

2D histology meets 3D topology: Cytoarchitectonic brain mapping with Graph Neural Networks

Christian Schiffer^{1,2}, Stefan Harmeling³, Katrin Amunts^{1,4}, and Timo Dickscheid^{1,2}

¹Institute of Neuroscience and Medicine (INM-1), Research Centre Jülich, Germany

²Helmholtz AI, Research Centre Jülich, Germany

³Heinrich Heine University, Düsseldorf, Germany

⁴Cécile & Oscar Vogt Institute for Brain Research, University Hospital Düsseldorf, Germany

Abstract

Cytoarchitecture describes the spatial organization of neuronal cells in the brain, including their arrangement into layers and columns with respect to cell density, orientation, or presence of certain cell types. It allows to segregate the brain into cortical areas and subcortical nuclei, links structure with connectivity and function, and provides a microstructural reference for human brain atlases. Mapping boundaries between areas requires to scan histological sections at microscopic resolution. While recent high-throughput scanners allow to scan a complete human brain in the order of a year, it is practically impossible to delineate regions at the same pace using the established gold standard method. Researchers have recently addressed cytoarchitectonic mapping of cortical regions with deep neural networks, relying on image patches from individual 2D sections for classification. However, the 3D context, which is needed to disambiguate complex or obliquely cut brain regions, is not taken into account. In this work, we combine 2D histology with 3D topology by reformulating the mapping task as a node classification problem on an approximate 3D midsurface mesh through the isocortex. We extract deep features from cortical patches in 2D histological sections which are descriptive of cytoarchitecture, and assign them to the corresponding nodes on the 3D mesh to construct a large attributed graph. By solving the brain mapping problem on this graph using graph neural networks, we obtain significantly improved classification results. The proposed framework lends itself nicely to integration of additional neuroanatomical priors for mapping.

1 Introduction

Cytoarchitectonic areas are characterized by a distinct spatial organization of neuronal cells in the brain, including their arrangement into layers and columns, as well as density, orientation, or presence of certain types of cells. As indicators for connectivity and functional modules, they provide an important microstructural reference for human brain atlases [2]. Mapping the borders of cytoarchitectonic areas requires to analyze brain sections at microscopic resolution. To capture the human brain’s considerable variability, it needs to be performed in many different brain samples. [2] created a comprehensive probabilistic cytoarchitectonic atlas based on delineations in serial histological sections stained for cell bodies in a sample of ten brains, using image analysis and statistical tools to identify architectonic borders [22]. These resulting maps are aggregated in a 3D reference space at the millimeter scale. Although today’s high-throughput scanners allow to digitize complete human brains at micrometer resolution, it is practically impossible to scale this approach for doing delineations in all sections of a human brain, which may have, in dependence of the size, approx. 6000-8000 sections. This motivates the development of automatic mapping algorithms.

Recent work formulated cytoarchitectonic mapping of cortical regions as an image segmentation [23, 24] or classification [21] problem, which can be approached with deep convolutional networks. These methods process each 2D section individually, responding to the lack of routine workflows for computing a precise 3D reconstruction from histology en par with high throughput imaging. In fact, as of today only one 3D reconstruction of a whole human brain from histology is available [1], with a spatial resolution of 20 micrometer isotropic. The 3D topology of a brain is thus not used by these models, although it is key to disambiguate complex or obliquely cut brain regions. Other authors suggested to incorporate inter-slice information using 3D convolutions [19] or recurrent networks [3], which again requires a precise 3D reconstruction.

Here we present a novel paradigm for cytoarchitectonic brain mapping which overcomes the above restrictions. The basic idea is to reformulate the mapping task as a node classification problem on the approximate 3D midsurface mesh through the isocortex. Since the mesh is not assumed to be precise, it can be derived from a simple linear reconstruction which is straightforward to compute from a histological image stack, and can handle a significant amount of missing sections. Building on [21], we extract deep features from 2D cortical patches at microscopic resolution using convolutional neural networks (CNNs) that were trained with a contrastive learning task to encode cytoarchitectonic characteristics. These features are then assigned to the corresponding nodes on the reconstructed surface mesh to construct a large, attributed graph. The brain mapping problem is solved on this graph using graph neural networks (GNNs) [11, 8].

We make the following contributions: 1) We introduce a novel deep learning approach for cytoarchitecture classification in large stacks of whole brain sections which integrates high-resolution 2D texture features with global 3D topology. 2) The approach outperforms the state of the art on DL-based cytoarchitectonic mapping on a dataset of histological sections from eight postmortem human brains. 3) The approach allows flexible integration of neuroanatomical priors into the mapping problem, which further boosts classification performance.

2 Methods

The proposed framework consists of three components: 1) A linear 3D reconstruction of the histological stack to compute an approximate midsurface mesh of the isocortex, which is interpreted as a graph. 2) A CNN trained with contrastive learning which extracts cortical features from the histological sections, serving as node attributes in the graph. 3) A GNN to label each graph node with a cytoarchitectonic area, exploiting both high-resolution texture and approximate 3D topology encoded in the graph.

Construction of the approximate midsurface graph A coarse segmentation of each section into background, white matter and cortical gray matter is performed using morphological active contours [17]. Histological brain sections (Fig. 1, left) are 3D reconstructed by rigid alignment of adjacent sections using feature based registration as described in [6]. From the segmented linear 3D reconstruction, a 3D cortical midsurface mesh is extracted by solving Laplace’s equation inside the cortical mantle [13] using the *highres-cortex* package¹ and extracting the 0.5 isosurface from the resulting volumetric Laplacian fields using marching cubes [14]. In a manual postprocessing stage, hemispheres are split, and small disconnected components as well as cerebellum and brainstem are removed from the mesh. Finally, isotropic explicit remeshing is applied using *meshlab*[4] to unify edge lengths in the mesh to $\approx 300\mu\text{m}$, thereby reducing the number of triangles and ensuring that connections between vertices represent comparable distances. Each resulting mesh (Fig. 1, right) is then interpreted as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with nodes $\mathcal{V}$ and edges $\mathcal{E}$.

¹<https://github.com/neurospin/highres-cortex>

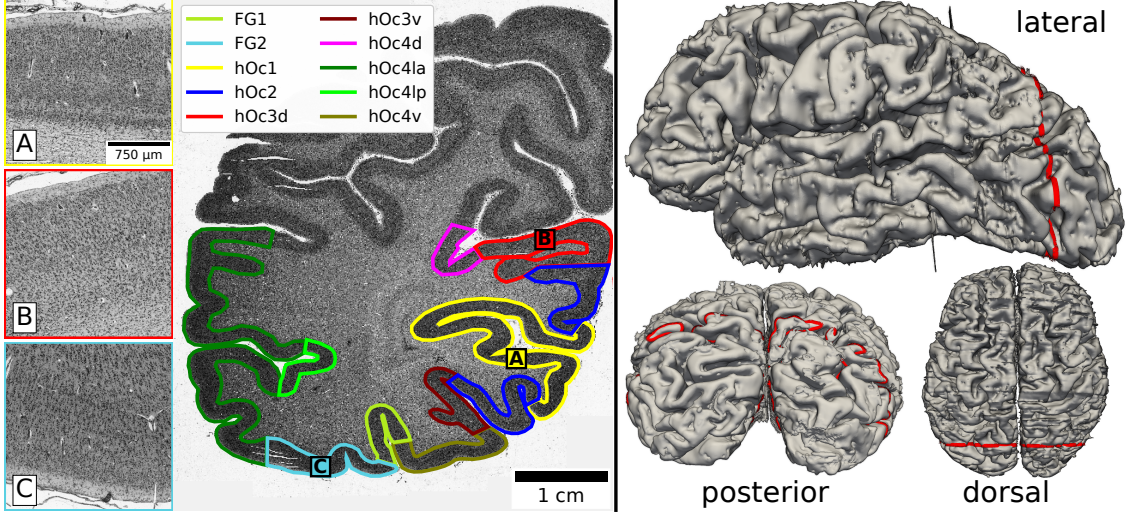


Figure 1: **Left:** Histological brain section from brain B01 (see also Fig. 2, left) with expert annotations of cytoarchitectonic areas shown in different colors. **Right:** Approximate midsurface through the isocortex extracted from a linear reconstruction of the histological stack. Points on the surface corresponding to points in the image on the left are shown in red.

Computing node level texture features Each node $u \in \mathcal{V}$ in the resulting graph can be uniquely identified with an image coordinate p_u in a 2D histological section by inverting the rigid transformation applied for 3D reconstruction. We apply an encoder CNN f to extract a cytoarchitectonic feature embedding $\mathbf{h}_u$ from an image patch $\mathbf{x}$ centered at each coordinate p_u . The encoder f is trained using the contrastive approach proposed in [21] to produce an embedding that maps image patches from the same brain area to similar feature vectors, and image patches from different brain areas to dissimilar feature vectors. Given an image patch $\mathbf{x}$, f produces a lower dimensional vector $\mathbf{h} = f(\mathbf{x}) \in \mathbb{R}^{D_e}$, which is passed through a projection network $g(\mathbf{h}) = \tilde{\mathbf{z}} \in \mathbb{R}^{D_p}$ and normalized as $\mathbf{z}_i = \tilde{\mathbf{z}}_i / \|\tilde{\mathbf{z}}_i\|_2$. Given a minibatch of N image patches $\mathbf{x}_i$ with corresponding labels y_i ($i = 1, \dots, N$), the contrastive loss

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N \mathcal{L}_i \quad (1)$$

$$\mathcal{L}_i = -\frac{1}{N_{y_i}} \sum_{j=1}^N \mathbb{I}_{i \neq j} \mathbb{I}_{y_i = y_j} \log \frac{e^{\langle \mathbf{z}_i, \mathbf{z}_j \rangle / \tau}}{\sum_{k=1}^N \mathbb{I}_{i \neq k} e^{\langle \mathbf{z}_i, \mathbf{z}_k \rangle / \tau}} \quad (2)$$

is optimized during training, where $\mathbb{I}$ is the indicator function, $\tau \in \mathbb{R}$ is a temperature scaling parameter and N_{y_i} is the number of batch elements with the same label as $\mathbf{x}_i$. The projection network g is discarded after training.

Graph neural networks We consider two GNN architectures to integrate spatial relationships of image patches into the classification model: GraphSAGE [8] and Graph Attention Network (GAT) [25]. Both architectures are suitable for inductive tasks, so that trained networks can be applied to previously unseen nodes or graphs. Due to common memory constraints, training on full graphs is not possible given the level of detail of our models. We therefore adapt the neighborhood sampling scheme proposed in [8]: Given a node $u \in \mathcal{V}$ and a GNN with K layers, we sample its K -hop neighborhood $\mathcal{N}_K(u)$ and use the resulting subgraph as input for the GNN. This way, information from the immediate neighborhood of u can be propagated through the GNN to classify u .

Integrating anatomical priors Previous work [23] showed that the integration of prior anatomical knowledge - there given in the form of probabilistic cytoarchitectonic maps [2] - can improve classification performance. Inspired by this finding, we adapt the registration workflow from [2] to project probabilistic maps of 152 brain areas from the MNI Colin27 reference space onto the histological sections, and assign an additional vector $\mathbf{h}_u^P$ to each node u . Each dimension of $\mathbf{h}_u^P$ encodes the probability to belong to a particular region of the Julich-Brain atlas [2]. We further consider spatial locations in Colin27 space as another anatomical prior by assigning a 3D coordinate vector $\mathbf{h}_u^C$ to each node u , again using the workflow from [2].

Implementation details Models were implemented using *pytorch* [20] and *pytorch-geometric* [7]. Training was performed on the HPC clusters JURECA [12] (NVidia K80, 12GB) and JURECA-DC (NVidia A100, 40GB) using 4 to 32 GPUs. Code will be made publicly available².

3 Experiments and Results

We systematically evaluate all components of the proposed approach: 1) We compare different encoder architectures for contrastive feature learning from images patches. 2) Using the best performing encoder architecture, we compare the performance of GNNs with **SAGE** and **GAT** architectures. 3) We investigate the additional benefit of adding neuroanatomical priors for classification.

Dataset We use an in-house dataset containing images of 1860 histological sections from 7 human postmortem brains acquired from the body donor programs of the Anatomical Institute of the University of Düsseldorf (Germany), and corresponding annotations of 113 cytoarchitectonic cortical areas. Sections have an approximate thickness of 20 μ m, were marked for neuronal cell bodies using a modified Merker stain [18], and imaged at a resolution of 1 μ m with a light-microscopic scanner (TissueScope, Huron Digital Pathology Inc.). 80% of the sections are used for training ($\mathcal{X}_{tr}$), remaining sections for testing ($\mathcal{X}_{te}$). Transferability to unseen brains is evaluated on 325 sections from an 8th brain ($\mathcal{X}_{un}$). Graph nodes $\mathcal{V}$ are split into $\mathcal{V}_{tr}$, $\mathcal{V}_{te}$ and $\mathcal{V}_{un}$, containing nodes with corresponding points p_u in $\mathcal{X}_{tr}$, $\mathcal{X}_{te}$ and $\mathcal{X}_{un}$, respectively. Performance is evaluated using macro-F1 score on nodes $\mathcal{V}_{te}$ and $\mathcal{V}_{un}$. The study requires no ethical approvals.

Performance of different feature encoders The encoder network f is trained on 1200 image patches per class sampled from $\mathcal{X}_{tr}$, oversampling small areas for class balancing. Data augmentation steps include random pixel intensity augmentation, rotation, mirroring, translation, blurring and sharpening with parameters detailed in [21]. Training is performed for 150 epochs using LARS optimizer [26] with constant learning rate $0.01 * N/128$, batch size $N = 4096$ ($N = 2048$ for **res[w,1]** due to memory constraints) and $\tau = 0.07$. We consider the two network architectures listed in Table 1: **base** is the architecture presented in [23, 24, 21] which we include as baseline. **res** uses pre-activation residual building blocks [9] and is based on ResNet18. For both architectures, we also consider wider variants with twice as many channels per layer (**base[w]**, **res[w]**). For **res**, we further investigate larger input image patches (**res[1]**, **res[w,1]**). The following image patch sizes are used (2 μ m/pixel): 1129² pixel for **base/base[w]**, 1024² pixel for **res/res[w]**, 2048² pixel for **res[1]/res[w,1]**. Performance is evaluated on $\mathcal{V}_{te}$ and $\mathcal{V}_{un}$ by training a multi-layer perceptron (MLP) (three layers à 256 hidden units, batch normalization (BN) [10], ReLU) to classify brain areas from features extracted by each considered model f for all nodes in $\mathcal{V}_{tr}$. MLP is trained for 100 epochs using SGD with Nesterov momentum 0.9, constant learning rate $0.001 * N/4096$, batch size $N = 16384$ and cross-entropy loss. We apply dropout with probability 0.5 and 0.25 to the input and hidden layers, respectively.

²https://jugit.fz-juelich.de/c.schiffer/miccai2021_2d_histology_meets_3d_topology

Table 1: Network architecture of **base** and **res** models. **res** uses pre-activation residual connections as in ResNet18 [9]. **base** uses no padding, **res** uses zero-padding (CONV(k, c, s): c -channel $k \times k$ stride s convolutional layer, MP(k, s): $k \times k$ stride s max-pooling, FC(d): d -dimensional fully connected layer, GAP: global average pooling).

layer name	base	res
conv_1_x	CONV(5, 16, 4), CONV(3, 16, 1), MP(2, 2)	
conv_2_x	CONV(3, 32, 1) $\times$ 2, MP(2, 2)	CONV(3, 32, 1) $\times$ 4
conv_3_x	CONV(3, 64, 1) $\times$ 2, MP(2, 2)	CONV(3, 64, 2), CONV(3, 64, 1) $\times$ 3
conv_4_x	CONV(3, 64, 1) $\times$ 2, MP(2, 2)	CONV(3, 64, 2), CONV(3, 64, 1) $\times$ 3
conv_5_x	CONV(3, 128, 1) $\times$ 2, MP(2, 2)	CONV(3, 128, 2), CONV(3, 128, 1) $\times$ 3
conv_6_x	CONV(3, 128, 1) $\times$ 2	-
projection	GAP, FC(128), FC(128)	

Table 2: Macro-F1 scores (average across three runs) obtained on $\mathcal{V}_{te}$ and $\mathcal{V}_{un}$. *Left*: Independent patch classification based on features extracted using different encoder architectures. *Center*: Node level classification on graphs using different GNN architectures and features extracted with **res[w,1]** (MLP performance for comparison). *Right*: Node level classification exploiting prior knowledge using **SAGE[5+r]**, features from **res[w,1]** and different combinations of cytoarchitectonic features (CY), probabilistic maps (PM) and canonical coordinates (CO).

model	$\mathcal{V}_{te}$	$\mathcal{V}_{un}$	model	$\mathcal{V}_{te}$	$\mathcal{V}_{un}$	model	$\mathcal{V}_{te}$	$\mathcal{V}_{un}$
base	30.62	11.14	MLP	50.57	18.99	CY	70.18	20.22
base[w]	36.94	10.40	SAGE[3]	66.48	20.75	CY/PM	77.06	30.90
res	35.17	13.47	SAGE[3+r]	68.64	19.84	CY/CO	79.28	33.40
res[1]	44.67	17.82	SAGE[5+r]	70.18	20.22	CY/PM/CO	79.93	32.67
res[w]	44.08	13.72	GAT[3]	66.40	20.17	PM	47.39	30.43
res[w,1]	50.57	18.99	GAT[3+r]	61.98	20.02	CO	56.30	34.19
			GAT[5+r]	63.46	20.06	PM/CO	56.03	32.30

We found that **res** models perform better than **base** models (Table 2 left). Wider networks and a larger input size improve performance on $\mathcal{V}_{te}$. On $\mathcal{V}_{un}$, increased input size (**res[1]**) outperforms wider networks (**res[w]**). Best results are obtained by combining both (**res[w,1]**).

Graph neural networks We create one graph per brain hemisphere, resulting in 16 graphs overall (1.2 million nodes with node degree 6, on average). Training nodes are sampled from $\mathcal{V}_{tr}$, inversely proportional to their class frequency to account for class imbalance. Node features are computed using **res[w,1]**. Node labels are assigned using annotations of cytoarchitectonic areas. We consider two basic GNN architectures: **SAGE[3]** consists of three GraphSAGE layers with mean aggregation [8], while **GAT[3]** consists of three GAT layers [25]. Following [15, 16], we further investigate variants with pre-activation residual connections (**SAGE[3+r]**, **GAT[3+r]**) and five layers (**SAGE[5+r]**, **GAT[5+r]**). All layers use 256 hidden units, BN and ReLU activation. GAT layers use 8 attention heads with 32 units each (256 units in total). Remaining training parameters are identical to those used for MLP. For **SAGE** models, we adapt fixed-size neighborhood sampling from [8] with a neighborhood size of three. For **GAT** models, we sample full neighborhoods, but apply dropout with probability 0.5 to attention coefficients as proposed in [25]. Both sampling methods aim to make models robust against missing nodes or edges.

All GNN architectures obtain significantly higher scores than MLP on both $\mathcal{V}_{te}$ and $\mathcal{V}_{un}$ (Table 2, center). **SAGE[5+r]** achieves overall best scores, with an increase by 19.61 points and 1.23 points on $\mathcal{V}_{te}$ and $\mathcal{V}_{un}$ compared to MLP, respectively. Using the same number of layers and residual connections, **SAGE** outperforms **GAT** in almost all cases.

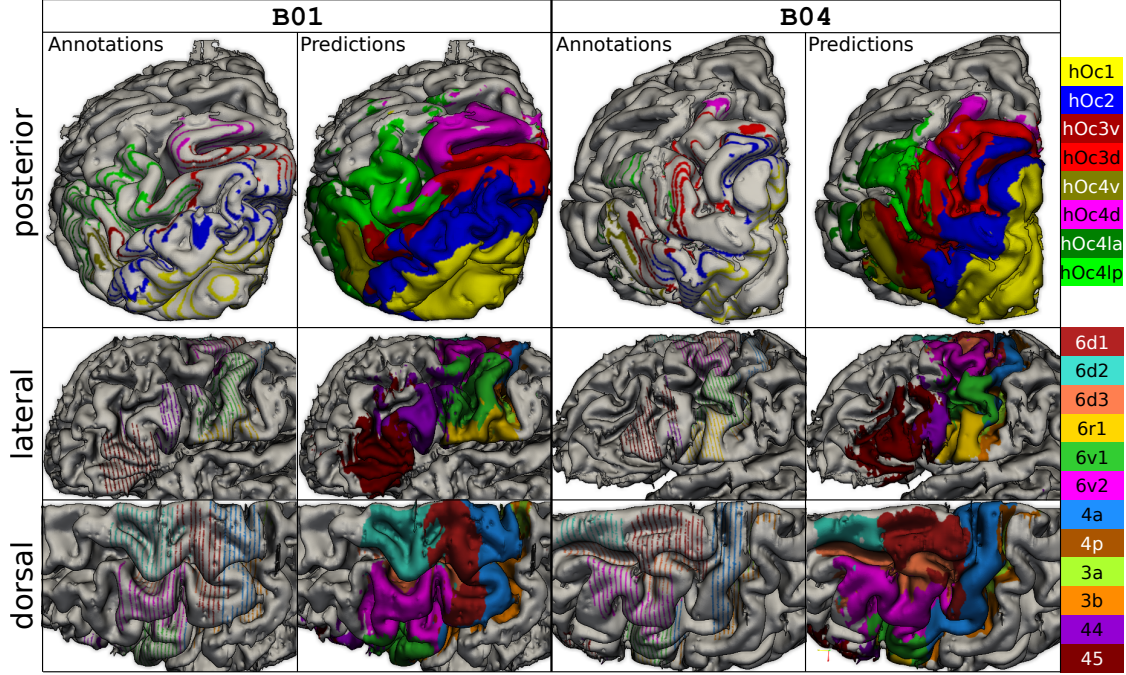


Figure 2: Visualization of available expert annotations and predictions by **SAGE[5+r]** for the left hemisphere of two brains (B01 and B04). The model was trained using features from **res[w,1]** (**CY**) as well as anatomical priors in the form of probabilistic maps (**PM**) and canonical coordinates (**C0**). Annotations include both training and test labels. Since they are only available for a subset of sections, they appear as stripes.

Incorporating prior knowledge We examine classification performance under all possible combinations of using cytoarchitectonic features extracted from images (**CY**), weights from probabilistic cytoarchitectonic maps (**PM**) [2], and canonical 3D locations (**C0**) at each node. Dropout with probability 0.5 and additive Gaussian white noise with standard deviation 0.05 is applied to $\mathbf{h}_u^P$ and $\mathbf{h}_u^C$, respectively. $\mathbf{h}_u^P$ and $\mathbf{h}_u^C$ are each projected by a fully connected layer (256 units, BN, ReLU) before concatenation.

Incorporating both **PM** and **C0** with the texture features **CY** improves scores by 9.75 points on $\mathcal{V}_{te}$ and 12.45 points on $\mathcal{V}_{un}$ (Table 2, right; Fig. 2). Using them in isolation, **C0** leads to slightly better performance than **PM**. Scores on $\mathcal{V}_{te}$ drop significantly when removing texture features **CY**. Best scores on $\mathcal{V}_{un}$ are obtained using only canonical coordinates, without texture features.

4 Discussion & conclusion

We presented a graph neural network approach for cytoarchitectonic classification of cortical image patches in microscopic scans of human brain sections, which combines 2D texture features with topological information from an approximate 3D surface reconstruction. While [5] employed GNNs for surface parcellation of Broca’s areas 44 and 45 using MRI-based features, the present work is the first to our knowledge that integrates deep texture features from histology with 3D topology in a graph framework to label a large number of highly different brain areas across several brains. The proposed method significantly outperforms previous methods that operate only on individual 2D sections: The best trained model increases classification scores by 161%/193% on $\mathcal{V}_{te}/\mathcal{V}_{un}$ wrt. the recent baseline model **base** representing the work from [21]. Our experiments further suggest that deeper and wider architectures for the CNN encoder f benefit performance, motivating more systematic architecture search in the future. Larger input sizes in the encoders **res[1]** and **res[w,1]** showed additional positive impact by

enabling models to capture the entire width of the isocortex (2mm-4mm). **SAGE** outperformed **GAT**, suggesting that mean aggregation (**SAGE**) might be better suited than self-attention (**GAT**) for our use case. The presented framework allows straightforward integration of anatomical priors, and the results indicate that this might be a crucial strategy for optimizing cytoarchitectonic mapping with deep networks. Unfortunately, transferability of learned features to completely unseen brains still seems to be limited, as indicated by the performance gaps between $\mathcal{V}_{te}$ and $\mathcal{V}_{un}$ and thus confirming observations in [21]. In the future, we plan to perform more rigorous architecture search for the encoders, study more anatomical priors, and investigate the reasons underlying the reduced performance on unseen brain samples.

Acknowledgements This project received funding from the European Union’s Horizon 2020 Research and Innovation Programme, grant agreement 945539 (HBP SGA3), and from the Helmholtz Association’s Initiative and Networking Fund through the Helmholtz International BigBrain Analytics and Learning Laboratory (HIBALL) under the Helmholtz International Lab grant agreement InterLabs-0015. Computing time was granted through JARA on the super-computer JURECA at Jülich Supercomputing Centre (JSC).

References

- [1] Amunts, K., Lepage, C., Borgeat, L., Mohlberg, H., Dickscheid, T., Rousseau, M.É., Bludau, S., Bazin, P.L., Lewis, L.B., Oros-Peusquens, A.M., Shah, N.J., Lippert, T., Zilles, K., Evans, A.C.: BigBrain: An Ultrahigh-Resolution 3D Human Brain Model. *Science* **340**(6139), 1472–1475 (2013). <https://doi.org/10.1126/science.1235381>
- [2] Amunts, K., Mohlberg, H., Bludau, S., Zilles, K.: Jülich-Brain: A 3D probabilistic atlas of the human brain’s cytoarchitecture. *Science* **369**(6506), 988 (2020). <https://doi.org/10.1126/science.abb4588>
- [3] Chen, J., Yang, L., Zhang, Y., Alber, M., Chen, D.Z.: Combining fully convolutional and recurrent neural networks for 3d biomedical image segmentation. In: *Advances in Neural Information Processing Systems*. pp. 3036–3044 (2016)
- [4] Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., Ranzuglia, G.: MeshLab: An open-source mesh processing tool. In: Scarano, V., Chiara, R.D., Erra, U. (eds.) *Eurographics Italian Chapter Conference*. Salerno, Italy, The Eurographics Association (2008)
- [5] Cucurull, G., Wagstyl, K., Casanova, A., Velićkovic, P., Jakobsen, E., Drozdal, M., Romero, A., Evans, A., Bengio, Y.: Convolutional neural networks for mesh-based parcellation of the cerebral cortex. In: *Medical Imaging with Deep Learning* (2018)
- [6] Dickscheid, T., Haas, S., Bludau, S., Glock, P., Huysegoms, M., Amunts, K.: Towards 3D Reconstruction of Neuronal Cell Distributions from Histological Human Brain Sections. *Future Trends of HPC in a Disruptive Scenario* **34**, 223 (2019)
- [7] Fey, M., Lenssen, J.E.: Fast graph representation learning with PyTorch Geometric. In: *ICLR Workshop on Representation Learning on Graphs and Manifolds* (2019)
- [8] Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. pp. 1025–1035 (2017)
- [9] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 770–778 (2016)
- [10] Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *International Conference on Machine Learning*. pp. 448–456 (2015)
- [11] Kipf, T.N., Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks. *arXiv:1609.02907 [cs, stat]* (2017)
- [12] Krause, D., Thörnig, P.: JURECA: Modular supercomputer at Jülich Supercomputing Centre. *Journal of large-scale research facilities JLSRF* **4**, A132 (2018). <https://doi.org/10.17815/jlsrf-4-121-1>
- [13] Leprince, Y., Poupon, F., Delzescaux, T., Hasboun, D., Poupon, C., Rivière, D.: Combined Laplacian-equivolumic model for studying cortical lamination with ultra high field MRI (7 T). In: *IEEE International Symposium on Biomedical Imaging*. pp. 580–583 (2015). <https://doi.org/10.1109/ISBI.2015.7163940>

- [14] Lewiner, T., Lopes, H., Vieira, A.W., Tavares, G.: Efficient Implementation of Marching Cubes' Cases with Topological Guarantees. *Journal of Graphics Tools* **8**(2), 1–15 (2003). <https://doi.org/10.1080/10867651.2003.10487582>
- [15] Li, G., Muller, M., Thabet, A., Ghanem, B.: DeepGCNs: Can GCNs go as deep as CNNs? In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 9267–9276 (2019)
- [16] Li, G., Xiong, C., Thabet, A., Ghanem, B.: DeeperGCN: All You Need to Train Deeper GCNs. *arXiv:2006.07739 [cs, stat]* (2020)
- [17] Márquez-Neila, P., Baumela, L., Alvarez, L.: A Morphological Approach to Curvature-Based Evolution of Curves and Surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36**(1), 2–17 (2014). <https://doi.org/10.1109/TPAMI.2013.106>
- [18] Merker, B.: Silver staining of cell bodies by means of physical development. *Journal of Neuroscience Methods* **9**(3), 235–241 (1983). [https://doi.org/10.1016/0165-0270\(83\)90086-9](https://doi.org/10.1016/0165-0270(83)90086-9)
- [19] Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: *2016 Fourth International Conference on 3D Vision (3DV)*. pp. 565–571 (2016)
- [20] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: PyTorch: An Imperative Style, High-Performance Deep Learning Library. *arXiv:1912.01703 [cs, stat]* (2019)
- [21] Schiffer, C., Amunts, K., Harmeling, S., Dickscheid, T.: Contrastive Representation Learning for Whole Brain Cytoarchitectonic Mapping in Histological Human Brain Sections. In: *arXiv:2011.12865 [Cs, Eess, q-Bio]* (2021)
- [22] Schleicher, A., Amunts, K., Geyer, S., Morosan, P., Zilles, K.: Observer-independent method for microstructural parcellation of cerebral cortex: A quantitative approach to cytoarchitectonics. *NeuroImage* **9**(1), 165–177 (1999). <https://doi.org/10.1006/nimg.1998.0385>
- [23] Spitzer, H., Amunts, K., Harmeling, S., Dickscheid, T.: Parcellation of visual cortex on high-resolution histological brain sections using convolutional neural networks. In: *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*. pp. 920–923 (2017). <https://doi.org/10.1109/ISBI.2017.7950666>
- [24] Spitzer, H., Kiwitz, K., Amunts, K., Harmeling, S., Dickscheid, T.: Improving Cytoarchitectonic Segmentation of Human Brain Areas with Self-supervised Siamese Networks. In: *Medical Image Computing and Computer Assisted Intervention*. pp. 663–671 (2018)
- [25] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph Attention Networks. *arXiv:1710.10903 [cs, stat]* (2018)
- [26] You, Y., Gitman, I., Ginsburg, B.: Large Batch Training of Convolutional Networks. *arXiv:1708.03888 [cs]* (2017)