

Heidelberg University
BioQuant, IPMB
Biomedical Computer Vision Group (BMCV)

Seminar Deep Learning for Biomedical Image Analysis
Summer semester 2022
PD Dr. Karl Rohr

Mentor: Carola Krug

Report for the seminar presentation
BrainGNN: Interpretable Brain Graph
Neural Network for fMRI Analysis [11]

Student: Nils Krehl, 3664130,
Data and Computer Science
nils.krehl@stud.uni-heidelberg.de

Abstract

The area of neuroscience deals with the understanding of the brain. Especially interesting is the connection of different brain regions to specific neurological disorders or cognitive stimuli. In the work by Li et al. [11] a novel Brain Graph Neural Network is proposed to analyze the data obtained through fMRI imaging of the brain. The novelties of this work are a new graph convolutional layer and the interpretability of the model results. The proposed model outperforms all compared baseline architectures in the prediction task on two evaluation datasets. Furthermore, the insights gained through the model's interpretability could be verified by literature.

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 2 | Related Work | 1 |
| 3 | Approach | 2 |
| 3.1 | Experiment pipeline | 2 |
| 3.2 | Network architecture | 3 |
| 3.2.1 | Graph convolutional layer | 4 |
| 3.2.2 | Graph pooling layer | 6 |
| 3.3 | Training process: loss functions | 6 |
| 4 | Experiments and results | 8 |
| 5 | Discussion | 8 |
| 6 | Conclusion | 9 |
| 7 | References | 10 |

1 Introduction

The area of modern neuroscience deals with the understanding of the brain. From special interest is the connection of specific parts of the brain to neurological disorders or cognitive stimuli. Based on this, patterns of specific neurological diseases can be detected and a personalized treatment for patients can be developed.

This report summarizes the work by Li et al. [11] about "BrainGNN: Interpretable Brain Graph Neural Network for fMRI Analysis". The work by Li et al. is located in the area of modern neuroscience. Through fMRI (functional magnetic resonance imaging) the brain is observed. Based on these fMRI recordings, a graph of the brain is built. The nodes of the graph correspond to different brain regions and the edges describe interactions between different brain regions. This work then proposes a Brain Graph Neural Network (BrainGNN) for graph classification (e.g. is a person healthy or has a specific neurological disease?). The main novelties proposed of this work are an improved graph convolutional layer and the inbuilt model interpretability.

2 Related Work

The work by Li et al. [11] is the continuation of Li et al. [13]. In addition to [13] in [11] a new graph convolutional layer is proposed and a new dataset is used for evaluation.

In the recent past, Graph Neural Networks (GNN) have developed to the state-of-the-art method for graph based data science tasks. A GNN uses a graph (with node features, edge features, graph topology) as input for a neural network. GNNs are a generalization of Convolutional Neural Networks (CNN). The general applicability of GNNs was shown in various works, e.g. Kim and Ye [9], Kazi et al. [8], Yan et al. [23], Yang et al. [24], Gopinath et al. [6] and Nandakumar et al. [16].

Depending on the application domain of the GNNs the underlying graphs have different properties. For example in social networks or protein networks the nodes of different instances are not related. Therefore, in this case the same embeddings are used for different nodes. Also for the analysis of fMRI images GNNs with the same embedding for different nodes are used. This assumes, that the different parts of the brain behave similar. This is not the case. That is why the work by Li et al. [11] models the brain with different embeddings for each node.

For the detection of biomarkers, there have been studies for group-level

and individual-level biomarkers. Group-level biomarkers detect the common patterns of a specific disease and are explored by Li et al. [10], Venkataraman et al. [21], Salman et al. [18] and Yan et al. [23]. Individual-level biomarker are examined by Brennan et al. [2], Mahowald and Fedorenko [15] and Li et al. [12]. These biomarkers are specific for one patient and are the basis for precision medicine. The work by Li et al. [11] investigates both, group-level and individual-level biomarkers.

Another modeling question is, what the nodes and the whole graph represent. In Parisot et al. [17] and Kazi et al. [8] each patient is one node in the graph. The work by Li et al. [11] constructs one graph for the brain of each patient.

3 Approach

3.1 Experiment pipeline

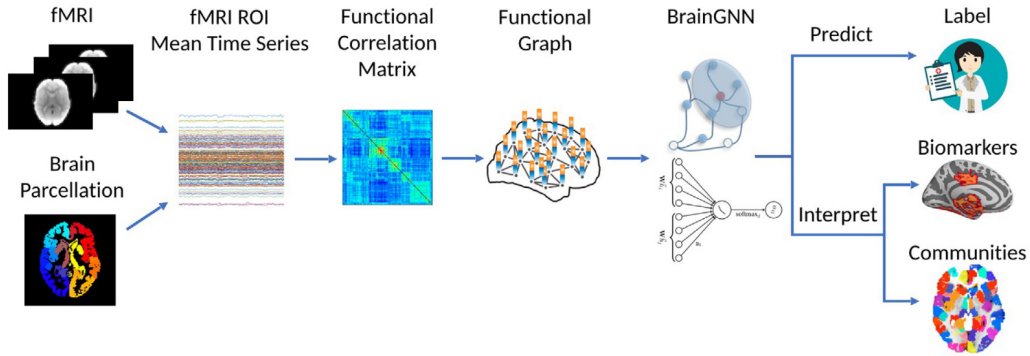


Figure 1: Pipeline [11]

Figure 1 illustrates the implemented experiment pipeline. The whole pipeline starts with a consecutive series of fMRI images. fMRI is an imaging technique, which measures blood flow changes. Due to the fact, that blood flow and neural activity are coupled, we can indirectly measure neural activity through fMRI imaging.

In the second step a brain parcellation atlas is used to cluster the brain into N areas. Each cluster is one region of interest (ROI).

For each ROI one time series is constructed. The mean value of the ROI from the first image is the first value of the time series, the mean ROI value of the second image the second time series value and so on. By doing this for all ROIs and all images, a high dimensional time series is received ($N =$

number of clusters from the brain parcellation atlas = number of ROIs = number of time series).

The functional correlation matrix is obtained through calculation of the correlation between all time series.

This matrix can be used as an adjacency matrix for construction of a graph. The column and row index values are the nodes. The cell values are the edge weights. The constructed graph is called functional graph and each node $V = \{v_1, \dots, v_N\}$ encodes one ROI (= one brain region) and the edges encode the coupling between the ROIs. The constructed graph is an undirected weighted graph.

To fully describe all features of this graph, two components are needed:

- **Topological structure:** $G = (V, \epsilon)$ defines the topological structure of the graph. V is the set of nodes as one hot encoding and ϵ is the edge list. The edge list contains two nodes, if they have a connecting edge. The entry (v_i, v_j) encodes a link from node v_i to node v_j . Furthermore the weight of each edge is defined by the correlation between the two connected nodes. The edge list can be represented as adjacency matrix E with $e_{ij} = \text{correlationvalue}$ if an edge connects v_i and v_j and $e_{ij} = 0$ if no edge exists between v_i and v_j .
- **Node features:** Each node has a feature vector. The node features for a graph can be represented as matrix $H = [h_1, \dots, h_N]$ with h_i being the feature vector of node v_i .

This functional graph is the input to the BrainGNN.

The results of the BrainGNN are on the one hand a prediction (e.g. whether a patient is healthy or not) and on the other hand the possibility to interpret the prediction results. BrainGNN makes it possible to detect biomarkers for the prediction tasks and also enables the detection of communities in the brain.

3.2 Network architecture

The network architecture is shown in figure 2. It consists of graph convolutional layers (Ra-GConv), pooling layers (R-pool) and a Multi Layer Perceptron (MLP). The network consists of two graph neural network (GNN) blocks followed by one MLP. One GNN block consists of one convolutional layer and one pooling layer. The results of these GNN blocks are finally concatenated and passed into a MLP for getting final predictions.

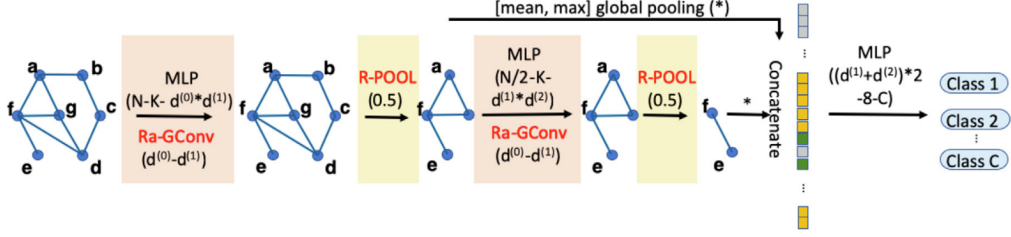


Figure 2: BrainGNN architecture [11]

3.2.1 Graph convolutional layer

The node feature vectors of the whole graph are stored in the H matrix. The goal of the convolutional layer is to learn new node feature vectors encoding the topological and semantical information of the graph.

The formula for the forward-pass update of the node feature vectors is defined by Schlichtkrull et al. [19] as:

$$\tilde{h}_i^{(l+1)} = \text{relu} \left(W_i^{(l)} h_i^{(l)} + \sum_{j \in \mathcal{N}^{(l)}(i)} e_{ij}^{(l)} W_j^{(l)} h_j^{(l)} \right). \quad (1)$$

The superscript (l) denotes the layer and the subscript i denotes the node. The feature vector of node i in layer $l+1$ is calculated by first multiplying the previous feature vector $h_i^{(l)}$ of node i with the weights $W_i^{(l)}$. Then the feature vectors of all neighbors of node i (neighbor = an edge exists between node i and node j) are multiplied with the weights and summed. Up to now the formula is linear. To achieve nonlinearity, the result is passed through an activation function (in this case relu).

The general idea of the convolutional layer and the novelties of this work are illustrated in figure 3.

The two proposed improvements in this work [11] for the graph convolutional layer are:

1. Learn different embedding weights for each ROI:

In eq 1 the weights W are the same for all nodes (ROIs). In this work individual weights are learned for each node. Finally, the vector $\text{vec}(W_i^{(l)})$ contains all learned node weights.

The underlying assumption is, that the nodes at the same location but from different graphs are similar. Furthermore, a vector r_i contains the regional information of node i (e.g. the position in the graph). The ordering of all nodes (ROIs) is consistent across different instances. To

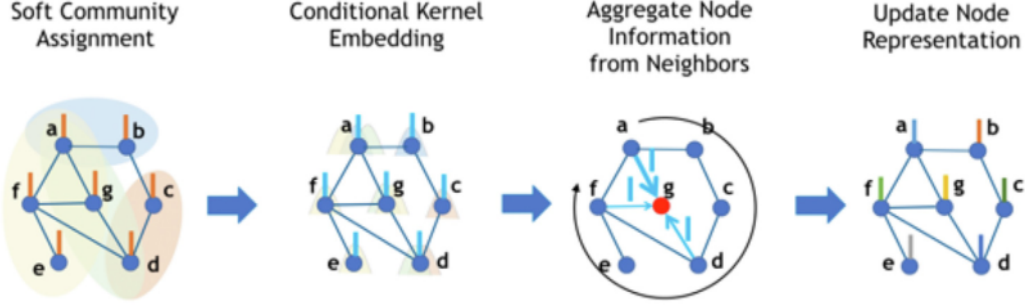


Figure 3: Graph convolutional layer operations [11]

be able to learn new node feature vectors, which are independent of the ordering method, the regional information r_i is stored as one-hot encoded vector.

As described above the new node feature vector is calculated by aggregating the neighboring node feature vectors. For each node a computation graph of the neighbors is set up. The aggregation of node feature vectors is then done by a neural network with the parameters $\{\Theta_1^{(l)}, \Theta_2^{(l)}\}$ and the input r_i .

$$\text{vec}(W_i^{(l)}) = f_{MLP}^{(l)}(r_i) = \Theta_2^{(l)} \text{relu}(\Theta_1^{(l)} r_i) + b^{(l)}. \quad (2)$$

Eq 2 can be reformulated to:

$$\text{vec}(W_i^{(l)}) = \sum_{u=1}^{K^{(l)}} (\alpha_{iu}^{(l)} + \beta_u^{(l)}) + b^{(l)}. \quad (3)$$

The weights are decomposed into α (corresponds to $\Theta_1^{(l)}$) and β (corresponds to $\Theta_2^{(l)}$). The matrix α contains information about the community membership of nodes. A basis vector for each community is stored in β .

2. Include the edge weight:

The edge weights in the graph describe the coupling between two nodes. By multiplying node features with the edge weight the connection strength is taken into account.

3.2.2 Graph pooling layer

The goal of the graph pooling layer is the dimensionality reduction of the graph. Kaiser et al. [7] and Baker et al. [1] have shown, that not all nodes are equally important for predicting neurological disorders. A dimensionality reduction is achieved by keeping the important nodes and removing the other nodes. The pooling layer is implemented as proposed by Cangea et al. [3] and Gao and Ji [4]. For details, please refer to [3] and [4].

The general idea of the pooling layer is illustrated in figure 4. The input are the feature vectors from the convolutional layer. These vectors are then projected onto a learned pooling vector $w^{(l)}$. Based on the difference between the pooling vector and the node feature vectors, a score is calculated. Low scoring nodes are removed, and high scoring nodes are kept.

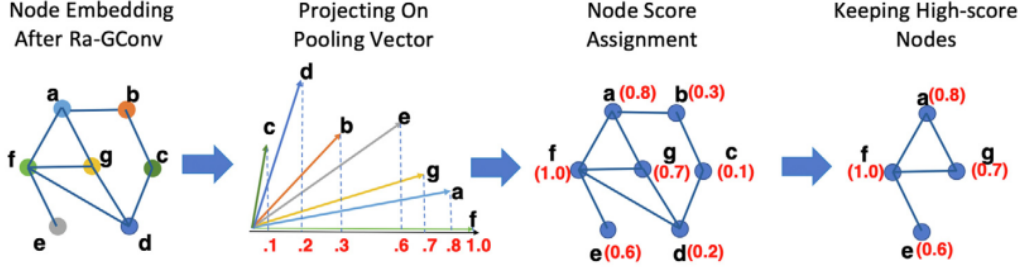


Figure 4: Graph pooling layer operations [11]

3.3 Training process: loss functions

As classification loss, the cross entropy loss is used:

$$L_{ce} = -\frac{1}{M} \sum_{m=1}^M \sum_{c=1}^C y_{m,c} \log(\hat{y}_{m,c}). \quad (4)$$

M denotes the number of instances, C the number of classes, $\hat{y}_{m,c}$ is the model prediction and $y_{m,c}$ is the ground truth label.

To enable interpretability further loss terms are used. As introduced in section 3.2.2 a projection from the node representation onto a learned vector $w^{(l)} \in \mathbb{R}^{d^{(l)}}$ is used. The scaling of this vector can vary, but the resulting scores remain the same. Consequently, it is not uniquely identifiable, which scores are produced by which learned vectors. To overcome this issue, a unit loss enforces, that $w^{(l)}$ is a unit vector.

$$L_{unit}^{(l)} = (\|w^{(l)}\|_2 - 1)^2. \quad (5)$$

A group-level consistency (GLC) loss is used to ensure, that in the pooling layer, the same nodes are selected across all input instances. This enables the detection of underlying patterns of a disease. In deeper layers of the network, the graph structure changes. That is why this loss is only used after the first pooling layer, because here the graph structure is still the same as the input graph.

The GLC loss is defined as:

$$L_{GLC} = \sum_{c=1}^C \sum_{m,n \in \mathcal{I}_c} \|\tilde{s}_m^{(1)} - \tilde{s}_n^{(1)}\|^2 = 2 \sum_{c=1}^C Tr((S_c^{(1)})^T L_c S_c^{(1)}). \quad (6)$$

It calculates the pooling score similarity between two instances m and n ($\|\tilde{s}_m^{(1)} - \tilde{s}_n^{(1)}\|^2$). By taking the sum over all classes C (first sum) and the sum over all instances M in one batch (second sum), the pairwise pooling score similarities are calculated between all instances. The way how this is calculated is expressed on the right hand side of the equation. For details about this, please refer to [11].

The TopK pooling loss makes sure, that the nodes which are removed have a significantly different score compared to the nodes which are not removed. First of all, the scores are passed through a sigmoid layer. After this, the score values are in the range between 0 and 1. These scores are then sorted in descending order for all M instances. Finally, a constraint is applied to the ordering for all M instances, which leads to a more dispersed distribution. Mathematically, it is formulated as binary cross-entropy:

$$L_{TPK}^{(l)} = -\frac{1}{M} \sum_{m=1}^M \frac{1}{N^{(l)}} \left(\sum_{i=1}^k \log(\hat{s}_{m,i}^{(l)}) + \sum_{i=1}^{N^{(l)}-k} \log(1 - \hat{s}_{m,i+k}^{(l)}) \right). \quad (7)$$

All presented loss functions above are part of the final loss function (with λ_1 and λ_2 as hyperparameters):

$$L_{total} = L_{ce} + \sum_{l=1}^L L_{unit}^{(l)} + \lambda_1 \sum_{l=1}^L L_{TPK}^{(l)} + \lambda_2 L_{GLC}. \quad (8)$$

4 Experiments and results

The proposed BrainGNN is evaluated on two datasets. On the one hand the Biopoint Autism Study Dataset (Biopoint) [21] is used. It deals with a binary prediction task, whether a patient has autism or not. On the other hand the Human Connectome Project (HCP) 900 Subject Release dataset [20] is used. The task is a multi class classification, to distinguish between different tasks (e.g. motorical task, language task, gambling). For details about the datasets, please refer to [21] and [20].

Table 5 shows the prediction results of BrainGNN in comparison to other approaches. For both datasets, BrainGNN outperforms all other approaches.

| | | SVM | Random Forest | MLP | BrainNetCNN | GAT | GraphSAGE | PR-GNN | BrainGNN |
|----------|----------------------------|--------------------------|---------------|--------------|--------------|--------------|--------------|--------------|--------------------------------|
| Biopoint | Accuracy (%) | 62.80(4.92) ^a | 68.60(3.58) | 58.80(1.79) | 75.20(3.49) | 77.40(3.51) | 78.60(5.90) | 77.10(8.71) | 79.80(3.63)^c |
| | F1 (%) | 60.08(3.91) | 63.97(4.95) | 55.25(9.49) | 65.58(14.48) | 75.08(5.19) | 75.55(7.03) | 75.20(7.01) | 75.80(6.03) |
| | Recall (%) | 60.20(4.49) | 71.11(8.12) | 61.00(4.85) | 66.20(10.85) | 71.60(6.07) | 75.20(6.46) | 78.26(10.28) | 72.60(5.64) |
| | Precision (%) | 60.00(3.81) | 67.80(5.36) | 53.40(12.52) | 65.60(17.95) | 79.40(8.02) | 76.20(8.11) | 76.50(14.32) | 79.60(8.59) |
| | Parameter (k) ^b | 3 | 3 | 138 | 1438 | 16 | 6 | 6 | 41 |
| HCP | Accuracy (%) | 90.00(8.20) | 90.20(4.15) | 67.20(34.40) | 90.60(4.04) | 78.60(10.45) | 89.80(12.51) | 91.20(8.28) | 94.40(4.04)^d |
| | F1 (%) | 90.20(5.81) | 90.14(5.55) | 63.49(41.80) | 90.96(3.50) | 77.00(11.58) | 88.60(13.19) | 91.09(8.35) | 94.34(3.27)^d |
| | Recall (%) | 89.57(8.04) | 90.06(7.35) | 67.97(41.66) | 91.12(4.13) | 78.60(10.45) | 89.43(12.43) | 91.00(8.95) | 94.29(3.73)^d |
| | Precision (%) | 90.85(9.35) | 90.22(4.77) | 62.97(42.47) | 90.81(3.27) | 91.20(3.32) | 87.80(14.02) | 91.14(8.52) | 94.40(3.59)^d |
| | Parameter (k) | 36 | 36 | 713 | 4547 | 34 | 12 | 12 | 96 |

^a Classification accuracy, f1-score, recall and precision of the testing sets are reported in mean (standard deviation) format. ^b The number of trainable parameters of each model is denoted. ^c We boldfaced the results generated from our proposed BrainGNN. ^d * indicates significantly outperforming ($p < 0.001$ under one tail two-sample t-test) all the alternative methods.

Figure 5: Prediction results of BrainGNN compared to other baseline approaches

[11]

Besides the classification, the interpretation of the results is also possible. On the one hand, the detection of biomarkers is possible. For both datasets the detected biomarkers are validated by literature. On the other hand the detected brain community patterns can be interpreted. For a detailed description of the interpretation process, please refer to [11].

5 Discussion

The preprocessing steps from Li et al. [11] (all steps from fMRI imaging up to the construction of the functional graph) are one possible way, how this could be done. In their work, they have used one brain parcellation atlas. For future work the usage of different brain parcellation atlases and the comparison of the results would be interesting. Furthermore, the authors regard an end-to-end training procedure, which extracts relevant features automatically from the fMRI imaging as part of the trained model as a challenging but promising direction for future work.

The BrainGNN model can be further improved through a more detailed hyperparameter tuning. Moreover, more quantitative and theoretical studies are needed to fully understand the proposed graph convolutional layer.

The results of BrainGNN can be improved by evaluating failure prediction cases to answer the question, if there are specific patterns, where the model always fails. Finally, the whole proposed model could be tested on other datasets and problems.

The work from Li et al. [11] was used in subsequent work as baseline for comparing results and as starting point for model architecture improvements. Lin et al. [14] use the proposed BrainGNN from Li et al. [11] as a baseline for comparing their 3D-CNN architecture. As evaluation dataset they used a schizophrenia dataset. Gao et al. [5] built their work upon Li et al. [11] and apply the GNN to sex prediction and interpretation of the differences related to sex. Wen et al. [22] propose an optimized GNN architecture and use the BrainGNN from Li et al. [11] among other approaches as baseline.

6 Conclusion

The work by Li et al. [11] proposes a novel Brain Graph Neural Network for fMRI analysis. The two main innovations are the improved graph convolutional layer and the inbuilt interpretability through various loss terms as part of the graph pooling layer.

The result of Li et al. [11] is a BrainGNN model, which outperforms all other approaches for both prediction tasks. Furthermore the detection of group-level and individual-level biomarkers is possible. Moreover it is possible to detect brain communities.

Through this work the understanding of neurological diseases can be improved, which is essential for the development of personalized treatments in the area of precision medicine.

7 References

- [1] Justin T. Baker et al. “Disruption of Cortical Association Networks in Schizophrenia and Psychotic Bipolar Disorder”. In: *JAMA Psychiatry* 71.2 (Feb. 2014). ISSN: 2168-622X. DOI: 10.1001/jamapsychiatry.2013.3469.

- [2] Brian P. Brennan et al. “Use of an Individual-Level Approach to Identify Cortical Connectivity Biomarkers in Obsessive-Compulsive Disorder”. In: *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging* 4.1 (2019), pp. 27–38. ISSN: 2451-9022. DOI: 10.1016/j.bpsc.2018.07.014.
- [3] Cătălina Cangea et al. *Towards Sparse Hierarchical Graph Classifiers*. 2018. DOI: 10.48550/ARXIV.1811.01287.
- [4] Hongyang Gao and Shuiwang Ji. “Graph U-nets”. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, June 9–15, 2019, pp. 2083–2092.
- [5] Yang Gao et al. “Sex Differences of Cerebellum and Cerebrum: Evidence from Graph Convolutional Network”. In: *Interdisciplinary Sciences: Computational Life Sciences* 14.2 (June 1, 2022), pp. 532–544. ISSN: 1867-1462. DOI: 10.1007/s12539-021-00498-5.
- [6] Karthik Gopinath, Christian Desrosiers, and Herve Lombaert. “Adaptive Graph Convolution Pooling for Brain Surface Analysis”. In: *Information Processing in Medical Imaging*. Ed. by Albert C. S. Chung et al. Cham: Springer International Publishing, 2019, pp. 86–98. ISBN: 978-3-030-20351-1.
- [7] Martha D. Kaiser et al. “Neural Signatures of Autism”. In: *Proceedings of the National Academy of Sciences* 107.49 (2010), pp. 21223–21228. DOI: 10.1073/pnas.1010412107. eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.1010412107>.
- [8] Anees Kazi et al. “InceptionGCN: Receptive Field Aware Graph Convolutional Network for Disease Prediction”. In: *Information Processing in Medical Imaging*. Ed. by Albert C. S. Chung et al. Cham: Springer International Publishing, 2019, pp. 73–85. ISBN: 978-3-030-20351-1.
- [9] Byung-Hoon Kim and Jong Chul Ye. “Understanding Graph Isomorphism Network for Rs-fMRI Functional Connectivity Analysis”. In: *Frontiers in Neuroscience* 14 (2020). ISSN: 1662-453X. DOI: 10.3389/fnins.2020.00630.
- [10] Xiaoxiao Li et al. “Brain Biomarker Interpretation in ASD Using Deep Learning and fMRI”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. Ed. by Alejandro F. Frangi et al. Cham: Springer International Publishing, 2018, pp. 206–214. ISBN: 978-3-030-00931-1.

- [11] Xiaoxiao Li et al. “BrainGNN: Interpretable Brain Graph Neural Network for fMRI Analysis”. In: *Medical Image Analysis* 74 (2021), p. 102233. ISSN: 1361-8415. DOI: 10.1016/j.media.2021.102233.
- [12] Xiaoxiao Li et al. “Graph Neural Network for Interpreting Task-fMRI Biomarkers”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*. Ed. by Dinggang Shen et al. Cham: Springer International Publishing, 2019, pp. 485–493. ISBN: 978-3-030-32254-0.
- [13] Xiaoxiao Li et al. “Pooling Regularized Graph Neural Network for FMRI Biomarker Analysis”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part 7*. Berlin, Heidelberg: Springer-Verlag, 2020, pp. 625–635. ISBN: 978-3-030-59727-6. DOI: 10.1007/978-3-030-59728-3_61.
- [14] Qiu-Hua Lin et al. “SSPNet: An Interpretable 3D-CNN for Classification of Schizophrenia Using Phase Maps of Resting-State Complex-Valued fMRI Data”. In: *Medical Image Analysis* 79 (2022), p. 102430. ISSN: 1361-8415. DOI: 10.1016/j.media.2022.102430.
- [15] Kyle Mahowald and Evelina Fedorenko. “Reliable Individual-Level Neural Markers of High-Level Language Processing: A Necessary Precursor for Relating Neural Variability to Behavioral and Genetic Variability”. In: *NeuroImage* 139 (2016), pp. 74–93. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2016.05.073.
- [16] Naresh Nandakumar et al. “A Novel Graph Neural Network to Localize Eloquent Cortex in Brain Tumor Patients from Resting-State fMRI Connectivity”. In: *Connectomics in NeuroImaging*. Ed. by Markus D. Schirmer et al. Cham: Springer International Publishing, 2019, pp. 10–20. ISBN: 978-3-030-32391-2.
- [17] Sarah Parisot et al. “Disease Prediction Using Graph Convolutional Networks: Application to Autism Spectrum Disorder and Alzheimer’s Disease”. In: *Medical Image Analysis* 48 (2018), pp. 117–130. ISSN: 1361-8415. DOI: 10.1016/j.media.2018.06.001.
- [18] Mustafa S. Salman et al. “Group ICA for Identifying Biomarkers in Schizophrenia: ‘Adaptive’ Networks via Spatially Constrained ICA Show More Sensitivity to Group Differences than Spatio-Temporal Regression”. In: *NeuroImage: Clinical* 22 (2019), p. 101747. ISSN: 2213-1582. DOI: 10.1016/j.nicl.2019.101747.

- [19] Michael Schlichtkrull et al. “Modeling Relational Data with Graph Convolutional Networks”. In: *The Semantic Web*. Ed. by Aldo Gangemi et al. Cham: Springer International Publishing, 2018, pp. 593–607. ISBN: 978-3-319-93417-4.
- [20] David C. Van Essen et al. “The WU-Minn Human Connectome Project: An Overview”. In: *NeuroImage* 80 (2013), pp. 62–79. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2013.05.041.
- [21] Archana Venkataraman et al. “Bayesian Community Detection in the Space of Group-Level Functional Differences”. In: *IEEE Transactions on Medical Imaging* 35 (2016), pp. 1866–1882.
- [22] Guangqi Wen et al. “MVS-GCN: A Prior Brain Structure Learning-Guided Multi-View Graph Convolution Network for Autism Spectrum Disorder Diagnosis”. In: *Computers in Biology and Medicine* 142 (2022), p. 105239. ISSN: 0010-4825. DOI: 10.1016/j.combiomed.2022.105239.
- [23] Yujun Yan et al. “GroupINN: Grouping-based Interpretable Neural Network for Classification of Limited, Noisy Brain Data”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery; Data Mining*. KDD ’19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 772–782. ISBN: 978-1-4503-6201-6. DOI: 10.1145/3292500.3330921.
- [24] Huzheng Yang et al. “Interpretable Multimodality Embedding of Cerebral Cortex Using Attention Graph Network for Identifying Bipolar Disorder”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*. Ed. by Dinggang Shen et al. Cham: Springer International Publishing, 2019, pp. 799–807. ISBN: 978-3-030-32248-9.