principal components to perform the same task. This is the red area in Fig. 8, from 79% to 85%, across all parcels in our dataset – an extra 7% improvement. These results confirm that exploiting a spectral domain provides a significant gain in performance, 29%, and exploiting convolutions over spectral neighborhoods provide an additional 7% improvement.

The experiments used the largest publicly available dataset of manually labeled brain surfaces (Klein et al., 2017). The performance of our spectral strategy is comparable to the state-of-the-art approaches for cortical parcellation. It reduces, however, the computation time by an order of magnitude. The Spectral Forest approach requires 23 seconds to run one pass over all 50 forests, yielding an accuracy of 79.89% ± 2.62%. Our method requires 3 seconds to run one forward-pass on a trained network, yielding an accuracy of 85.37% ± 2.36%. Both models roughly take 15 seconds to obtain the spectral coordinates and align to a reference brain. This is an 83% improvement in computation time. Using an additional MRF regularizer, as is often used in network approaches, brings up accuracy to 86.61% ± 2.45%. The parcellation process of FreeSurfer, starting from a brain surface, requires 3 hours of computation and yields an accuracy of 84.39% ± 1.91%. It is to be also noted that in the protocol of the Mindboggle dataset, annotations by experts are, in effect, manual corrections from FreeSurfer parcellations (Klein et al., 2017). This creates a positive bias for FreeSurfer results. Our claim in our experiments is not necessarily a superiority of our approach, but to rather provide a parcellation accuracy that is at least equivalent to FreeSurfer.

The advantage of using a spectral method is, on one hand, computational, by providing parcellation in seconds rather than hours, and on the other, methodological, by opening up a new learning strategy for processing cortical surface data. The technical contribution leveraged recent work on transfer of spectral bases across brain surface domains (Lombaert et al., 2015a,b). This enables the learning of spectral convolution filters across *multiple* brain geometries. This overcomes a major limitation in current graph convolutional approaches (Monti et al., 2017; Boscaini et al., 2016; Masci et al., 2015; Veličković et al., 2018), which are restrained to a unique fixed graph structure. Our method ameliorates graph spectral approaches by exploiting transfers of spectral bases. Furthermore, our experiments also used a multi-centric, multi-data and publicly available dataset. This provides an exhaustive, reproducible, evaluation for directly exploiting spectral features.

While the potential of our method is demonstrated on cortical parcellation, it can be applied to other analyses of surface data. For instance, our framework has a direct impact on regression problems that involve predictions of cortical thickness over time, potentially leading to new families of geometry-based biomarkers for neurological disorders.

Acknowledgments

This work was supported financially by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of a Titan X GPU used for this research.

References

Arbabshirani, M.R., Plis, S., Sui, J., Calhoun, V.D., 2017. Single subject prediction of brain disorders in neuroimaging: promises and pitfalls. NeuroImage 145, 137–165.

Auzias, G., Brun, L., Deruelle, C., Coulon, O., 2015. Deep sulcal landmarks: algorithmic and conceptual improvements in the definition and extraction of sulcal pits. Neuroimage 111, 12–25.

Auzias, G., Lefèvre, J., Le Troter, A., Fischer, C., Perrot, M., Régis, J., Coulon, O., 2013. Model-driven harmonic parameterization of the cortical surface: hip-hop. IEEE Trans. Med. Imaging 32 (5), 873–887.

Boscaini, D., Masci, J., Rodolà, E., Bronstein, M., 2016. Learning shape correspondence with anisotropic convolutional neural networks. In: NIPS, pp. 3189–3197.

Boykov, Y., Kolmogorov, V., 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Trans. PAMI 26 (9), 1124–1137.

Bronstein, M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P., 2017. Geometric deep learning: Going beyond Euclidean data. IEEE Trans. Signal Process. 34 (4), 18–42.

Bronstein, M., Glashoff, K., Loring, T., 2013. Making Laplacians Commute. 1307.6549.

Bruna, J., Zaremba, W., Szlam, A., Lecun, Y., 2014. Spectral networks and locally connected networks on graphs. In: ICLR, p. 1.

Chung, F., 1997. Spectral Graph Theory. AMS.

Cointepas, Y., Geffroy, D., Souedet, N., Denghien, I., Rivière, D., 2010. The BrainVISA Project: AShared Software Development Infrastructure for Biomedical Imaging Research, 16. HBM.

Craddock, R.C., James, G.A., Holtzheimer III, P.E., Hu, X.P., Mayberg, H.S., 2012. A whole brain fMRI atlas generated via spatially constrained spectral clustering. Human brain mapping 33 (8), 1914–1928.

Cucurull, G., Wagstyl, K., Casanova, A., Veličković, P., Jakobsen, E., Drozdzal, M., Romero, A., Evans, A., Bengio, Y., 2018. Convolutional neural networks for mesh-based parcellation of the cerebral cortex. In: MIDL, p. 1.

Defferrard, M., Bresson, X., Vandergheynst, P., 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In: NIPS, pp. 3844–3852.

Destrieux, C., Fischl, B., Dale, A., Halgren, E., 2009. A sulcal depth-based anatomical parcellation of the cerebral cortex. NeuroImage 47, 151–163.

Dolz, J., Desrosiers, C., Ben Ayed, I., 2017. 3D fully convolutional networks for subcortical segmentation in MRI: A large-scale study. NeuroImage 170, 456–470.

Eynard, D., Kovnatsky, A., Bronstein, M.M., Glashoff, K., Bronstein, A.M., 2015. Multimodal manifold analysis by simultaneous diagonalization of Laplacians. IEEE Trans. PAMI 12, 2505–2517.

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., 2004. Automatically parcellating the human cerebral cortex. Cereb.Cortex 14 (1), 11–22.

Glasser, M.F., Coalson, T.S., Robinson, E.C., Hacker, C.D., Harwell, J., Yacoub, E., Ugurbil, K., Andersson, J., Beckmann, C.F., Jenkinson, M., et al., 2016. A multi-modal parcellation of human cerebral cortex. Nature 536, 171–178.

Grady, L., Polimeni, J.R., 2010. Discrete Calculus: Applied Analysis on Graphs for Computational Science. Springer.

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, 2013. Unbiased tensor-based morphometry: improved robustness and sample size estimates for Alzheimer's disease clinical trials. NeuroImage 66, 648–661.

Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks. In: CVPR, 1, pp. 1–10.

Kamnitsas, K., Ledig, C., Newcombe, V.F.J., Simpson, J.P., Kane, A.D., Menon, D.K., Rueckert, D., Glocker, B., 2017. Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. Medical Image Analysis 36, 61–78.

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., 2017. Mindboggling morphometry of human brains. PLoS Comput. Biol. 13 (2), e1005350.

Kovnatsky, A., Bronstein, M.M., Bronstein, A.M., Glashoff, K., Kimmel, R., 2013. Coupled quasi-harmonic bases. In: Computer Graphics Forum, 32, pp. 439–448.

Ktena, S.I., Parisot, S., Ferrante, E., Rajchl, M., Lee, M., Glocker, B., Rueckert, D., 2017. Distance metric learning using graph convolutional networks: application to functional brain networks. In: MICCAI, 10433, pp. 469–477.

Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. IEEE Intell. Signal Process. 86 (11), 2278–2324.

Lefèvre, J., Pepe, A., Muscato, J., De Guio, F., Girard, N., Auzias, G., Germanaud, D., 2018. SPANOL (SPectral ANalysis Of lobes): a spectral clustering framework for individual and group parcellation of cortical surfaces in lobes. Front. Neurosci. 12, 354–366.

Li, G., Wang, L., Shi, F., Gilmore, J.H., Lin, W., Shen, D., 2015. Construction of 4D high-definition cortical surface atlases of infants: methods and applications. Medical Image Analysis 25 (1), 22–36.

Lohmann, G., Von Cramon, D.Y., Colchester, A.C., 2007. Deep sulcal landmarks provide an organizing framework for human cortical folding. Cereb. Cortex 18 (6), 1415–1420.

Lombaert, H., Arcaro, M., Ayache, N., 2015a. Brain transfer: spectral analysis of cortical surfaces and functional maps. In: IPMI, 24, pp. 474–487.

Lombaert, H., Criminisi, A., Ayache, N., 2015b. Spectral forests: Learning of surface data, application to cortical parcellation. In: MICCAI, 9349, pp. 547–555.

Masci, J., Boscaini, D., Bronstein, M., Vandergheynst, P., 2015. Geodesic convolutional neural networks on Riemannian manifolds. In: ICCV-3dRR, pp. 37–45.

Monti, F., Boscaini, D., Masci, J., Rodolà, E., Svoboda, J., Bronstein, M.M., 2017. Geometric deep learning on graphs and manifolds using mixture model CNNs. In: CVPR, 1, pp. 1–10.

Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., Guibas, L., 2012. Functional maps: a flexible representation of maps between shapes. In: SIGGRAPH, 31, p. 30.

Parisot, S., Ktena, S.I., Ferrante, E., Lee, M., Guerrero, R., Glocker, B., Rueckert, D., 2018. Disease prediction using graph convolutional networks: Application to autism spectrum disorder and Alzheimerâs disease. Medical Image Analysis 48 (1), 117–130.

Rivière, D., Régis, J., Cointepas, Y., Papadopoulos-Orfanos, D., Cachia, A., Mangin, J.F., 2003. A freely available Anatomist-BrainVISA package for structural morphometry of the cortical sulci. Neuro Image 19 (2), 934.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: MICCAI, 9351, pp. 234–241.

Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G., 2009. The graph neural network model. IEEE Trans. Neural Netw. 20 (1), 61–80.

Simonovsky, M., Komodakis, N., 2017. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In: CVPR, 1, pp. 29–38.

Styner, M., Oguz, I., Xu, S., Brechbühler, C., Pantazis, D., Levitt, J.J., Shenton, M.E., Gerig, G., 2006. Framework for the statistical shape analysis of brain structures using SPHARM-PDM. Insight J. 1071, 242–250.

Tustison, N.J., Cook, P.A., Klein, A., Song, G., Das, S.R., Duda, J.T., Kandel, B.M., van Strien, N., Stone, J.R., Gee, J.C., Avants, B.B., 2014. Large-scale evaluation of ANTs and FreeSurfer cortical thickness measurements. NeuroImage 99, 166–179.

Valverde, S., Cabezas, M., Roura, E., González-Villà, S., Pareto, D., Vilanova, J.C., Ramió-Torrentà, L., Rovira, À., Oliver, A., Lladó, X., 2017. Improving automated multiple sclerosis lesion segmentation with a cascaded 3D convolutional neural network approach. NeuroImage 155, 159–168.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y., 2018. Graph attention networks. In: ICLR, p. 1.

Wachinger, C., Reuter, M., Klein, T., 2017. DeepNAT: Deep convolutional neural network for segmenting neuroanatomy. NeuroImage 170, 434–445.

Yeo, B.T., Sabuncu, M.R., Vercauteren, T., Ayache, N., Fischl, B., Golland, P., 2010. Spherical demons: Fast diffeomorphic landmark-free surface registration. IEEE Trans. Med. Imaging 29 (3), 650–668.

Zhang, T., Davatzikos, C., 2011. ODVBA: Optimally-discriminative voxel-based analysis. IEEE Trans. Med. Imaging 30 (8), 1441–1454.