Graph Domain Adaptation for Alignment-Invariant Brain Surface Segmentation

Karthik Gopinath, Christian Desrosiers, and Herve Lombaert

ETS Montreal, Canada

Abstract. The varying cortical geometry of the brain creates numerous challenges for its analysis. Recent developments have enabled learning cortical data directly across multiple brain surfaces via graph convolutions. However, current graph learning algorithms fail when brain surface data are misaligned across subjects, thereby requiring to apply a costly alignment procedure in pre-processing. Adversarial training is widely used for unsupervised domain adaptation to improve segmentation performance on target data whose distribution differs from the training source data. In this paper, we exploit this technique to learn surface data across inconsistent graph alignments. This novel approach comprises a segmentator that uses graph convolution layers to enable parcellation across brain surfaces of varying geometry, and a discriminator that predicts the alignment-domain of surfaces from their segmentation. By trying to fool the discriminator, the adversarial training learns an alignment-invariant representation which yields consistent parcellations for differently-aligned surfaces. Using manually-labeled brain surface from MindBoggle, the largest publicly available dataset of this kind, we demonstrate a 2%–13% improvement in mean Dice over a non-adversarial training strategy, for test brain surfaces with no alignment or aligned on a different reference than source examples.

1 Introduction

The cerebral cortex is essential to a wide range of cognitive functions. Automated algorithms for brain surface analysis thus play an important role in understanding the structure and working of this complex organ. Nowadays, deep learning models such as convolutional neural networks (CNNs) provide state-of-the-art performance for most image analysis tasks, including image classification, registration, and segmentation [1]. However, these models typically require large annotated datasets for training, which are often expensive to obtain in medical applications. This limitation is especially true for the task of cortical segmentation, also known as *parcellation*, where generating ground truth data requires labeling possibly thousands of nodes on a highly-convoluted surface. This burden also explains why datasets for such tasks are relatively small. For instance, the largest publicly-available dataset for cortical parcellation, MindBoggle [2], contains only 101 manually-annotated brain surfaces. Moreover, another common problem of deep learning models is their lack of robustness to differences in the

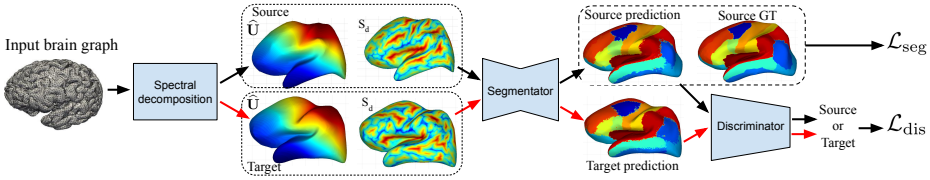

Fig. 1. Overview of our architecture: The input brain graph is mapped to a spectral domain by decomposition of the graph Laplacian. The source and target domain are obtained by aligning the eigenbases to source reference and targets reference respectively. A segmentator GCN learns to predict a generic cortical parcel label for each domain. The discriminator aims at classifying the segmentator predictions, thereby assisting the segmentator GCN in adapting to both source and target domains.

distribution of training and test data. Hence, a CNN model trained on the data from a source domain usually fails to generalize to samples from other domains, i.e., the *target* domains.

Unsupervised domain adaptation (UDA) [3] has proven to be a powerful approach for making algorithms trained on source data generalize to examples from a target domain, without having explicit labels for these examples. Generative adversarial networks (GANs) [4] leverage adversarial training to produce realistic images. In this type of approach, a discriminator network classifies images produced by a generator network as real or fake, and the generator improves by learning to fool the discriminator. Following the success of GANs, adversarial techniques have later been proposed to improve the learning capability of CNNs across different domains. Adversarial domain adaptation methods for segmentation [5,6,7,8,9,10] involve the concurrent training of two networks: a segmentator that learns to produce accurate segmentation outputs for labeled source examples, and a discriminator which forces the segmentator to have a similar prediction for examples of both source and target domains. These adversarial techniques usually rely on either feature space adaptation or output space adaptation. Initial works [11,12] focused on matching the distributions of features from source and target domain examples for classification tasks. As the output of CNNs for segmentation contains rich semantic information, [13] proposed a method that instead leverages output space adaptation. Various pixel-wise domain adaptation approaches have been developed for natural color images [12,14]. In medical image analysis, [15] proposed an adversarial neural network for MRI image segmentation which does not require additional labels on test examples from the target domain. Likewise, [10] presented a vessel segmentation approach for fundus images, which uses a gradient reversal layer for adversarial training. Recent work [16] also addressed the problem of domain adaptation by adding a differentiable penalty on the target domain. However, these domain adaptation techniques focus on data lying in the Euclidean space (natural or medical images) and, therefore, are not suitable for graph structures such as surface meshes.

The image space is inadequate to capture the varying geometry of the cerebral cortex. Differences in brain surface geometry hinder statistical frameworks from exploiting spatial information in Euclidean space. The extension of standard convolutions to non-Euclidean spaces like manifolds and graphs has led to the development of various geometric deep learning frameworks [17,18]. A recent work [19] proposed to use geometric deep learning for segmenting three cortical regions by relying on the spatial representation of the brain surface mesh. Later, based on the spectral representation of such meshes, [20] developed a graph convolution network (GCN) to parcellate the cerebral cortex. Despite offering more flexibility than Euclidean-based approaches, these methods are domain-dependent and would fail to generalize to new datasets (domains) without explicit re-training. Moreover, obtaining annotations for these new datasets is also challenging and time-consuming, due to the complexity of visualizing and labeling intricate surfaces.

In this paper, we address the limitations of existing techniques for cortical parcellation by proposing an adversarial domain adaptation method on surface graphs. Specifically, we focus on a problem shared by most GCN-based approaches, which is the need for a common basis to represent and operate on graphs. For approaches operating in Euclidean space, bringing surface graphs to this common basis usually involves transforming and possibly sub-sampling meshes to match a given reference, which is particularly difficult for convoluted surfaces like the cortex. As described in [20], this process can be greatly simplified by instead operating in the spectral domain, for instance using spectral GCNs [21,22]. Nevertheless, spectral GNCs also need to perform some alignment to work. Hence, these models require computing the eigendecomposition of the graph Laplacian matrix to embed graphs in a space defined by a fixed eigenbasis. However, separate graphs may have different eigenbases, and the eigenvectors obtained for a given graph are only defined up to a sign and a rotation (if different eigenvectors share close eigenvalues). Due to these ambiguities, spectral GCNs cannot be used to compare multiple graphs directly and need an explicit alignment of graph eigenbases as an additional pre-processing step. Here, we focus on generalizing parcellation across multiple brain surface domains by removing the dependency on these domain-specific alignments.

The contributions of our work are multifold:

- We present, to the best of our knowledge, the first adversarial graph domain adaptation method for surface segmentation. Our novel method trains two networks in an adversarial manner, a fully-convolutional GCN segmentator and a GCN domain discriminator, both of which operate on the spectral components of surface graphs.
- Compared to existing approaches, our surface segmentation method offers greater robustness to differences in domain-specific alignment. Hence, our method yields a higher accuracy for non-aligned brain surfaces compared to a strategy without adversarial learning. Moreover, it also provides a better generalization for surfaces aligned to a different reference, without requiring an explicit re-alignment or manual annotations of these surfaces.

– We demonstrate the potential of our method for alignment-invariant parcellation of brain surfaces, using data from MindBoggle, the largest publicly-available manually-labeled surface dataset. Our results show significant mean Dice improvements compared to the same segmentation network without adversarial training and over a strong baseline approach based on Spectral Random Forest.

In the next section, we detail the fundamentals of our graph domain adaptation method for surface segmentation, followed by experiments validating the advantages of our method and a discussion of results.

2 Method

An overview of our proposed method is shown in Fig. 1. In the initial step, the cortical brain graph is embedded into the spectral domain using the graph Laplacian operator. Next, samples from the source domain only are aligned to a reference template using the Iterative Closest Point (ICP) algorithm. This algorithm works by repeating the following two steps until convergence: 1) mapping each node of the graph to align to its nearest reference node in the embedding space; 2) computing the orthogonal transformation (i.e., rotation and flip) which brings nodes nearest to their corresponding reference node. Since this process is iterative and external to the network architecture, it can be computationally expensive to run. However, we only need to apply it during training and, as shown in experiments, the proposed method can achieve good performance on non-aligned test examples by learning an alignment-invariant representation. Finally, a graph domain adaptation network is trained to perform alignment-independent parcellation. The segmentator network learns a generic mapping from input surface features, e.g. the spectral coordinates and sulcal depth of cortical points, to cortical parcel labels.

2.1 Spectral embedding of brain graphs

We start by describing the spectral graph convolution model used in this work. Denote as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ a brain surface graph with node set $\mathcal{V}$, such that $|\mathcal{V}| = N$, and edge set $\mathcal{E}$. Each node i has a feature vector $\mathbf{x}_i \in \mathbb{R}^3$ representing its 3D coordinates. We map $\mathcal{G}$ to a low-dimension manifold using the normalized graph Laplacian operator $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$, where $\mathbf{A}$ is the weighted adjacency matrix and $\mathbf{D}$ the diagonal degree matrix. Here, we consider weighted edges and measure the weight between two adjacent nodes as the inverse of their Euclidean distance, i.e. $a_{ij} = (\|\mathbf{x}_i - \mathbf{x}_j\| + \epsilon)^{-1}$ where ϵ is a small positive constant. Letting $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ be the eigendecomposition of $\mathbf{L}$, the normalized spectral coordinates of nodes are given by $\widehat{\mathbf{U}} = \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}$. The normalization with $\mathbf{\Lambda}^{-\frac{1}{2}}$ is used so that coordinates corresponding to smaller eigenvalues are given more importance in the embedding.

Denote the neighbors of node $i \in \mathcal{V}$ as $\mathcal{N}_i = \{j \,|\, (i,j) \in \mathcal{E}\}$. The convolution operation used in our spectral GCN is defined as

$$\begin{aligned} z_{ip}^{(l)} &= \sum_{j \in \mathcal{N}_i} \sum_{q=1}^{M_l} \sum_{k=1}^{K_l} w_{pqk}^{(l)}\, y_{jq}^{(l)}\, \varphi(\widehat{\mathbf{u}}_i, \widehat{\mathbf{u}}_j;\, \Theta_k^{(l)}) \;+\; b_p^{(l)}, \\ y_{ip}^{(l+1)} &= \sigma\big(z_{ip}^{(l)}\big) \end{aligned} \tag{1}$$

where $y_{jq}^{(l)}$ is the feature of node j in the q-th feature map of layer l, $w_{pqk}^{(l)}$ is the weight in the k-th convolution filter between feature maps q and p of subsequent layers, $b_p^{(l)}$ is the bias of feature map p at layer l, and σ is a non-linear activation function. The information of the spectral embedding relating nodes i and j is included via a symmetric kernel $\varphi(\widehat{\mathbf{u}}_i, \widehat{\mathbf{u}}_j; \Theta_k)$ parameterized by Θ_k. In this work, we follow [20] and use a Gaussian kernel: $\varphi(\widehat{\mathbf{u}}_i, \widehat{\mathbf{u}}_j; \boldsymbol{\mu}_k, \sigma_k) = \exp\big(-\sigma_k \,\|(\widehat{\mathbf{u}}_j - \widehat{\mathbf{u}}_i) - \boldsymbol{\mu}_k\|^2\big)$.

2.2 Graph domain adaptation

Our graph domain adaptation architecture contains two blocks: a segmentator GCN S performing cortical parcellation and a discriminator GCN D, which predicts if a given parcellation comes from a source or target graph. Let $\mathcal{X}_{\text{src}}$ be the set of source graphs and $\mathcal{X}_{\text{tgt}}$ the set of unlabeled domain graphs, with $\mathcal{X} = \mathcal{X}_{\text{src}} \cup \mathcal{X}_{\text{tgt}}$ the entire set of graphs available in training. In the first step, we optimize the segmentator GCN using labeled source graphs $\mathcal{G} \in \mathcal{X}_{\text{src}}$. We feed the segmentation network's prediction $S(\mathcal{G})$ to the discriminator D whose role is to identify the input's domain (i.e., source or target). The gradients computed from an adversarial loss on target domain graphs are back-propagated from D to S, forcing the segmentation to be similar for both the source and target domain graphs.

As in other adversarial approaches, we define the learning task as a minimax problem between the segmentator and discriminator networks,

$$\max_D \min_S \; \mathcal{L}(D,S) = \frac{1}{|\mathcal{X}_{\text{src}}|} \sum_{\mathcal{G} \in \mathcal{X}_{\text{src}}} \mathcal{L}_{\text{seg}}(S(\mathcal{G}), \mathbf{y}_{\mathcal{G}}) \;-\; \frac{\lambda}{|\mathcal{X}|} \sum_{\mathcal{G} \in \mathcal{X}} \mathcal{L}_{\text{dis}}\big(D(S(\mathcal{G})), z_{\mathcal{G}}\big), \tag{2}$$

where $\mathcal{L}_{\text{seg}}$ is the supervised segmentation loss on labeled source graphs, and $\mathcal{L}_{\text{dis}}$ is the discriminator loss on both source and target graphs, which is optimized in an adversarial manner for S and D.

Segmentator loss For each input graph, the segmentator network outputs a parcellation prediction $\widehat{\mathbf{y}}$ where $\widehat{y}_{ic}$ is the probability that node i belongs to parcel c. In this work, we define the supervised segmentation loss as a combination of weighted Dice loss and weighted cross-entropy (CE),

$$\mathcal{L}_{\text{seg}}(\widehat{\mathbf{y}}, \mathbf{y}) = \left[1 - \frac{\epsilon + 2\sum_{i=1}^{N}\sum_{c=1}^{C} \omega_c\, y_{ic}\, \widehat{y}_{ic}}{\epsilon + \sum_{i=1}^{N}\sum_{c=1}^{C} \omega_c (y_{ic} + \widehat{y}_{ic})}\right] - \sum_{i=1}^{N}\sum_{c=1}^{C} \omega_c\, y_{ic}\, \widehat{y}_{ic}, \tag{3}$$

with y_{ic} being a one-hot encoding of the reference segmentation and ϵ a small constant to avoid zero-division. The weights ω_c balances the loss for parcels by increasing the importance given to smaller-sized regions. We follow [20] and set class weights ω_c as the total number of nodes divided by the number of nodes with label c. In the loss of Eq. (3), CE improves overall accuracy of node classification while Dice helps to have structured output for each parcel.

Discriminator loss Since the discriminator D is a domain classifier, we define its loss as the binary cross-entropy between its domain prediction (i.e., $\widehat{z} = 1$ for source or $\widehat{z} = 0$ for target):

$$\mathcal{L}_{\text{dis}}(\widehat{z}, z) \;=\; -(1-z)\log(1-\widehat{z}) \;-\; z\log\widehat{z}. \tag{4}$$

As mentioned before, this loss is maximized while updating the segmentator's parameters and minimized when updating the discriminator. Thus, the segmentator learns to produce surface parcellations that are alignment-invariant.

2.3 Network architecture

Segmentator: The segmentator is a fully-convolutional GCN comprised of 3 graph convolution layers with respective feature map sizes of 256, 128, and 32. At the input of the network, each node has 4 features: 3D spectral coordinates and an additional scalar measuring sulcal depth. All layers have $K_l = 6$ Gaussian kernels, similar to [20]. Since the output has 32 parcels, our last layer size is set to 32. In the last layer, softmax operation is applied for parcellation prediction, and the remaining layers employ Leaky ReLU as an activation function to obtain filter responses in Eq. (1).

Discriminator: Similar to the segmentator network, we use 2 graph convolution layers, an average pooling layer, and 3 fully connected (linear) layers for classifying the segmentation domain. The first graph convolution layer takes segmentation predictions with 32 feature maps as input. Moreover, the output sizes of the first two layers output are 128 and 64, respectively. Average pooling is used to reduce the input graph to a 1-D vector for the classification task. Three fully-connected layers are placed at the end of the network, with respective sizes of 32, 16, and 1. Each graph convolution layer has $K_l = 6$ Gaussian kernels. Sigmoid activation is applied to the last linear layer to predict the input domain of the graph sample and the remaining layers use Leaky ReLU.

3 Results

We evaluate the performance of our method using MindBoggle [2], the largest manually-labeled brain surface dataset. This dataset contains the cortical mesh data of 101 subjects aggregated from multiple sites. Each brain surface includes 32 manually labeled parcels. We split this dataset into 70-10-20 training, validation and test sets. The training set has only 35 samples for the source and target domains each. To have more training samples and thus reduce overfitting,

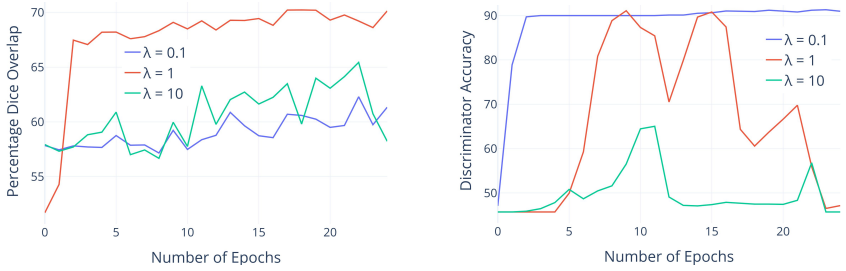

Fig. 2. Effect of hyper-parameter λ: Segmentation performance in mean Dice (**left**) and Discriminator classification accuracy (**right**) on test examples, obtained for $\lambda \in \{0.1, 1, 10\}$.

we sub-sample the node embeddings of each mesh to generate 25 examples of 10K nodes. This data augmentation technique, which is not possible in regular CNNs, is enabled by the spectral embedding of our approach.

Let P_c be the nodes predicted as having label $c \in \{1, \ldots, 32\}$, and G_c be the actual set of nodes with this label in the ground-truth parcellation. We evaluate performance using the mean Dice overlap:

$$\text{MeanDice}(\mathbf{P}, \mathbf{G}) \;=\; \frac{1}{32} \sum_{c=1}^{32} \frac{2\,|P_c \cap G_c|}{|P_c| + |G_c|}. \tag{5}$$

All experiments were carried out on an i7 desktop computer with 16GB of RAM and an Nvidia Titan X 12 GB GPU. The code for our work is available at the following URL: `https://tinyurl.com/yawdw7hh`.

3.1 Effect of λ on parcellation

The loss function for adversarial training involves hyper-parameter λ, which controls the trade-off between parcellation accuracy on labeled source data and fooling the discriminator (i.e., alignment invariance). To assess the impact of this important hyper-parameter on performance, we show in Fig. 2 the segmentator mean Dice and discriminator classification accuracy on test examples at different training epochs, for $\lambda \in \{0.1, 1, 10\}$. As expected, when using a large $\lambda = 10$, the model focuses mostly on fooling the discriminator. This results in a low segmentation Dice, and a discriminator accuracy near 50% since the discriminator cannot distinguish between source and target parcellation outputs. Conversely, for a small $\lambda = 0.1$, the adversarial training gives less importance to fooling the discriminator, which translates in a high discriminator accuracy. However, this also leads to a poor performance on target examples, since the parcellation output for these examples differs greatly from those of source examples. This illustrates that a stronger adversarial learning is required to align the source and target domains. For the rest of our experiments, we selected $\lambda = 1$ based on the parcellation accuracy for *validation* examples.

Table 1. Comparison with surface segmentation approaches: Mean Dice and standard deviation on test data. The first result column corresponds to the default setting where test (i.e., target domain) graphs are not aligned. For the second column, test graphs were aligned on the same reference as training (i.e., source domain) graphs. Result columns 3-7 correspond to the setting where all test graphs are aligned to four randomly-selected target graphs (a different graph for each column). Bold font highlights a performance statistically higher than all other methods (t-test $p < 0.01$).

Method	**No alignment**	**Alignment to reference graph**				
		Source	**Rand. target 1**	**Rand. target 2**	**Rand. target 3**	**Rand. target 4**
Spectral RF [23]	65.4 ± 9.0	81.9 ± 3.4	60.0 ± 1.8	55.3 ± 2.1	60.2 ± 4.0	55.2 ± 3.0
Seg-GCN [20]	71.4 ± 7.9	86.5 ± 2.8	67.8 ± 2.0	58.8 ± 2.8	63.5 ± 3.2	60.1 ± 3.6
Adv-GCN (ours)	$\mathbf{73.8 \pm 6.0}$	85.7 ± 3.5	$\mathbf{73.5 \pm 2.0}$	$\mathbf{72.5 \pm 2.6}$	$\mathbf{72.4 \pm 2.4}$	$\mathbf{71.7 \pm 3.3}$

3.2 Comparison with the state-of-the-art

We next compare our method, called Adv-GCN in the following results, against two other graph-based approaches for surface parcellation. This first one is the Spectral Random Forest (RF) algorithm proposed in [23], which performs the same spectral graph embedding as our method, and then uses the spectral coordinates and sulcal depth at individual nodes to train a RF classifier. As done in [23], we employed 50 trees to build the RF model. This comparison baseline was included to show the limitation of point-based approaches which ignore the relationship between nodes when predicting labels. The second approach, called Seg-GCN, is the same segmentation GCN as in our method, but trained without the adversarial loss. For this baseline, which is similar to the method presented in [20], our goal is to show the benefit of learning an alignment-invariant representation with adversarial domain adaptation.

The surface parcellation approaches are compared in three different test settings. In the first one, the approaches are applied on target examples without any alignment. This corresponds to the normal application setting of our alignment-invariant method. For the second one, we align all target examples on the same reference surface as the one used for source examples. This setting requires to retain the reference surface and apply ICP alignment in pre-processing for each test surface. Finally, in the third setting, target examples are aligned to a reference surface chosen randomly in the test set. This last setting corresponds to the case where we want to parcel surfaces from a dataset which was processed differently than the source dataset.

Results of this experiment are summarized in Table 1. When test examples are aligned to the same source reference (i.e., no domain shift), our segmentation GCN architecture, with or without adversarial learning, outperforms Spectral RF by a large margin. This illustrates the importance of considering the relationship between different nodes in the graph, as in our graph convolution model. However, when applied to non-aligned test surface, our Adv-GCN

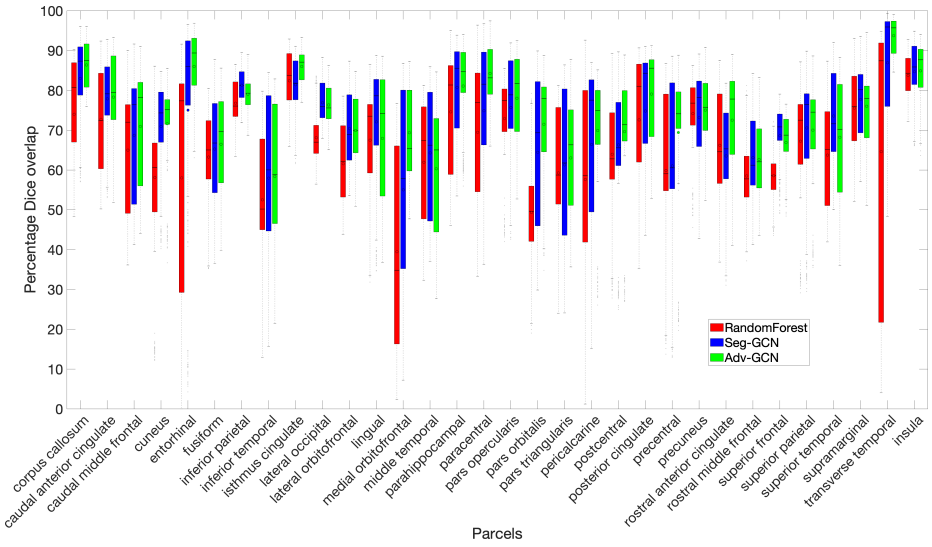

Fig. 3. Segmentation Dice for individual parcels: Box-plot of mean Dice overlap achieved by three different methods for all 32 cortical parcels when *test subjects are not aligned*.

method achieves a 2.4% improvement in mean Dice over Seg-GCN, and 8.4% over Spectral RF. This demonstrates the benefit of learning an alignment-invariant representation via adversarial domain adaptation. Furthermore, the improvement provided by our Adv-GCN method is even more significant for surfaces aligned to a random target reference (last four columns of Table 1). Thus, across the four random target references, Adv-GCN yields an average improvement of 14.9% compared to Spectral RF and 10.0% compared to Seg-GCN. This shows the strength of adversarial learning to match the output distribution for two fixed domains.

The average Dice overlap for individual parcels is shown in Fig. 3. As can be seen, Adv-GCN provides a higher mean and smaller variance for most of the 32 parcels. By inspecting results, we find that accuracy is correlated with parcel size, with larger parcels generally better segmented than smaller ones. Figure 4 shows qualitative results for different graph segmentation methods. As highlighted by the red circle, our Adv-CGN gives a more accurate segmentation compared to Seg-GCN and Spectral RF, with an improvement over 13% in parcel-averaged Dice.

4 Conclusion

In this paper, we presented a novel adversarial domain adaptation framework for brain surface parcellation. The proposed algorithm leverages an adversarial training mechanism to obtain an alignment-invariant surface segmentation,

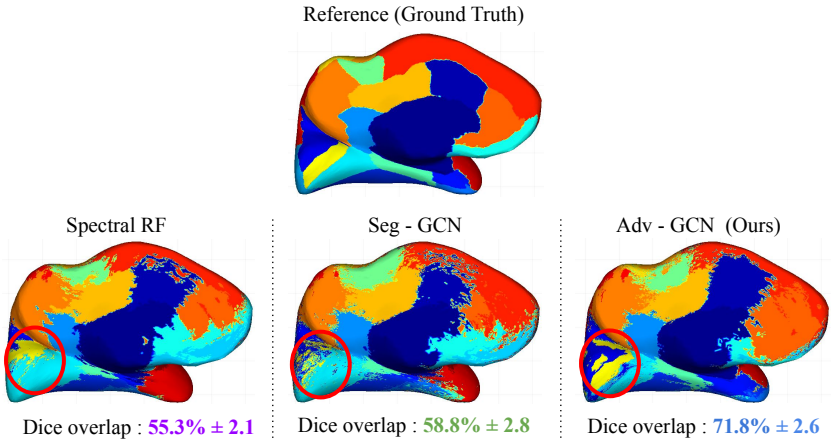

Fig. 4. Qualitative comparison: Parcellation outputs of the three surface segmentation approaches for a single non-aligned test surface. For better visualization, segmented parcels are drawn on an inflated surface. For each approach, we report the average Dice and standard deviation computed over the 32 parcels. As highlighted by the red circle, our adversarial GCN (Adv-CGN) gives a more accurate segmentation compared to the same model without adversarial training (Seg-GCN) and Spectral Random Forest (RF).

and overcomes the limitations of spectral GCNs [21,22] that require finding an explicit alignment of graph eigenbases. Table 1 shows a clear improvement in performance over the same spectral GCN without adversarial training (Seg-GCN) and the Spectral Random Forest (RF) algorithm [23]. Specifically, our method yields a 2.4% mean Dice improvement over Seg-GCN and 8.4% over Spectral RF, for non-aligned test surfaces. This improvement reaches over 10% for test surface aligned to a random target reference. Qualitative results in Fig. 4 illustrate the better parcellation of our method for non-aligned surfaces.

In some experiments, we observed a tendency of the discriminator to overfit the training set, which impeded domain adaptation in the learning process. In a future study, two strategies could be explored to overcome this problem: using other types of discriminator, for instance the Least Squares GAN [24] or Wasserstein GAN [25], and applying data augmentation on labeled brain surface meshes. While our adversarial graph domain adaptation technique was demonstrated on cortical parcellation, it also has potential for other surface segmentation problems where a domain shift is present. Likewise, our method could be useful for semi-supervised surface segmentation, thereby mitigating the need for large amounts of labeled surfaces. In this setting, the same architecture could be used, however the discriminator would predict if the segmentation output is for a labeled or unlabeled example from the same domain. We plan to evaluate the impact of higher frequency input representations with performance measures such as Hausdorff distance in future work.

Acknowledgments This research work was partly funded by the Fonds de Recherche du Quebec (FQRNT) and Natural Sciences and Engineering Research Council of Canada (NSERC). We gratefully acknowledge the support of NVIDIA Corporation for the donation of the Titan X Pascal GPU used for this research.

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. 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)
3. Tajbakhsh, N., Jeyaseelan, L., Li, Q., Chiang, J., Wu, Z., Ding, X.: Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. Medical Image Analysis (2019)
4. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. (2014)
5. Zhang, Y., David, P., Gong, B.: Curriculum domain adaptation for semantic segmentation of urban scenes. In: Proceedings of the IEEE International Conference on Computer Vision. (2017)
6. Zou, Y., Yu, Z., Vijaya Kumar, B., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: Proceedings of the European conference on computer vision (ECCV). (2018)
7. Ghafoorian, M., Mehrtash, A., Kapur, T., Karssemeijer, N., Marchiori, E., Pesteie, M., Guttmann, C.R., de Leeuw, F.E., Tempany, C.M., van Ginneken, B., et al.: Transfer learning for domain adaptation in mri: Application in brain lesion segmentation. In: International conference on medical image computing and computer-assisted intervention, Springer (2017)
8. Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P.: Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019)
9. Zhang, Y., Miao, S., Mansi, T., Liao, R.: Task driven generative modeling for unsupervised domain adaptation: Application to x-ray image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer (2018)
10. Javanmardi, M., Tasdizen, T.: Domain adaptation for biomedical image segmentation using adversarial training. In: 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), IEEE (2018)
11. Long, M., Cao, Y., Wang, J., Jordan, M.: Learning transferable features with deep adaptation networks. In: International Conference on Machine Learning. (2015)
12. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: International Conference on Machine Learning. (2015)
13. Tsai, Y.H., Hung, W.C., Schulter, S., Sohn, K., Yang, M.H., Chandraker, M.: Learning to adapt structured output space for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018)
14. Hoffman, J., Wang, D., Yu, F., Darrell, T.: Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. arXiv preprint arXiv:1612.02649 (2016)

15. Kamnitsas, K., Baumgartner, C., Ledig, C., Newcombe, V., Simpson, J., Kane, A., Menon, D., Nori, A., Criminisi, A., Rueckert, D., et al.: Unsupervised domain adaptation in brain lesion segmentation with adversarial networks. In: International conference on information processing in medical imaging. (2017)
16. Bateson, M., Kervadec, H., Dolz, J., Lombaert, H., Ayed, I.B.: Constrained domain adaptation for segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. (2019)
17. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: Going beyond euclidean data. IEEE Signal Processing (2017)
18. Monti, F., Boscaini, D., Masci, J., Rodolà, E., Svoboda, J., Bronstein, M.: Geometric deep learning on graphs using mixture model CNNs. In: CVPR. (2017)
19. Cucurull, G., Wagstyl, K., Casanova, A., Veličković, P., Jakobsen, E., Drozdzal, M., Romero, A., Evans, A., Bengio, Y.: Convolutional neural networks for mesh-based parcellation of the cerebral cortex. In: MIDL. (2018)
20. Gopinath, K., Desrosiers, C., Lombaert, H.: Graph convolutions on spectral embeddings for cortical surface parcellation. Medical image analysis (2019)
21. Bruna, J., Zaremba, W., Szlam, A., Lecun, Y.: Spectral networks and locally connected networks on graphs. In: ICLR. (2014)
22. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: NIPS. (2016)
23. Lombaert, H., Criminisi, A., Ayache, N.: Spectral forests: Learning of surface data, application to cortical parcellation. In: MICCAI. (2015)
24. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: Proceedings of the IEEE international conference on computer vision. (2017) 2794–2802
25. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein GAN. arXiv preprint arXiv:1701.07875 (2017)