University of São Paulo

Institute of Science Mathematics and Computation

Final report

Explainability of Graph Neural Networks for the Assessment of Autism with fMRI Analysis

Student: Matheo Angelo Pereira Dantas

Advisor: Dr. André Carlos Ponce de Leon Ferreira de Carvalho

Concentration area: Artificial Intelligence, Machine Learning, Decision Making

Abstract

The diagnosis of autism today can only be done through clinical behavioral analysis, which is a procedure that is not very accessible. The use of Deep Neural Networks can help improve diagnosis with more complex biological data, but Deep Learning models are not interpretable, so predictions can not be reliable and the knowledge acquired by the model cannot be easily transferred to its users. For this reason, it is important to mitigate this interpretability obstacle with Explainable AI tools. In this context, we propose an investigation of these techniques. The goal of the project is to study explainability methods for Graph Neural Networks, which are neural networks designed to make predictions from data with complex connectivity, and use them to investigate the neurological patterns present in brain data from autistic people. To do this, we will apply different explainability techniques present in the scientific literature, evaluate them with objective interpretability metrics, and view and interpret the results.

Contents

1 Introduction

1.1 Autism Spectrum Disorder

According to the most important reference catalog for mental disorders, the DSM-V[1], ASD (Autism Spectrum Disorder) is a neurodevelopmental disorder characterized by deficits in social communication and repetitive patterns of behavior, in different contexts, manifesting since childhood. These characteristics can present with varying intensities among autistic people.

It is estimated that around 1% of the world's population has ASD[2]. People with ASD have a greater chance of developing depression and other conditions of psychological distress [1]. In addition, ASD is legally recognized as a disability in Brazil[3], which guarantees the right to enjoy accessibility policies and affirmative actions. Thus, the identification of autism is a matter of great social relevance, and is necessary to ensure the appropriate support for this group.

However, the condition does not have any biological markers known to science, and diagnosis is only possible through clinical analysis, through tests, interviews and observation of behavior by psychology professionals. This process can be inaccessible and inefficient on a large scale, motivating the academic community to search for biological markers of ASD.

One of the most widely used tools for this purpose is Machine Learning, where computers are "trained" through statistical calculations to identify patterns in data sets. This makes it possible to detect patterns that are difficult to perceive through manual human analysis, and thus, the discovery

of new knowledge in science[4]. Machine learning has already been used on various types of data from autistic people, such as facial expressions[5] and brain scans[6].

Among these data modalities, fMRI (Functional Magnetic Resonance Imaging[7]) stands out, an exam that films the patient's brain using magnetic resonance imaging. The exam records the intensity of the BOLD (Blood-Oxygen-Level-Dependent) signal in each region of the brain, aiming to analyze brain activity based on blood circulation in the brain. fMRI exams can be subdivided into task-fMRI, which observes the brain's response to a specific task performed by the patient, and R-fMRI (Resting-State fMRI), which only observes the brain normally, in a resting state.

1.2 Graph Neural Networks

One of the machine learning techniques that has been used to try to extract biological patterns from ASD, mainly in fMRI exams, is Graph Neural Networks, also called GNNs ("Graph Neural Networks"). Deep Neural Networks, in general, have been successful in tasks that use complex data, such as images and texts, because they can approximate complex functions from the composition of simpler functions, the layers of the neural network, and thus represent data at different levels of abstraction. Graph Neural Networks, in particular, operate on data in the form of graphs, where a graph consists of a set of points (nodes) and a set of connections between pairs of nodes (edges).

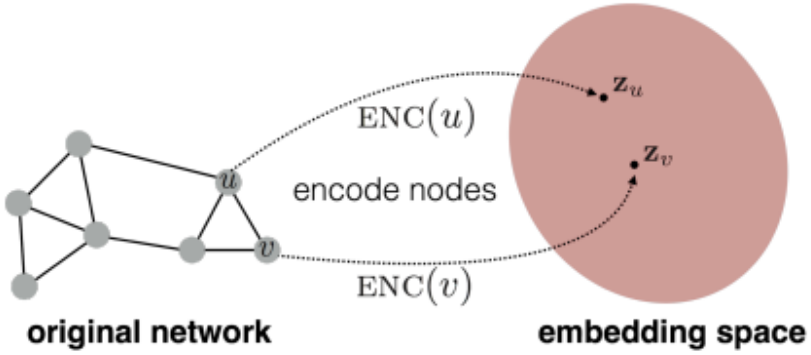

Figure 1: Illustration of the representation learning process in a graph. Each node is associated with a vector in the embedding space, which represents its original function in the graph. Source: [8].

1.2.1 GNNs as Graph Representation Learning Methods

As illustrated in Figure 1, GNNs are trained to obtain representations of the nodes of a graph in $\mathbb{R}^n$ (also called "embeddings") that encode important properties of the graph. These representations can be taken as the attributes of the node and used to classify the node, or they can be aggregated to represent the structure of the entire graph, which enables the task of graph classification.

1.2.2 Message passing

In GNNs, these representations are obtained through the message passing mechanism. Initially, each node has a vector (which can be a scalar) and, at each layer of the neural network, the nodes update their value based on the values of the neighborhood. When this process is complete, the result is

the numerical representations of the graph nodes. To obtain a representation of the entire graph, the node representations are aggregated, which is often called *pooling* or *readout*. Pooling is usually a permutation-invariant operation, like a simple average, because most applications assume that the graph representation should not be influenced by the order in which the nodes are numbered.

More formally, $h_l(u)$ is the vector stored at node u of the graph after the l-th layer of the neural network (where $h_0(u) = x_u$, the initial feature vector of node u), a message passing layer obtains $h_{l+1}(u)$ from the following algorithm[8]:

- **Message:** every node u in the graph receives a message $MSG_{l+1}(u, v, e_{uv})$ from each of its neighbors $v \in N(u)$, where e_{uv} is the edge connecting nodes u and v.
- **Aggregation:** the messages $MSG_{l+1}(u, v, e_{uv})$ received by u are aggregated with the permutation-invariant operation $m_{l+1}(u) = AGG_{l+1}(\{MSG_{l+1}(u, v, e_{uv}), v \in N(u)\})$, such as an average, producing the vector $m_{l+1}(u)$.
- **Update:** the vector $h_{l+1}(u)$ is obtained from an update function of the form $UPD_{l+1}(h_l(u), m_{l+1}(u))$.

1.3 Main GNN architectures

The following are the main Graph Neural Network architectures currently used:

1.3.1 Graph Convolutional Network

Graph Convolutional Networks (GCNs)[9] approximate spectral convolutions on graphs, similarly to ChebNet, but limit each convolution filter to consider only the signals from the neighborhood of each node, functioning as a message passing layer. Thus, a GCN convolution layer is of the form:

$$H^{(l+1)} = \sigma\big(D^{-\frac{1}{2}}\tilde{A}D^{-\frac{1}{2}}H^{(l)}W^{(l)}\big)$$

Where D is a diagonal matrix such that d_{ii} is the degree (number of neighbors) of vertex i.

1.3.2 GraphSAGE

A message passing layer in GraphSAGE[10] follows the equation below:

$$h_{l+1}(u) = \sigma(W^a_{l+1} \cdot h_l(u) + W^b_{l+1} \cdot AGG(\{h_l(v), v \in N(u)\}))$$

Where $AGG(\cdot)$ is a pre-selected aggregation function, such as simple average or maximum value of each coordinate of the vectors; W^a_{l+1} and W^b_{l+1} are weight matrices to be optimized in the neural network training; and $\sigma(\cdot)$ is a non-linear activation function.

1.3.3 Graph Attention Network

Graph Attention Networks (GAT)[11] aim to apply the attention mechanism popularized by Transformers[12] in GNN, and calculate attention scores be-

tween only adjacent nodes, instead of calculating attention for all pairs of Tokens, as in Transformers. A standard GAT layer is of the form:

$$h(u) = \sigma\big(\sum_{v \in N_u} \alpha_{uv} W h(v)\big)$$

Where we assume that every node is adjacent to itself, and α_{uv} is the attention from node u to node v, given by:

$$\alpha_{uv} = \frac{\exp(e_{uv})}{\sum_{v \in N_u} \exp(e_{uv})} = \frac{\exp(\text{LeakyRELU}\big(a^T [Wh(u) \| Wh(v)]\big))}{\sum_{v \in N_u} \exp(\text{LeakyRELU}\big(a^T [Wh(u) \| Wh(v)]\big))}$$

Where $\|$ indicates vector concatenation and LeakyRELU[...].

1.3.4 Graph Isomorphism Network

The Graph Isomorphism Network (GIN)[13] uses the concept of representation power, which is the ability of a graph neural network to generate different outputs for graphs that are not isomorphic to each other. Message passing-based GNNs have representation power at most as high as the Weisfeiler-Lehman Isomorphism Test[14], and GIN is designed to be able to achieve this representation power, having the following message passing formula:

$$h_u^{(k)} = \text{MLP}^{(k)}\big((1 + \epsilon^{(k)}) h_u^{(k-1)} + \textstyle\sum_{v \in N(u)} h_v^{(k-1)}\big)$$

Where $\epsilon^{(k)}$ is a number that can be fixed as a hyperparameter or can be

a trainable parameter of the neural network, and MLP is a Multi-Layer Perceptron function[15] with trainable weights. Furthermore, the global pooling layer uses a sum rather than an average.

1.4 Explainability of Graph Neural Networks

One of the main obstacles to implementing neural networks in society is the lack of interpretability. Since neural networks perform very complex calculations, the knowledge acquired during the training process is not humanly understandable. As a result, it is not possible to understand the decision-making process of neural networks, which makes it difficult to ensure the reliability of predictions and the analysis of their ethical implications. To mitigate this problem, the area of Explainable Artificial Intelligence (XAI) emerged, which aims to explain trained Machine Learning models that have low interpretability.

Graphs present an additional challenge for interpretability, because unlike most deep learning techniques, graphs can present different connectivity patterns in the same application, which must be interpreted in conjunction with the numerical signals of the data.

There are a number of aspects of GNNs that can be the subject of investigation by explainability algorithms, and they can be subdivided according to a taxonomy of algorithms[16][17], illustrated in Figure 2.

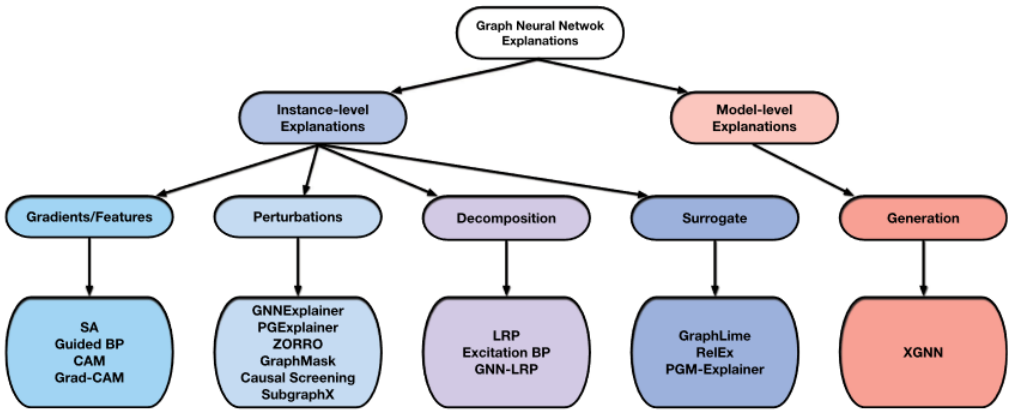

Figure 2: Taxonomy of Graph Neural Network Explainability algorithms. Source: [18].

1.4.1 Saliency Methods

Saliency methods, adapted from Convolutional Neural Networks, present heat maps that measure the degree of influence of each part of the graph in a given prediction. This can be measured by the gradient of the features of each node in relation to the output of the neural network, as is the case with SA and Guided BP[19], or by class activation maps (CAM and GradCAM[20]), which highlight nodes with high values in the features that have a positive influence on the predicted class, using the features of the final embeddings of the nodes.

1.4.2 Perturbation Methods

The largest group of GNN interpretability methods is that of perturbation methods. These methods consist of perturbing the features (nodes, edges,

or node features) of the graph and analyzing which small perturbations can have a significant effect on the prediction, usually with the aim of generating a "mask" with the most important parts of the graph. Depending on the type of perturbation, these masks can be node masks (Zorro[21], SubgraphX[22]), edge masks (GNNExplainer[23], PGExplainer[24], GraphMask[25], CausalScreening[26]), or node features (GNNExplainer[23], Zorro[21]).

In addition, we can classify these methods into soft mask generators (GNNExplainer[23]), which accept partial insertion of features into the mask; discrete masks (Zorro[21], Causal Screening[26], SubgraphX[22], Gem[27]), where each feature is either entirely inside or outside the mask; and approximately discrete (PGExplainer[24], GraphMask[25]), which also accept partial insertion but are optimized to avoid intermediate insertion values, approximating discrete masks. Soft masks can address the problem of introduced evidence[28], where partial removal of a feature can change its meaning for the neural network, instead of masking the contribution of the feature.

In addition to mask generation, another possible approach is the generation of counterfactuals (CF-GNNExplainer[29], RCExplainer[30]), that is, graphs that are similar to the input graph (in the case of node classification, the computational graph of the node) but that generate different results.

1.4.3 Decomposition Methods

In decomposition methods, the output of the neural network is successively decomposed into the layers of the neural network, applying decomposition rules based on Backpropagation[15] from the last layer to the input of the

GNN, where we can visualize the contribution share of each feature. This decomposition can be done between nodes (LRP[31] and Excitation BP[20]), or between paths (GNN-LRP[32]) of size k, where k is the number of message passing layers of the GNN, in order to display the circulation of information in the graph through the GNN message passing.

1.4.4 Replacement Methods

Replacement methods employ interpretable classical machine learning models (such as Linear Regression) that fit the local decision boundary of a given prediction, replacing the GNN. The currently existing graph replacement models are GraphLIME[33], RelEx[34] and PGMExplainer[35]. GraphLIME and RelEx explain only node classifications, while PGMExplainer explains both node and graph classifications.

1.4.5 Model-Level Explanations

All the methods mentioned in the above sections explain neural networks at the instance level, i.e., they serve to justify a specific prediction of a GNN. In model-level explanations, the goal is to explain the decision-making process of the neural network in general. To this end, it is possible to extract the most frequent masks in instance-level explanations, or to use methods specialized in model-level explanations.

In the literature, there are only two widely established interpretability methods for general-purpose GNNs at the model level: XGNN[36], which generates input graphs for the GNN by maximizing the predicted probability

for each class, and GCFExplainer[37], which generates for each class a small set of prototypes that are counterfactuals of many examples of the class.

1.4.6 Self-explaining GNNs

Another possible approach is, instead of explaining trained models ted with the aforementioned algorithms, train GNNs focused on explainability [16]. Self-explaining GNNs make predictions through interpretable mechanisms and use these mechanisms to generate explanations in conjunction with the predictions. The models that currently follow this purpose are SE-GNN[38], which classifies nodes based on comparison with similar labeled nodes, and ProtGNN[39], a generative neural network of class prototypes (like XGNN[36]), where the predicted class for a new graph is the class with the most similar prototypes to the new graph. Furthermore, GIB[40], which is trained to make accurate predictions using the least amount of information possible (formalized through mutual information functions) from the input graph, was created to mitigate the problem of adversarial attacks, but the succinct representation generated from the "information bottleneck" can be exploited for explainability.

We can also include among the self-explaining GNNs causality-based models (DIR[41], DisC[42], CIGA[43]). Causal GNNs are trained to separate causal correlations from spurious correlations in the data, and when making predictions, they first identify causal features (e.g. subgraphs with evidence in favor of a given class) and use only these features to make predictions. In addition to the gain in performance and reliability in predictions, this ap-

proach also favors interpretability, which is possible through the analysis of causal relationships.

1.4.7 Objective explainability metrics for GNNs

The main objective explainability metrics for GNNs are:

- **Fidelity:** Fidelity[18] evaluates whether the removal of masks selected by the explanation has a high impact on the result of the neural network. For methods that do not generate discrete masks, a mask is obtained from all features whose importance is greater than a certain threshold. Therefore, there are different ways to calculate Fidelity:
  - $Fidelity{+}_{acc} = \frac{1}{N}\sum_{i=1}^{N}\big(1(\hat{y}_i = y_i) - 1(\hat{y}_i{}^{1-m_i} = y_i)\big)$
  - $Fidelity{-}_{acc} = \frac{1}{N}\sum_{i=1}^{N}\big(1(\hat{y}_i = y_i) - 1(\hat{y}_i{}^{m_i} = y_i)\big)$
  - $Fidelity{+}_{prob} = \frac{1}{N}\sum_{i=1}^{N}\big(f(G_i)_{y_i} - f(G_i^{1-m_i})_{y_i}\big)$
  - $Fidelity{-}_{prob} = \frac{1}{N}\sum_{i=1}^{N}\big(f(G_i)_{y_i} - f(G_i^{m_i})_{y_i}\big)$

  Where $\hat{y}_i$ is the predicted class for example i in the neural network, y_i is the true class of i, $\hat{y}_i{}^{m_i}$ is the neural network prediction for i masking all features outside the mask m_i, $1 - m_i$ is the complementary mask to m_i, $1(\hat{y}_i{}^{m_i} = y_i)$ is a function that returns 1 if $\hat{y}_i{}^{m_i} = y_i$ and 0 otherwise, and $f(G_i)_{y_i}$ is the probability output of the neural network for the graph G_i in the class of G_i, y_i.

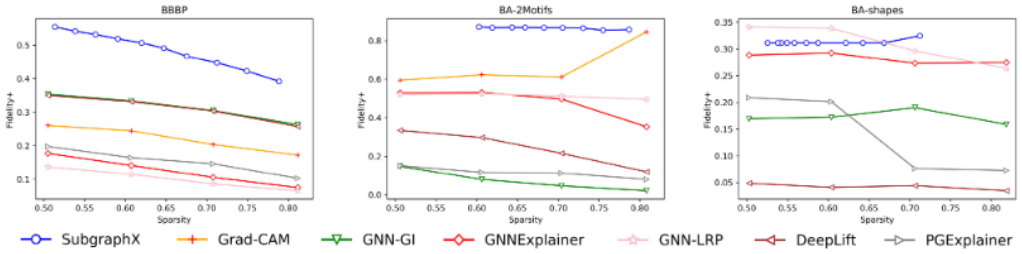

Figure 3: Comparison graphs of fidelity and sparsity. Each graph corresponds to a different data set, and each line of the graph corresponds to scores obtained in an explainability method. Source: [18].

- **Sparsity:** A good explanation needs to be succinct and interpretable. The sparsity[**?**] of a prediction can be given by $1 - \frac{|m|}{|M|}$, where $|m|$ is the number of features selected by the mask and $|M|$ is the total number of features of the analyzed modality (e.g., the number of nodes in the graph). The sparsity of a model-level method is the average of the instance-level sparsities in the tested samples.

- **Accuracy:** In cases where the features that determine the class of an example are known, it is possible to measure the accuracy of the explanations in identifying these features.

We consider that there is a trade-off between Fidelity and Sparsity, since the more complete a feature mask is, the closer the behavior of the mask is to the complete set of features. Thus, aiming for a fairer comparison, it is common to visualize the fidelity of the explanation as a function of sparsity, as in Figure **??**.

In addition to fidelity, sparsity and accuracy, which are the main metrics,

there are alternatives:

- **Contrastivity:** In a coherent explanation, we want evidence in favor of one class not to also be evidence in favor of the opposite class. Thus, for binary classification problems, we can evaluate the contrastivity[20] of the explanations as $\frac{d_H(m_0,m_1)}{m_0 \vee m_1}$, where m_0 and m_1 are respectively the negative and positive class masks for the same prediction, $d_H(m_0, m_1)$ is the Hamming distance (number of features that are in exactly one of the masks between m_0 and m_1), and $m_0 \vee m_1$ is the size of the union between the masks m_0 and m_1.

- **Stability:** Stable interpretability methods identify important features for the model regardless of small variations in implementation, so that explanations do not change with broad replication of the model's results. In situations where this is desirable, stability is evaluated[**?**].

- **Consistency:** If there are specific features that determine the class of a given example, high-performance models are expected to be consistently more sensitive to these features. This can be assessed using the Consistency metric[**?**].

1.5 GNNs applied to the study of ASD

Given their ability to extract complex patterns in connected data, Graph Neural Networks have been used to predict ASD diagnosis several times in the academic literature[6], with many of these using the public fMRI data

from ABIDE I[1].

1.5.1 GNN models for ASD diagnosis

In the literature, there are two main ways to represent fMRI data as graphs. The first is the population graph, where each patient is represented as a node in the graph and edges are inserted between similar patients, with diagnosis prediction being a node classification task. The second, which is the focus of the present study, is the individual graph ("subject graph"). In this representation, the patient's brain is modeled as a graph, where each node corresponds to a brain region, and edges are inserted between pairs of regions that are highly correlated in the fMRI BOLD series. Thus, GNNs applied to individual graphs predict the diagnosis of ASD through the graph classification task. In addition, phenotypic data are concatenated to the representations generated by the GNNs to aid in the prediction.

[...]

1.5.2 Use of explainability for extracting ASD biomarkers

Interpretability is very important for the application of neural networks in the medical field, not only to facilitate the responsible use of predictions, but also to identify possible biological markers of the conditions investigated, and through more rigorous studies, consolidate them as scientific knowledge. In the case of diagnosis based on fMRI analysis, one of the main focuses of interpretability is to identify brain regions that are more active in people

[1] http://fcon_1000.projects.nitrc.org/indi/abide/abide_I.html

with that diagnosis.

In the case of GNNs, interpretability algorithms are used to detect important nodes in the graph, which would correspond to the brain's ROI. Algorithms based on node masks are used, or node masks are created from the discretization of saliency maps. These algorithms originally work at the instance level, and to obtain interpretations at the class level, the most frequent brain regions in the node masks are selected.

In the experiments already recorded in the literature, both general-purpose methods for interpretability of GNNs and *ad-hoc* methods, focused only on the application studied, were used.

[...]

2 Data and Preprocessing

2.1 Data collection

The data used in the project come from the ABIDE I database, mentioned earlier in the text. The database contains data from 1,112 individuals, 539 of whom have ASD (Autism Spectrum Disorder) and 573 of whom do not have ASD (Typical Control). These data were obtained from 16 different collection points. Each collection point has a set of files for download, and in addition to data from the equipment used in the exams, there are two main files: the phenotypic data file and the exam data file.

The phenotypic data, which can be either in the file of their respective collection point or in the file that has the complete set of phenotypic data, consist of tables in the *.csv* format, where each row represents a patient and each column represents different characteristics of the patients, such as sex, age, presence or absence of ASD, and the patient's diagnosis based on the DSM-IV-TR criteria[44].

The exam database has two types of exams: anatomical and R-FMRI (Resting State Functional Magnetic Resonance Imaging), which is the data of interest for training the neural networks. The exams from each collection point must be downloaded separately and come with several folders identified by the corresponding patient identification number. The exams are in the *.nii.gz* format, which can be manipulated using the *nibabel* and *nilearn* libraries of the Python language.

2.2 Overview

2.2.1 Phenotypic Data

In total, there are 74 columns in the phenotypic database table, one of which is the collection point, another is the patient's identification number in the database (corresponding to a folder with the same number in the exam database), and the rest are the phenotypic data itself. The complete list of table attributes with their respective legends is on the ABIDE website I[2]. Among these attributes, we can highlight the presence of demographic attributes (sex, age, DSM-IV diagnosis, BMI), and neuropsychological test scores (IQ, ADI-R, ADOS, SRS, SCQ, Total Autism Quotient, Vineland, WISC-IV), which describe in more detail the patient's neuropsychological profile.

2.2.2 Exam Data

An fMRI exam from the ABIDE I database comes as a video composed of three-dimensional images with a resolution of 64 x 28 x 28, collected in a total of 240 frames over time.

2.3 Preprocessing

To build a graph from the fMRI scan, we first need to group the signals from each region of interest (ROI) of the brain. This can be done using a

[2]https://fcon_1000.projects.nitrc.org/indi/abide/ABIDE_LEGEND_V1.02.pdf

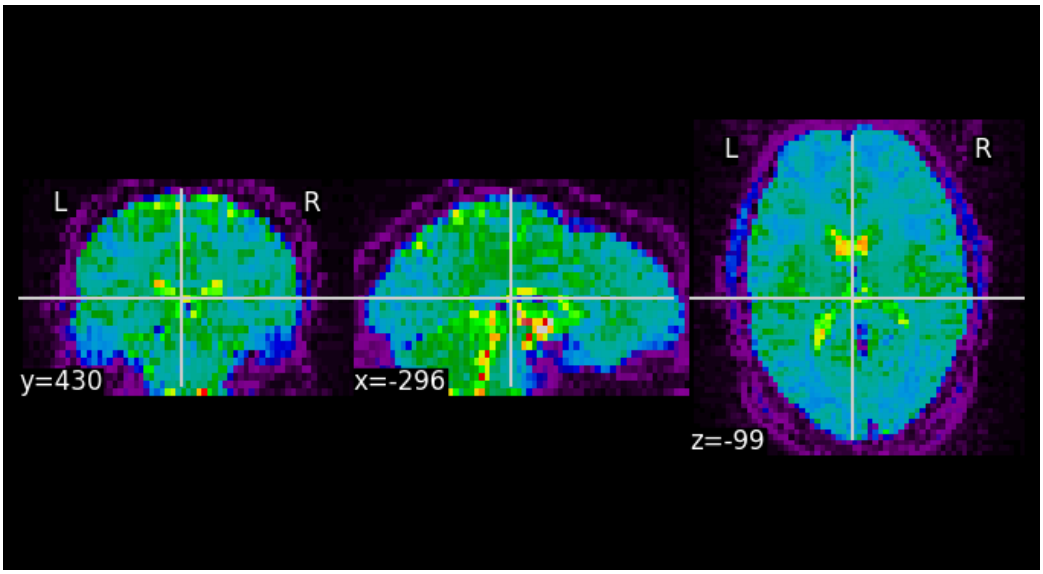

Figure 4: Example of a frame from an ABIDE I fMRI scan. Each image shows a cross-section of the scan in 3D, with the caption indicating the coordinate in the (x,y,z) system.

Brain Atlas, which maps voxels from a brain image to their corresponding regions in the brain. We will use the Harvard-Oxford Atlas, which is widely used in Machine Learning applications using ABIDE I data. These atlases are represented as three-dimensional arrays, where the value contained in (i, j, k) indicates the region to be assigned to voxel (i, j, k) in the fMRI scan. Each ROI region receives the average of the BOLD series of the voxels assigned to it by the atlas.

It is possible to bypass the manual execution of the task with the Preprocessed Connectomes Project[3]. This project provides the fMRI data of the ABIDE I volunteers pre-processed, with exactly one BOLD series for each ROI, calculated in the manner described above. There are several atlas options available in the project.

Given the BOLD series of each ROI of a patient, we can construct the graph of the patient's brain by inserting an edge between each pair of highly correlated regions according to a given metric, such as Pearson's Correlation or DTW. In addition, we can use the BOLD series to calculate the feature vector of each node in the graph. The identification of each ROI also determines the feature vector, since classical GNNs have output invariant to node permutation, and this is not a desirable property in this context.

[3]http://preprocessed-connectomes-project.org/abide/index.html

2.4 Exploratory Analysis

2.4.1 Phenotypic Data

The phenotypic dataset has a significant amount of missing data, but this lack of data is concentrated in the columns of neuropsychological tests, since each collection point chose its own set of tests to assess the diagnosis of ASD in the volunteers. The columns with no missing data are 'SITE_ID', 'SUB_ID', 'DX_GROUP', 'DSM_IV_TR', 'AGE_AT_SCAN', 'SEX' and 'EYE_STATUS_AT_SCAN'.

Analyzing the demographics of the data, it is worth noting that the majority of ABIDE I volunteers are male, with the female population having a greater need for access to diagnostic technologies, as it is suspected that autism is significantly underdiagnosed in women[1].

3 Neural Network Training

4 Application of Explainability Methods

4.1 Methodology

The GNNs will be trained several times with small hyperparameter perturbations, in order to later focus the explanations less on the trained models and more on the objective patterns of the data. After training the models, we will apply explainability algorithms in order to find the most relevant nodes for the classification of each instance, generating masks of important nodes. In addition, we will observe the frequency of each node in the node masks of each method, in order to evaluate the relevance of each brain ROI at the model level.

To evaluate the explanations offered by each interpretability method, we will use the Fidelity, Sparsity, and Contrastivity metrics, already consolidated in the interpretability of Graph Neural Networks. In addition, given the probability distribution of the nodes from the frequencies of each node in the node masks, we will calculate the entropy of this probability distribution, in order to evaluate the stability of the explanation methods and the ability to find biological markers of ASD in general.

4.2 Results

4.2.1 Explainability metrics

4.2.2 Main possible bioindicators

References

[1] American Psychiatric Association, *Diagnostic and Statistical Manual of Mental Disorders, 5th Edition*. Washington, D.C.: American Psychiatric Publishing, 5 ed., 2013.

[2] J. Zeidan, E. Fombonne, J. Scorah, A. Ibrahim, M. S. Durkin, S. Saxena, A. Yusuf, A. Shih, and M. Elsabbagh, "Global prevalence of autism: A systematic review update," *Autism research*, vol. 15, no. 5, pp. 778–790, 2022.

[3] Presidência da República Federativa do Brasil, "Lei nº 12.764, de 27 de dezembro de 2012," 2012. Institui a Política Nacional de Proteção dos Direitos da Pessoa com Transtorno do Espectro Autista.

[4] H. Wang, T. Fu, Y. Du, W. Gao, K. Huang, Z. Liu, P. Chandak, S. Liu, P. Van Katwyk, A. Deac, *et al.*, "Scientific discovery in the age of artificial intelligence," *Nature*, vol. 620, no. 7972, pp. 47–60, 2023.

[5] W. Liu, M. Li, and L. Yi, "Identifying children with autism spectrum disorder based on their face processing abnormality: A machine learning framework," *Autism Research*, vol. 9, no. 8, pp. 888–898, 2016.

[6] S. Zhang, J. Yang, Y. Zhang, J. Zhong, W. Hu, C. Li, and J. Jiang, "The combination of a graph neural network technique and brain imaging to diagnose neurological disorders: A review and outlook," *Brain Sciences*, vol. 13, no. 10, p. 1462, 2023.

[7] S. Zhang and R. L. Chiang-shan, "Functional connectivity mapping of the human precuneus by resting state fmri," *Neuroimage*, vol. 59, no. 4, pp. 3548–3562, 2012.

[8] W. L. Hamilton, "Graph representation learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 14, no. 3, pp. 1–159.

[9] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[10] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[11] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[12] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[13] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?," *arXiv preprint arXiv:1810.00826*, 2018.

[14] B. Weisfeiler and A. Leman, "The reduction of a graph to canonical form and the algebra which appears therein," *nti, Series*, vol. 2, no. 9, pp. 12–16, 1968.

[15] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[16] E. Dai, T. Zhao, H. Zhu, J. Xu, Z. Guo, H. Liu, J. Tang, and S. Wang, "A comprehensive survey on trustworthy graph neural networks: Privacy, robustness, fairness, and explainability," *Machine Intelligence Research*, pp. 1–51, 2024.

[17] H. Yuan, H. Yu, S. Gui, and S. Ji, "Explainability in graph neural networks: A taxonomic survey," *CoRR*, vol. abs/2012.15445, 2020.

[18] H. Yuan, H. Yu, S. Gui, and S. Ji, "Explainability in graph neural networks: A taxonomic survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 5, pp. 5782–5799, 2022.

[19] F. Baldassarre and H. Azizpour, "Explainability techniques for graph convolutional networks," *arXiv preprint arXiv:1905.13686*, 2019.

[20] P. E. Pope, S. Kolouri, M. Rostami, C. E. Martin, and H. Hoffmann, "Explainability methods for graph convolutional neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10772–10781, 2019.

[21] T. Funke, M. Khosla, and A. Anand, "Hard masking for explaining graph neural networks," 2020.

[22] H. Yuan, H. Yu, J. Wang, K. Li, and S. Ji, "On explainability of graph neural networks via subgraph explorations," in *International conference on machine learning*, pp. 12241–12252, PMLR, 2021.

[23] Z. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "Gnnexplainer: Generating explanations for graph neural networks," *Advances in neural information processing systems*, vol. 32, 2019.

[24] D. Luo, W. Cheng, D. Xu, W. Yu, B. Zong, H. Chen, and X. Zhang, "Parameterized explainer for graph neural network," *Advances in neural information processing systems*, vol. 33, pp. 19620–19631, 2020.

[25] M. S. Schlichtkrull, N. De Cao, and I. Titov, "Interpreting graph neural networks for nlp with differentiable edge masking," *arXiv preprint arXiv:2010.00577*, 2020.

[26] X. Wang, Y. Wu, A. Zhang, X. He, and T.-s. Chua, "Causal screening to interpret graph neural networks," 2020.

[27] W. Lin, H. Lan, and B. Li, "Generative causal explanations for graph neural networks," in *International Conference on Machine Learning*, pp. 6666–6679, PMLR, 2021.

[28] P. Dabkowski and Y. Gal, "Real time image saliency for black box classifiers," *Advances in neural information processing systems*, vol. 30, 2017.

[29] A. Lucic, M. A. Ter Hoeve, G. Tolomei, M. De Rijke, and F. Silvestri, "Cf-gnnexplainer: Counterfactual explanations for graph neural networks," in *International Conference on Artificial Intelligence and Statistics*, pp. 4499–4511, PMLR, 2022.

[30] M. Bajaj, L. Chu, Z. Y. Xue, J. Pei, L. Wang, P. C.-H. Lam, and Y. Zhang, "Robust counterfactual explanations on graph neural networks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 5644–5655, 2021.

[31] R. Schwarzenberg, M. Hübner, D. Harbecke, C. Alt, and L. Hennig, "Layerwise relevance visualization in convolutional text graph classifiers," *arXiv preprint arXiv:1909.10911*, 2019.

[32] T. Schnake, O. Eberle, J. Lederer, S. Nakajima, K. T. Schütt, K.-R. Müller, and G. Montavon, "Higher-order explanations of graph neural networks via relevant walks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7581–7596, 2022.

[33] Q. Huang, M. Yamada, Y. Tian, D. Singh, and Y. Chang, "Graphlime: Local interpretable model explanations for graph neural networks," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[34] Y. Zhang, D. Defazio, and A. Ramesh, "Relex: A model-agnostic relational model explainer," in *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 1042–1049, 2021.

[35] M. Vu and M. T. Thai, "Pgm-explainer: Probabilistic graphical model explanations for graph neural networks," *Advances in neural information processing systems*, vol. 33, pp. 12225–12235, 2020.

[36] H. Yuan, J. Tang, X. Hu, and S. Ji, "Xgnn: Towards model-level explanations of graph neural networks," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 430–438, 2020.

[37] Z. Huang, M. Kosan, S. Medya, S. Ranu, and A. Singh, "Global counterfactual explainer for graph neural networks," in *Proceedings of the*

Sixteenth ACM International Conference on Web Search and Data Mining, pp. 141–149, 2023.

[38] E. Dai and S. Wang, "Towards self-explainable graph neural network," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 302–311, 2021.

[39] Z. Zhang, Q. Liu, H. Wang, C. Lu, and C. Lee, "Protgnn: Towards self-explaining graph neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 9127–9135, 2022.

[40] T. Wu, H. Ren, P. Li, and J. Leskovec, "Graph information bottleneck," *Advances in Neural Information Processing Systems*, vol. 33, pp. 20437–20448, 2020.

[41] Y.-X. Wu, X. Wang, A. Zhang, X. He, and T.-S. Chua, "Discovering invariant rationales for graph neural networks," *arXiv preprint arXiv:2201.12872*, 2022.

[42] S. Fan, X. Wang, Y. Mo, C. Shi, and J. Tang, "Debiasing graph neural networks via learning disentangled causal substructure," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24934–24946, 2022.

[43] Y. Chen, Y. Zhang, Y. Bian, H. Yang, M. Kaili, B. Xie, T. Liu, B. Han, and J. Cheng, "Learning causally invariant representations for out-of-distribution generalization on graphs," *Advances in Neural Information Processing Systems*, vol. 35, pp. 22131–22148, 2022.

[44] A. P. Association, *Diagnostic and Statistical Manual of Mental Disorders: DSM-IV-TR*. Washington, DC: American Psychiatric Association, 4th, text revision ed., 2000.