



HAL
open science

Structure-Function Mapping via Graph Neural Networks

Yang Ji, Samuel Deslauriers-Gauthier, Rachid Deriche

► **To cite this version:**

Yang Ji, Samuel Deslauriers-Gauthier, Rachid Deriche. Structure-Function Mapping via Graph Neural Networks. MLCN 2021 - 4th International Workshop on Machine Learning in Clinical Neuroimaging, Sep 2021, Strasbourg, France. pp.135-144, 10.1007/978-3-030-87586-2_14 . hal-03352550

HAL Id: hal-03352550

<https://hal.science/hal-03352550>

Submitted on 23 Sep 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Structure-Function Mapping via Graph Neural Networks

Yang Ji^{1,2}[0000-0002-1300-4292],
Samuel Deslauriers-Gauthier^{1,3}[0000-0003-2781-121X], and
Rachid Deriche^{1,4}[0000-0002-4643-8417]

¹ Université Côte d’Azur, Inria, France

² Yang.Ji@inria.fr

³ Samuel.Deslauriers-gauthier@inria.fr

⁴ Rachid.Deriche@inria.fr

Abstract. Understanding the mapping between structural and functional brain connectivity is essential for understanding how cognitive processes emerge from their morphological substrates. Many studies have investigated the problem from an eigendecomposition viewpoint, however, few have taken a deep learning viewpoint, even less studies have been engaged within the framework of graph neural networks (GNNs). As deep learning has produced significant results in several fields, there has been an increasing interest in applying neural networks to graph problems. In this paper, we investigate the structural connectivity and functional connectivity mapping within a deep learning GNNs based framework, including graph convolutional networks (GCN) and graph transformer networks (GTN). To our knowledge, this original GTN based framework has never been studied in the context of structure-function and brain connectivity mapping. To achieve this goal, we use a GNNs based encoder-decoder system, where the encoder takes structural connectivity (SC) matrix as input and generates a latent representation of each node in a lower dimension, then the decoder uses the latent representation to reconstruct or predict the associated functional connectivity (FC) matrix. Besides comparing different encoders for node embedding, we also demonstrate that a decoder, which projects lower dimension vectors onto higher dimensional space, can improve the model performance. Our experiments demonstrate that both GCN encoder and GTN encoder combined with the proposed decoder can provide better results on our data than the previously proposed GCN autoencoder model. GTN encoder is also shown to be much more effective when it comes to noisy data and outliers.

Keywords: Brain connectivity mapping · Graph neural network · Graph convolutional network · Graph transformer network · Autoencoder.

1 Introduction

Studying the relationship between structural connectivity and functional connectivity is essential to understand how brain function emerges from their un-

derlying structural substrate. A better characterization of the link between SC and FC will provide insights into how lesions to the structural substrate affect brain function [1–3], with great potential clinical applications. Previous studies have shown that functional connectivity and structural connectivity patterns are correlated [4–6]. Therefore, it is reasonable to expect that FC matrices can be predicted from SC matrices.

Many models have been proposed to map structure to function [9–13]. A subset of these models are specifically based on the link between the eigenvalues and eigenvectors of SC and FC matrices. However, this approach is based on the assumption that there exists a correspondence between structural and functional eigenmodes that can be captured by the models. Moreover, they are usually computationally expensive when dealing with large data. In contrast, deep learning (DL) approaches provide a more general approach to learn the mapping between SC and FC. Deep learning has produced significant results in several fields, including computer vision, speech recognition and natural language processing [7]. To integrate non-Euclidean graph data, graph neural networks (GNNs) have been developed to deal with graph problems [8]. For example, graph convolutional networks (GCN), which were proposed by [14], are motivated by the graph convolution operator [15]. At a higher level, GCNs use the graph Fourier transform to aggregate neighboring nodes’ features, thus exploiting the topology of the graph during the learning process. GCNs have been studied on structure-function mapping in [16], where a graph encoder-decoder based on GCNs was proposed to recover the SC-FC mapping. However, the decoder in this model is mainly an inner product operation with no trainable parameters. The linear encoder used by the model can also be sensitive to outliers presented in data. A second example are graph transformer networks (GTN) [17] are analogous to Spatial Transformer Networks [18]. The most advantageous point of GTNs over GCNs is that they consider heterogeneous graph structures, and can therefore be used to predict FC from several different SC matrices. This feature makes GTN more robust when dealing with outliers or noisy data, where GCN is much less effective.

In this work, we revisit structure-function mapping via deep graph neural networks. First, we investigate the importance of a trainable decoder in the GCN architecture of [16] and show that it improves the prediction of FC from SC. Second, we propose a GTN architecture to predict FC from several diffusion derived SC matrices and show that is more robust to outliers. To our knowledge, this is the first time GTNs have been studied in the context of structure-function and brain connectivity mapping. All of the models are tested on 1050 subjects of the Human Connectome Project, therefore providing reproducible results.

The rest of the paper is composed of three parts. The first part is Preliminaries, where we state the main problem and then introduce the mathematical theories of GCN, GTN and encoder-decoder system. Then we propose numerous models based on GNNs. The second part is Experiments, where we present how SC/FC matrices are extracted from Magnetic resonance imaging(MRI), and de-

tails of the implementation. The final part is Results and conclusion, where we compare and discuss in detail the performance of each proposed model.

2 Preliminaries

In this section, we first state the problem and the objective of this paper and review some GNNs architectures including GCN, GTN, and the main framework Autoencoder that includes them all. At the end of each architecture description, a model is proposed.

2.1 Problem statement

Given an undirected SC matrix, which is extracted from diffusion MRI, we define $A \in \mathbb{R}^{N \times N}$ as its symmetric adjacency matrix, $D_{ii} = \sum_{j=0}^N A_{ij}$ as the diagonal degree matrix which denotes the number of connections of each node, N as the number of nodes and in our case it is the number of regions of the atlas. The normalized Laplacian of A is defined as $L = I_N - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = U \Lambda U^T$, where Λ denotes its eigenvalues and U as its eigenvector. The brain connectivity mapping is between the SC matrix and the FC matrix. The objective is to identify a mapping that takes A as input and predicts a FC matrix $Z \in \mathbb{R}^{N \times N}$, such that Z is close to the empirical FC matrix. The problem can be reformulated as

$$\text{minimize } \frac{1}{K} \sum_{k=1}^K \|f(A_k) - Z_k\|_F^2 \quad (1)$$

where f is the mapping function (in our case a DL architecture), K is the number of subjects, and $\|\cdot\|_F$ is the Frobenius norm.

2.2 Autoencoder

The idea of autoencoder [19] is to use an encoder to learn a latent representation of the data, then use a decoder to reconstruct its input. Here, we use the autoencoder as our main structure, but we alter it into reconstructing the associated FC matrix instead of SC matrix. An encoder generates a latent vector representation y_i for each node, then a decoder uses these representation vectors to compute the prediction of the FC matrix, denoted as $Z' = f'(Y)$ with $Y \in \mathbb{R}^{N \times F}$ where F is the dimension of the latent node representation. In this work, we test both GCN (as in [16]) and GTN as encoders. A diagram of the model is presented in Figure 1.

The decoder mainly serves to transform latent nodes embedding Y to target symmetric FC prediction Z . In [16], the authors proposed a decoder based on inner product, defined as:

$$Z' = \tanh(\text{ReLU}(YY^T)) \quad (2)$$

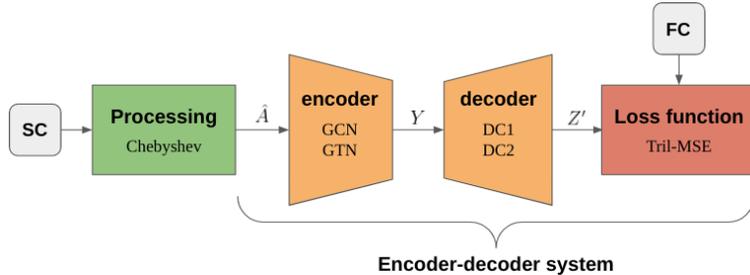


Fig. 1: Model pipeline. The first step is to process SC matrix to obtain \hat{A} . Then the encoder (GCN or GTN) takes \hat{A} as input and generate embedding Y . Finally, the decoder (DC1 or DC2) takes Y as input and reconstruct FC matrix Z' . A loss function (Tril-MSE) calculates the error between Z' and the empirical FC.

where $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ and $\text{ReLU}(x) = \max(0, x)$ are activation functions. Z' is a symmetric matrix whose values are in the range $[0, 1]$, like the observed FC matrix. However, apart from reconstructing a prediction that fits the FC matrix' symmetry and range property, it is not trainable. In the rest of the article, this decoder is referred to as **DC1**.

To overcome the disadvantage of DC1, we propose a decoder that is trainable, defined as

$$\begin{aligned} Y' &= YW_0^d + B^d, \\ Y'' &= \tanh(Y' * W_1^d), \\ Z' &= \text{ReLU}(Y'' Y''^T) \end{aligned} \quad (3)$$

where $W_0^d \in \mathbb{R}^{F \times N}$ is a trainable parameter which augments the dimension of the latent embedding Y , $B^d \in \mathbb{R}^{N \times N}$ is a bias parameter, $W_1^d \in \mathbb{R}^{N \times 1}$ is also a trainable parameter, initialized with constant 1 to keep the output's range close to $[0, 1]$ from the start. It balances the dynamic of the inner product operation. The $*$ operator denotes a row-wise multiplication. To predict a semidefinite positive FC matrices, for example when FC matrices are not thresholded, it suffices to omit $\text{ReLU}(\cdot)$. In the rest of the article, this decoder is referred to as **DC2**.

2.3 Graph Convolutional Networks (GCN)

A spectral graph convolution operator, as defined by [21], is given by

$$g_\theta \star x = U g_\theta U^T x \quad (4)$$

with the filter $g_\theta = \text{diag}(\theta)$, $\theta \in \mathbb{R}^N$ is the parameter. Therefore g_θ can be considered as a function of the eigenvalues of the Laplacian \mathcal{L} . In order to avoid expensive computation of eigendecomposition of graph Laplacian L , [20] proposed using Chebyshev polynomials to approximate $g_\theta(\mathcal{L})$. Given the recursively defined

Chebyshev polynomial as $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$, with $T_0 = 1, T_1 = x$, $g_\theta(\Lambda)$ is reformulated as

$$g_{\theta'}(\Lambda) \approx \sum_{k=0}^K \theta'_k T_k(\tilde{\Lambda}) \quad (5)$$

where $\tilde{\Lambda} = \frac{2}{\lambda_{max}}\Lambda - I_N$, λ_{max} is the largest eigenvalue of L . $\theta' \in \mathbb{R}^K$ is the coefficient vector of Chebyshev polynomial. Combining equations (4) and (5), a new graph convolution operator can be defined as

$$g_{\theta'} \star x \approx \sum_{k=0}^K \theta'_k T_k(\tilde{L})x \quad (6)$$

where $\tilde{L} = \frac{2}{\lambda_{max}}L - I_N$.

In [14], a simple graph convolution operator is proposed, which is motivated by a first-order approximation of Chebyshev polynomials [20]. By limiting $K = 1$ and setting $\lambda_{max} = 2, \theta'_0 = -\theta'_1 = \theta$ in equation (6), we have the graph convolution operator that is motivated by the first order of Chebyshev

$$g_{\theta'} \star x \approx \theta(I_N + D^{-\frac{1}{2}}AD^{-\frac{1}{2}})x = \theta\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}x \quad (7)$$

where $\tilde{A} = A + I_N$ is a symmetric self-looped adjacency matrix, $\tilde{D}_{ii} = \sum_{j=0}^N \tilde{A}_{ij}$ denotes the degree matrix of the \tilde{A} . Finally the first order approximation GCN is defined as

$$Y^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}Y^{(l)}W_0^{(l)}\right), \quad (8)$$

where $Y^0 = X$, $Y^{(l)}$ denotes the l th hidden layer, $W_0^{(l)}$ is the weight parameter of l th layer. $\sigma(\cdot)$ is an activation function.

In our implementation, the GCN encoder is a single layer GCN defined as

$$Y = \text{ReLU}(\hat{A}XW_0), \quad (9)$$

where $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$, \tilde{A} was defined in equation (7), $W_0 \in \mathbb{R}^{N \times F}$ is a trainable parameter of the encoder, N and F are previously defined.

2.4 Graph Transformer Networks(GTN)

Graph Transformer Networks [17] take heterogeneous graphs as multi-channel input and use these channels to compute multi-channel meta-path graph tensors. Then, GTNs apply a single layer GCN to each channel of the meta-path tensor and generate a node representation for each channel. Finally, a perceptron layer is added on top of the concatenation of the node representations of each channel.

Consider $\mathbb{A} \in \mathbb{R}^{N \times N \times C}$ a set of adjacency matrices of heterogeneous graphs, C denotes channel number, which in our case $\mathbb{A} = [\hat{A}_1, \hat{A}_2, \hat{A}_3]$ is composed by the first order approximation of three types of SC matrices, whose nature is detailed

in Section 3.1. First the Graph Transformer (GT) layer generates two intermediate adjacency matrices, $\mathbb{Q}_1 \in \mathbb{R}^{N \times N \times C}$ and $\mathbb{Q}_2 \in \mathbb{R}^{N \times N \times C}$, then performs a 1x1 convolution as

$$\mathbb{Q}_i = \phi(\mathbb{A}, \text{softmax}(W^\phi)), \quad (10)$$

where ϕ is a graph convolution, $W^\phi \in \mathbb{R}^{1 \times 1 \times K}$ is the parameter of ϕ , $\text{softmax}(\cdot)$ is an activation function defined as $\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$. Second, the GT layer combines the intermediate adjacency matrices \mathbb{Q}_i together to compute a normalized new graph structure. For the $A^{(1)} = \mathbb{D}_{(1)}^{-1} \mathbb{Q}_1 \mathbb{Q}_2$, $\mathbb{D} \in \mathbb{R}^{N \times N \times C}$ is the multi channels degree tensor of $\mathbb{Q}_1 \mathbb{Q}_2$. For multi layers GTN, we repeat the procedure from Equation (10) by generating \mathbb{Q}_{l+1} , then we compute the new graph $A^{(l)} = \mathbb{D}_{(l)}^{-1} A^{(l-1)} \mathbb{Q}^{(l+1)}$. Finally, a graph convolution is performed on each channel of the new structure $A^{(l)}$ before combining all channels' node embedding. Instead of concatenating all channels' embedding as in the original paper [17], we found that summing them performs better. Thus the intermediate output is

$$Y' = \sum_{i=1}^C \sigma \left(\tilde{D}_i^{-1} \tilde{A}_i^{(1)} X W_0 \right) \quad (11)$$

where $\tilde{A}_i^{(l)} = A_i^{(l)} + I$. C denotes the number of channels, $W_0 \in \mathbb{R}^{N \times N}$ is a trainable weight parameter. In our implementation of GTN, we set the layer number to $l = 1$ and the channel number to $C = 3$. Upon the output defined in Equation (11) in section 2.4, an additional layer is added in order to obtain lower dimension embedding. The ultimate GTN encoder is defined as

$$Y = \text{ReLU} \left(Y' W_1 + B_1 \right) \quad (12)$$

where $W_1 \in \mathbb{R}^{N \times F}$ is a trainable parameter, $B_1 \in \mathbb{R}^{F \times F}$ denotes bias.

3 Experiments

3.1 Data

The SC and FC dataset, which contain 1050 subjects, is provided by Human Connectome Project ⁵ (HCP). The SC and FC data are extracted from diffusion MRI (dMRI) and functional MRI (fMRI) using the same pipelines as in [27]. We used the streamlines produced by the diffusion pipeline to compute three types of SC matrices, *Count*, *SIFT2*, *Length*. The first one *Count*(i, j) represents the total number of streamlines between regions i and j . The second one *SIFT2*(i, j) was computed by summing the SIFT2 weights of streamlines connecting i and j [28]. The third one *Length*(i, j), denotes the reciprocal average length of the streamlines connecting i and j . Both SC and FC adjacency matrix contain $N = 68$ rows and columns, each row and column corresponds to a brain region of the Desikan-Killiany atlas.

⁵ <http://www.humanconnectomeproject.org/>

The SC matrix A extracted from dMRI is not directly used as model input. First, we set the diagonal values of A to be zero. Then, as explained in Section 2.3, we use first order approximation of Chebyshev polynomial on A to obtain the input \hat{A} , defined in Equation (7). As $Length$ data has infinite values (between disconnected brain regions), for computational stability, we set its diagonal values as 1, then compute $1/Length(i, j)$ beforehand. The FC originally ranges between -1 and 1, but in our implementation, the negative values in FC are set to 0 as in previous researches dealing with functional brain connectivity [25, 26]. After our processing pipeline, the average correlation between the FC matrix and the $Count$, $Length$, and $SIFT2$ matrices is 0.24, 0.28, and 0.24, respectively.

3.2 Implementation

The GCN uses only the $Count$ matrix, while GTN uses all three SC matrices mentioned previously. The dataset is split into three parts: 80% for training, 10% for validation and 10% for test. Cross-validation (10 folds) and early stop are applied. We use Adam as optimizer [22], set learning rate to 0.001; batch to 25 for GCN, 1 for GTN; dropout rate to 0.05; encoder’s output dimension to $F = 32$ for all models. The weight parameters are initialized using [23]. The models are implemented via Pytorch [24]. We use Mean Square Error(MSE) as all models’ loss function, however, it is only applied on the lower triangular part of the predicted FC matrix, since FC matrix is symmetric and whose diagonal values are always 1. We reference it as **Tril-MSE** in the following. We also use Pearson Correlation as a supplementary measure to double evaluate the quality of predictions, although we do not train our models on it. Same as Tril-MSE, we apply it only on the lower triangular part of output matrices, we reference it as **TPearson** for the rest of the paper.

4 Results and discussion

We evaluate the average and median Tril-MSE test error, as well as the average and median TPearson between predicted FC and empirical FC. It is also important to evaluate the median test error because outliers are often present in data and difficult to identify beforehand. A model with a lower median test error is more robust when it comes to a dataset with outliers, an important consideration in clinical applications.

As presented in Table 1, the first architecture "GCN+DC1" from [16] is the baseline for comparison, while the rest are the new models proposed in this work. We observe that both GCN and GTN encoder combined with DC2, defined in Equation (3) have obtained better scores in all measures than when they are combined with DC1, defined in Equation (2). This improvement can be explained by the additional trainable parameters of the decoder which projects the latent space to the FC prediction. When comparing the median Tril-MSE, we can notice that GTN increases the performance by up to 18% when compared

Results	Average Tril-MSE	Median Tril-MSE	Average TPearson	Median TPearson
GCN+DC1	0.0430±0.0016	0.0431	0.6628±0.0077	0.6671
GCN+DC2	0.0404±0.0018	0.0402	0.7003±0.0073	0.7069
GTN+DC1	0.0401±0.0020	0.0351	0.7160±0.0077	0.7228
GTN+DC2	0.0397±0.0018	0.033	0.7154±0.0061	0.7231

Table 1: Average and median test error for the graph convolutional and transformer networks using different decoders: DC1 and DC2 are defined in Equation (2) and (3), respectively.

to GCN. This significant change in the median error highlights the robustness of GTN to outliers. In addition, GTN also outperforms GCN at the supplementary measure TPearson. Also note that the GTN always provides a better FC prediction regardless of the decoder selected, showing that important information is captured in the *Length*, and *SIFT2* matrices not exploited by the GCN architecture.

5 Conclusion

A better understanding of the link between SC and FC has the potential to provide insights into how lesions to the structural substrate affect brain function. In this work, we have revisited the structure-function mapping via graph neural networks. We investigated the importance of a trainable decoder in a previously proposed GCN architecture and proposed a novel GTN architecture to predict FC from several diffusion derived SC matrices. We presented prediction results on 1050 subjects of the Human Connectome Project, showing that trainable parameters in the decoder improve the performance of the models. In addition, our results indicated that GTNs, which simultaneously exploit different realisations of the SC matrix, outperformed GCNs. Overall, the preliminary results presented in this work highlight the potential for graph neural networks in structure-function mapping.

6 Acknowledgment

This work has been supported by the ERC under the European Union’s Horizon 2020 research and innovation program (ERC Advanced Grant agreement No 694665:CoBCoM : Computational Brain Connectivity Mapping) and by the French government, through the 3IA Côte d’Azur Investments in the Future project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002.

Data were provided by the Human Connectome Project, WU-Minn Consortium (Principal Investigators: David Van Essen and Kamil Ugurbil; 1U54MH091657)

funded by the 16 NIH Institutes and Centers that support the NIH Blueprint for Neuroscience Research; and by the McDonnell Center for Systems Neuroscience at Washington University.

The authors are grateful to Inria Sophia Antipolis - Méditerranée [https://wiki.inria.fr/ClustersSophia/Usage_policy"Nef"](https://wiki.inria.fr/ClustersSophia/Usage_policy) computation cluster for providing resources and support.

The authors are grateful to the OPAL infrastructure from Université Côte d’Azur for providing resources and support.

References

1. O. Sporns, G. Tonomi, and R. Kötter. The human connectome: A structural description of the human brain. *PLoS Computational Biology*, 1:245–251, 2005.
2. J. Alstott, M. Breakspear, P. Hagmann, L. Cammoun, and O. Sporns, “Modeling the impact of lesions in the human brain,” *PLoS Computational Biology*, vol. 5, Article ID e1000408, 2
3. F. Váša, M. Shanahan, P. Hellyer, G. Scott, J. Cabral, and R. Leech, “Effects of lesions on synchrony and metastability in cortical networks,” *NeuroImage*, vol. 118, pp. 456–467, 2
4. C. Honey, O. Sporns, L. Cammoun, X. Gigandet, J.-P. Thiran, R. Meuli, and P. Hagmann, “Predicting human resting-state functional connectivity from structural connectivity,” *Proc. Natl. Acad. Sci. U.S.A.*, vol. 106, no. 6, pp. 2035–2040, 2009.
5. G. Deco, M. L. Kringelbach, V. K. Jirsa, and P. Ritter, “The dynamics of resting fluctuations in the brain: metastability and its dynamical cortical core,” *Scientific Reports*, vol. 7, p. 3095, 2
6. A. M. Hermundstad, D. S. Bassett, K. S. Brown et al., “Structural foundations of resting-state and task-based functional connectivity in the human brain,” *Proceedings of the National Academy of Sciences*, vol. 110, no. 15, pp. 6169–6174, 2
7. Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436, 2015.
8. Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang and P. S. Yu, "A Comprehensive Survey on Graph Neural Networks," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4-24, Jan. 2021, doi: 10.1109/TNNLS.2020.2978386.
9. F. Deligianni, G. Varoquaux, B. Thirion, D. J. Sharp, C. Ledig, R. Leech, and D. Rueckert. A framework for inter-subject prediction of functional connectivity from structural networks. *IEEE Transactions on Medical Imaging*, 32(12):2200–2214, 2013.
10. F. Abdelnour, H. U. Voss, and A. Raj. Network diffusion accurately models the relationship between structural and functional brain connectivity networks. *NeuroImage*, 90:335–347, 2014.
11. J. Meier, P. Tewarie, A. Hillebrand, L. Douw, B. W. van Dijck, S. M. Stufflebeam, and P. Van Mieghem. A mapping between structural and functional brain networks. *Brain Connectivity*, 6(4):298–311, 2016.
12. H. Liang and H. Wang. Structure–function network mapping and its assessment via persistent homology. *PLoS Computational Biology*, 13(1), 2017.
13. C. O. Becker, S. Pequito, G. J. Pappas, M. B. Miller, S. T. Grafton, D. S. Bassett, and V. M. Preciado. Spectral mapping of brain functional connectivity from diffusion imaging. *Scientific Reports*, 8(1411), 2018.

14. T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," ICLR, 2017.
15. J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," Proc. ICLR, 2014.
16. Y. Li, R. Shafipour, G. Mateos and Z. Zhang, "Mapping Brain Structural Connectivities to Functional Networks Via Graph Encoder-Decoder With Interpretable Latent Embeddings," 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2019, pp. 1-5, <https://doi.org/10.1109/GlobalSIP45357.2019.8969239>.
17. S. Yun, M. Jeong, R. Kim, J. Kang, "Graph Transformer Networks" NeurIPS 2019.
18. M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In Advances in neural information processing systems, pages 2017–2025, 2015.
19. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chap. Learning Internal Representations by Error Propagation, pp. 318-362. MIT Press, Cambridge, MA, USA (1986).
20. David K. Hammond, Pierre Vandergheynst, and Remi Gribonval. "Wavelets on graphs via spectral graph theory". Applied and Computational Harmonic Analysis, 30(2):129–150, 2011.
21. Michael Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In Advances in neural information processing systems, pp. 3844–3852, 2016.
22. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," ICLR, 2015.
23. X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," Intl. Conf. on Artificial Intelligence and Statistics, pp. 249–256, 2010.
24. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L. Lerer, A. (2017). Automatic Differentiation in PyTorch. NIPS 2017 Workshop on Autodiff.
25. J. D. Power, D. A. Fair, B. L. Schlaggar, and S. E. Petersen, "The development of human functional brain networks," Neuron, vol. 67, no. 5, pp. 735–748, 2010.
26. M. Rubinov and O. Sporns, "Complex network measures of brain connectivity: uses and interpretations," Neuroimage, vol.52, no. 3, pp. 1059–1069, 2010.
27. Samuel Deslauriers-Gauthier, Mauro Zucchelli, Matteo Frigo, Rachid Deriche. "A Unified Framework for Multimodal Structure-function Mapping Based on Eigenmodes". Medical Image Analysis, Elsevier, 2020, pp.22.10.1016/j.media.2020.101799. hal-02925913
28. Smith RE, Tournier JD, Calamante F, Connelly A. SIFT2: Enabling dense quantitative assessment of brain white matter connectivity using streamlines tractography. Neuroimage. 2015 Oct 1;119:338-51. doi: 10.1016/j.neuroimage.2015.06.092. Epub 2015 Jul 8. PMID: 26163802.