Principle of Relevant Information for Graph Sparsification

Abstract

Graph sparsification aims to reduce the number of edges of a graph while maintaining its structural properties. In this paper, we propose the first general and effective information-theoretic formulation of graph sparsification, by taking inspiration from the Principle of Relevant Information (PRI). To this end, we extend the PRI from a standard scalar random variable setting to structured data (i.e., graphs). Our Graph-PRI objective is achieved by operating on the graph Laplacian, made possible by expressing the graph Laplacian of a subgraph in terms of a sparse edge selection vector $\mathbf{w}$. We provide both theoretical and empirical justifications on the validity of our Graph-PRI. We also analyze its analytical solutions in a few special cases. We finally present three representative real-world applications, namely graph sparsification, graph regularized multi-task learning, and medical imaging-derived brain network classification, to demonstrate the effectiveness, the versatility and the enhanced interpretability of our approach over prevalent sparsification techniques.

1 INTRODUCTION

Many complex structures and phenomena are naturally described as graphs and networks (e.g., social networks, brain functional connectivity [Zhou et al., 2020], climate causal effect network [Nowack et al., 2020], etc.). However, it is always prohibitive to exactly visualize and analyze a graph even with moderate size due to the quadratic growth in the number of edges. Therefore, techniques to sparsify graphs by pruning less informative edges have gained increasing attention in last two decades [Spielman and Srivastava, 2011, Bravo Hermsdorff and Gunderson, 2019, Wu and Chen, 2020]. Apart from offering a much easier visualization, graph sparsification can be used in multiple ways. For example, it may reduce the storage space and accelerate the running time of machine learning algorithms involving graph regularization, with negligible accuracy loss [Sadhanala et al., 2016]. When differentiable privacy is a major concern, it can remove or hide edges for the purpose of information protection [Arora and Upadhyay, 2019].

On the other hand, there is a recent trend to leverage the ideas or principles in Information Theory to problems related to graph or graph neural networks. Let $\mathcal{X}$ denote graph input data which may encode both graph structure information (characterized by either adjacency matrix A or graph Laplacian L) and node attributes, and Y the desired response such as node labels or graph labels. A notable example is the famed Information Bottleneck (IB) approach [Tishby et al., 1999], which formulates the learning as:

$$\mathcal{L}_{\text{IB}} = \min I(\mathcal{X};T) - \beta I(Y;T), \tag{1}$$

in which $I(\cdot;\cdot)$ denotes the mutual information. T is the object we want to learn or infer from $\{\mathcal{X}, Y\}$ that can be used as graph node representation [Wu et al., 2020] or as the most infromative and interpretable subgraph for the label Y [Yu et al., 2020]. β is a Lagrange multiplier that controls the trade-off between the **sufficiency** (the performance of T on down-stream task, as quantified by $I(Y;T)$) and the **minimality** (the complexity of the representation, as measured by $I(\mathcal{X};T)$).

Instead of using the IB approach, we explore the feasibility and potency of another less well-known information-theoretic principle - the Principle of Relevant Information (PRI) [Principe, 2010, Chapter 8] - in graph data, assuming only a single random variable $\mathcal{X}$ is given. Different from IB that requires an auxiliary relevant variable Y and possibly the joint distribution of $\mathbb{P}(\mathcal{X}, Y)$, the PRI is fully unsupervised and aims to obtain a reduced statistical representation T by decomposing $\mathcal{X}$ with:

$$\mathcal{L}_{\text{PRI}} = \min H(T) + \beta D(\mathbb{P}(\mathcal{X}) \| \mathbb{P}(T)), \tag{2}$$

where $H(T)$ refers to the entropy of T. The minimization

Submitted to the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022). **To be used for reviewing only**.

of entropy can be viewed as a means of reducing uncertainty and finding the statistical **regularity** in T. $D(\mathbb{P}(\mathcal{X})\|\mathbb{P}(T))$ is the divergence between the distributions of $\mathcal{X}$ (i.e., $\mathbb{P}(\mathcal{X})$) and T (i.e., $\mathbb{P}(T)$), which quantifies the **descriptive power** of T about $\mathcal{X}$.

So far, PRI has only been used in a standard scalar random variable setting. Recent applications of PRI include, but not limited to, selecting the most relevant examples from the majority class in imbalanced classification [Hoyos-Osorio et al., 2021], and learning disentengled representations with variational autoencoder [Li et al., 2020]. Usually, one uses the 2-order Rényi's entropy [Rényi, 1961] to quantify $H(T)$ and the Cauchy-Schwarz (CS) divergence [Jenssen et al., 2006] to quantify $D(\mathbb{P}(X)\|\mathbb{P}(T))$ for ease of optimization.

In this paper, we extend PRI to graph data. This is not a trivial task, as the Rényi's quadratic entropy and CS divergence are defined over probability space and do not capture any structure information. We also exemplify our Graph-PRI with an application in graph sparsification. To summarize, our contributions are fourfold:

- Taking graph Laplacian as the input variable, we propose a new information-theoretic formulation for graph sparsification, by taking inspiration from PRI.
- We provide theoretical and empirical justifications to the objective of Graph-PRI on graph sparsification. We also analyze the analytical solutions in some special cases of hyperparameter β.
- We demonstrate that the graph Laplacian of the resulting subgraph can be elegantly expressed in terms of a sparse edge selection vector $\mathbf{w}$, which significantly simplify the learning argument of Graph-PRI. We also show that the objective of Graph-PRI is differentiable, which further simplifies the optimization.
- Experimental results on graph sparsification, graph-regularized multi-task learning, and brain network classification demonstrate the versatility and compelling performance of Graph-PRI.

2 PRELIMINARY KNOWLEDGE

2.1 PROBLEM DEFINITION AND NOTATIONS

Consider an undirected graph $G = (V, E)$ with a set of nodes $V = \{v_1, \cdots, v_N\}$ and a set of edges $E = \{e_1, \cdots, e_M\}$ which reveals the connections between nodes. The objective of graph sparsification is to preferentially retain a small subset of edges from G to obtain a sparsified surrogate graph $G_s = (V, E_s)$ with the edge set $E_s \in E$ such that $|E_s| \ll M$.

The topology of G is essentially determined by its graph Laplacian $L = D - A$, where A is the adjacency matrix and $D = \mathrm{diag}\,(\mathbf{d})$ is the diagonal matrix formed by the degrees of the vertices $d_i = \sum_{j=1}^{N} A_{ij}$. Consider an arbitrary orientation of edges of G, the incidence matrix $B = [\mathbf{b}_1, \cdots, \mathbf{b}_M]$ of G is a $N \times M$ matrix whose entries is given by:

$$[\mathbf{b}_m]_i = \begin{cases} +1 & \text{if node } v_i \text{ is the head of edge } e_m \\ -1 & \text{if node } v_i \text{ is the tail of edge } e_m \\ 0 & \text{otherwise} \end{cases}. \quad (3)$$

Mathematically, L can be expressed in terms of B as:

$$L = BB^T = \sum_{m=1}^{M} \mathbf{b}_m \mathbf{b}_m^T. \quad (4)$$

Suppose the subgraph G_s contains K edges, one can obtain G_s from G through a edge selection vector $\mathbf{w} = [w_1, \cdots, w_M]^T \in \{0, 1\}^M$ such that $\|\mathbf{w}\|_0 = K$, where $w_m = 1$ if an edge belongs to the edge subset E_s, and $w_m = 0$ otherwise. Finally, one can write the graph Laplacian L_s of the K-sparse graph as a function of $\mathbf{w}$ as:

$$L_s(\mathbf{w}) = \sum_{m=1}^{M} w_m \mathbf{b}_m \mathbf{b}_m^T = B\,\mathrm{diag}(\mathbf{w})B^T, \quad (5)$$

in which $\mathrm{diag}\,(\mathbf{w}) \in \mathbb{R}^{M \times M}$ is a square diagonal matrix with $\mathbf{w}$ on the main diagonal.

Note that, Eq. (5) also applies for weighted graph $G = (V, W)$ by simply reformulate the incidence matrix B as:

$$[\mathbf{b}_m]_i = \begin{cases} +\sqrt{\mu_m} & \text{if node } v_i \text{ is the head of edge } e_m \\ -\sqrt{\mu_m} & \text{if node } v_i \text{ is the tail of edge } e_m \\ 0 & \text{otherwise} \end{cases}, \quad (6)$$

in which μ_m is the weight of edge e_m.

In what follows, we will design a learning-based approach to optimally obtain the edge selection vector $\mathbf{w}$ by making use the general idea of PRI.

2.2 GRAPH SPARSIFICATION

Substantial efforts have been made on graph sparsification in the last decades. In general, existing methods can be roughly divided into two categories [Wu and Chen, 2020]: 1) graph property preserving sparsifiers; and 2) application-oriented sparsifiers. The first category sparsifies a graph by preserving its main properties, such as shortest path distances, graph cuts [Benczúr and Karger, 2015], or graph Laplacian. The most notable example in this category is the spectrum-preserving approach [Spielman and Srivastava, 2011, Spielman and Teng, 2011] that generates a γ-spectral approximation to G such that:

$$\frac{1}{\gamma}\vec{x}^T L_s \vec{x} \leq \vec{x}^T L \vec{x} \leq \gamma \vec{x}^T L_s \vec{x} \quad \text{for all} \quad \vec{x}. \quad (7)$$

Remarkably, authors also proved that every graph G has an $(1+\epsilon)$-spectral approximation G_s with nearly $\mathcal{O}(\frac{N}{\epsilon^2})$ edges. By contrast, the second category targets specific downstream applications, such as community detection Local Similarity (LS) [Satuluri et al., 2011], link prediction [Chen et al., 2015]. Our method belongs to the first category.

The learning-based algorithms for graph sparsification are still less-investigated. GSGAN [Wu and Chen, 2020] is designed mainly for community detection, whereas SparRL [Wickman et al., 2021] uses deep reinforcement learning to sequentially prune edges by preserving the subgraph modularity. Different from GSGAN and SparRL, we demonstrate, in this work, that a sparsified graph can be learned simply by a gradient-based method in a principled (information-theoretic) manner, avoiding the necessity of reinforcement learning or the tuning of a generative adversarial network (GAN) [Goodfellow et al., 2014].

3 PRI FOR GRAPH SPARSIFICATION

3.1 THE LEARNING OBJECTIVE

Suppose we are given a graph G with a known but fixed topology that is characterized by its graph Laplacian ρ, from which we want to obtain a surrogate subgraph G_s with graph Laplacian σ, by preferentially removing less informative (or redundant) edges in G. Motivated by the objective of PRI in Eq. (2), we can cast this problem as a trade-off between the entropy $S(\sigma)$ of G_s and its descriptive power about G in terms of their divergence (or dissimilarity) $D(\sigma||\rho)$:

$$\mathcal{J}_{\text{Graph-PRI}} = \arg\min_{\sigma} S(\sigma) + \beta D(\sigma||\rho), \tag{8}$$

In this paper, we choose von Neumann entropy on the trace normalized graph Laplacian (i.e., $\tilde{\sigma} = \sigma / \operatorname{tr}(\sigma)$ to quantify the entropy of G_s, which is defined on the cone of symmetric positive semi-definite (SPS) matrix with trace 1 as [Nielsen and Chuang, 2002]:

$$S_{\text{vN}}(\tilde{\sigma}) = -\operatorname{tr}(\tilde{\sigma}\log\tilde{\sigma}) = -\sum_i (\lambda_i \log \lambda_i), \tag{9}$$

in which $\log(\cdot)$ is the matrix logarithm, $\operatorname{tr}(\cdot)$ denotes the trace, $\{\lambda_i\}$ are the eigenvalues of $\tilde{\sigma}$.

We then use the quantum Jenssen-Shannon (QJS) divergence between two trace normalized graph Laplacians $\tilde{\sigma}$ and $\tilde{\rho}$ to quantify the divergence between G and G_s [Lamberti et al., 2008]:

$$D_{\text{QJS}}(\tilde{\sigma}||\tilde{\rho}) = S_{\text{vN}}\left(\frac{\tilde{\sigma}+\tilde{\rho}}{2}\right) - \frac{1}{2}S_{\text{vN}}(\tilde{\sigma}) - \frac{1}{2}S_{\text{vN}}(\tilde{\rho}). \tag{10}$$

In this paper, we absorb a scaling constant 2 into the expression for $D_{\text{QJS}}(\tilde{\sigma}||\tilde{\rho})$, the resulting objective combining Eqs. (8)-(10) is given by:

$$\begin{aligned}\mathcal{J}_{\text{Graph-PRI}} &= \arg\min S_{\text{vN}}(\tilde{\sigma}) + \beta D_{\text{QJS}}(\tilde{\sigma}||\tilde{\rho}) \\ &\equiv \arg\min (1-\beta) S_{\text{vN}}(\tilde{\sigma}) + 2\beta S_{\text{vN}}\left(\frac{\tilde{\sigma}+\tilde{\rho}}{2}\right).\end{aligned} \tag{11}$$

The second equation holds for the fact that the extra term $\beta S_{\text{vN}}(\tilde{\rho})$ is a constant with respect to $\tilde{\sigma}$.

3.2 JUSTIFICATION OF THE OBJECTIVE OF GRAPH-PRI

One may ask why we choose the von Neumann entropy in $\mathcal{J}_{\text{graph-PRI}}$. In fact, the Laplacian spectrum contains rich information about the multi-scale structure of graphs [Mohar, 1997]. For example, it is well-known that the second smallest eigenvalue $\lambda_2(L)$, which is also called the algebraic connectivity, is always considered to be a measure of how well-connected a graph is [Ghosh and Boyd, 2006].

On the other hand, it is natural to use the QJS divergence to quantify the dissimilarity between the original graph and its sparsified version. The QJS divergence is symmetric and its square root has also recently been found to satisfy the triangle inequality [Virosztek, 2021]. In fact, as a graph dissimilarity measure, QJS has also found applications in multilayer networks compression [De Domenico et al., 2015] and anomaly detection in graph streams [Chen et al., 2019].

A few recent studies indicate the close connections between $S_{\text{vN}}(L)$ with the structure *regularity* and *sparsity* of a graph [Passerini and Severini, 2008, Han et al., 2012, Liu et al., 2021, Simmons et al., 2018]. We shall now highlight two theorems therein and explain our justifications in Sections 3.2.1 and 3.2.2 in detail.

Theorem 1 ([Passerini and Severini, 2008])**.** *Given an undirected graph $G = \{V, E\}$, let $G' = G + \{u, v\}$, where $V(G) = V(G')$ and $E(G) = E(G') \cup \{u, v\}$, we have:*

$$S_{vN}(L_{G'}) \geq \frac{d_{G'} - 2}{d_{G'}} S_{vN}(L_G), \tag{12}$$

where $d_{G'} = \sum_{v \in V(G')} d(v)$ is the degree-sum *of G', L_G and $L_{G'}$ refer to respectively the graph Laplacians of G and G'.*

Theorem 1 bounds with variation of $S_{\text{vN}}(L)$ with edge addition. Although Eq. (12) does not demonstrate a rigorous monotonic increasing trend for $S_{\text{vN}}(L)$, it still indicates that minimizing $S_{\text{vN}}(L)$ may lead to a sparse graph, especially when the degree-sum is large.

Theorem 2 ([Liu et al., 2021])**.** *For any undirected graph $G = \{V, E\}$, we have:*

$$0 \leq \Delta H(G) = H(G) - S_{vN}(L_G) \leq \frac{\log_2 e}{\delta} \frac{\operatorname{tr}(W^2)}{d_G}, \tag{13}$$

where $H(G) = -\sum_{i=1}^{N}\left(\frac{d_i}{d_G}\right)\log_2\left(\frac{d_i}{d_G}\right)$*, in which* $\delta = \min d_i | d_i > 0$ *is the minimum positive node degree,* d_G *is the* degree-sum*,* W *is the weight matrix.*

Theorem 2 bounds the difference between $S_{\text{vN}}(L)$ and $H(G)$, the Shannon discrete entropy on degree of node.

3.2.1 β controls the sparsity of G_s

Different from the spectral sparsifiers [Spielman and Srivastava, 2011, Spielman and Teng, 2011] in which the sparsity of the subgraph is hard to control (i.e., there is no monotonic relationship between the hyperparameter ϵ and the degree of sparisity as measured by $|E_s|$), we argue that the sparsity of subgraph obtained by Graph-PRI is mainly determined by the value of hyperparameter β.

Our argument is mainly based on Theorem 1. Here, we additionally claim that, under a mild condition (Assumption 1), the QJS divergence $D_{\text{QJS}}(L\|L_s)$ is prone to decrease with edge addition (Corollary 1).

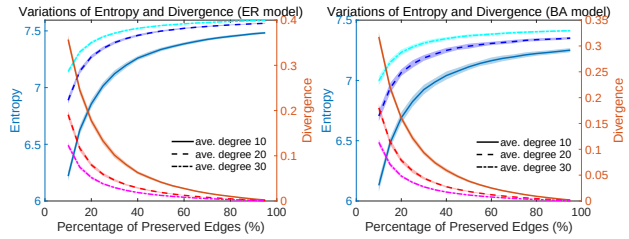
Assumption 1. *Given an undirected graph* $G = \{V, E\}$*, let* $G' = G + \{u, v\}$*, where* $V(G) = V(G')$ *and* $E(G) = E(G') \cup \{u, v\}$*, we have* $S_{vN}(L_{G'}) \geq S_{vN}(L_G)$*, i.e., there exists a strictly monotonically increasing relationship between the number of edges* $|G|$ *and the von Neumann entropy* $S_{vN}(L_G)$*.*

Corollary 1. *Under Assumption 1, suppose* $G_s = \{V_s, E_s\}$ *is a sparse graph obtained from* $G = \{V, E\}$ *(by removing edges), let* $G'_s = G_s + \{u, v\}$*, where* $V(G_s) = V(G'_s)$ *and* $E(G_s) = E(G'_s) \cup \{u, v\}$*, we have* $D_{QJS}(L_{G'_s}\|L_G) \leq D_{QJS}(L_{G_s}\|L_G)$*, i.e., adding an edge is prone to decrease the QJS divergence.*

Combining Theorem 1 and Corollary 1, it is interesting to find that the edge addition has opposite effects on S_{vN} and D_{QJS}: the former is likely to increase whereas the latter will decrease. Therefore, when minimizing the weighted sum of S_{vN} and D_{QJS} together as in Graph-PRI, one can expect the number of edges in G_s is mainly determined by the hyperprameter β: a smaller β gives more weight to S_{vN} and thus encourages a more sparse graph.

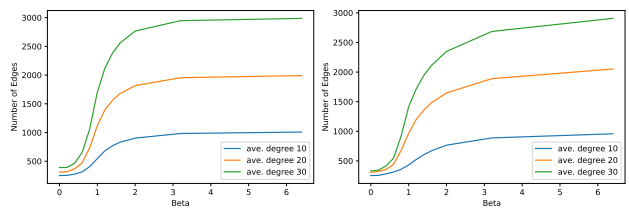
To empirically justify our argument, we generate a set of graphs with 200 nodes by either the Erdös-Rényi (ER) model or the Barabási-Albert (BA) model [Barabási and Albert, 1999]. For both models, we generate the original dense graph G where the average of the node degree $\bar{d}$ is approximately 10, 20 and 30, respectively. We then sparsify G to obtain G_s by the random sparsifier, which satisfy the spectral property, however it is computationally very cheap [Sadhanala et al., 2016].

We finally evaluate the von Neumann entropy of G_s and the QJS divergence between G and G_s with respect to different percentages (pct.) of preserved edges. We repeat the procedure 100 independent times and the averaged results are plotted in Fig. 1, from which we can clearly observe the opposite effects mentioned above. We also sparisify the original graph G by our Graph-PRI with different values of β. The number of preserved edges in G_s with respect to β is illustrated in Fig. 2, from which we can observe an obvious monotonic increasing relationship.



(a) ER model (b) BA model

Figure 1: The variations of entropy and divergence with respect to different percentages of preserved edges.



(a) ER model (b) BA model

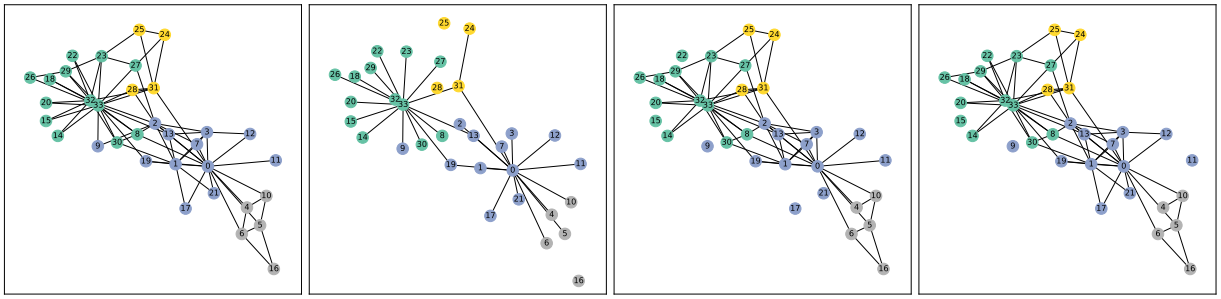
Figure 2: The monotonic increasing relationship between number of preserved edges and hyperparameter β in our Graph-PRI.

3.2.2 Graph-PRI in special cases of β

Continuing our discussion in Sec. 3.2.1, it would be interesting to infer what may happen in some special cases of β. Here, we restrict our discussion with $\beta = 0$ and $\beta \to \infty$.

When $\beta = 0$, our objective can be interpreted as $\min H(G)$. $H(G)$ takes the mathematical form of the Shannon discrete entropy (i.e., $-\sum_{i=1}^{N}\mathbb{P}(x_i)\log\mathbb{P}(x_i)$, in which $\mathbb{P}(x_i)$ is the probability of the i-th state) on the degree of node. In this sense, $H(G)$ reaches to maximum for uniformly distributed degree of node (*i.e.*, $d_1 = \cdots = d_N = k$, which is also called the k-regular graph) and reduces to minimum if the degree of one node dominates (i.e., the graph possesses a high level of centralization). Thus, $S_{\text{vN}}(L)$ can also be interpreted as a measure of degree heterogeneity or graph centrality [Simmons et al., 2018]. It also indicates that minimizing $S_{\text{vN}}(L)$ pushes the Graph-PRI to learn a graph that has more graph centrality. When $\beta \to \infty$, we are expect to recover original graph G by Corollary 1. Fig. 3 corroborates our analysis.

Interestingly, similar properties also hold for the original PRI in scalar random variable setting (see Appendix B).



(a) Karate (original) (b) $\beta = 0$ (c) $\beta = 5$ (d) $\beta = 1000$

Figure 3: Illustration of the sparsified graph structures revealed by our Graph-PRI for (a) Zachary's karate club Zachary [1977]. As the values of β increases, the solution passes through (b) an approximately star graph to the extreme case of (d) $\beta \to \infty$, in which we get back the original graph as the solution.

3.3 OPTIMIZATION

We define a gradient descendent algorithm to solve Eq. (11). As have been discussed in Section 2.1, we have $\rho = BB^T$, $\sigma_{\mathbf{w}} = B\,\mathrm{diag}\,(\mathbf{w})B^T$, and $\mathbf{w}$ is the edge selection vector we aim to optimize. For simplicity, we assume that the selection of edges from G are conditionally independent to each other [Luo et al., 2020], that is $\mathbb{P}_{\mathbf{w}} = \prod_{i=1}^{M} \mathbb{P}_{w_i}$. Due to the discrete nature of G_s, we relax $\mathbf{w} = [w_1, w_2, ..., w_M]$ from a binary vector $\{0,1\}^M$ to a continuous real-valued vector in $[0,1]^M$. In this sense, the value of w_i can be interpreted as the probability of selecting the i-th edge.

In practice, we use the Gumbel-softmax [Maddison et al., 2017, Jang et al., 2016] to update w_i. Particularly, suppose we want to approximate a categorical random variable represented as a one-hot vector in $\mathbb{R}^K$ with category probability $p_1, p_2, \cdots, p_K$ (here, $K = 2$), the Gumbel-softmax gives a K-dimensional sampled vector with the i-th entry as:

$$\hat{p}_i = \frac{\exp\left((\log p_i + g_i)/\tau\right)}{\sum_{j=1}^{K} \exp\left((\log p_j + g_j)/\tau\right)}, \tag{14}$$

where τ is a temperature for the Concrete distribution and g_i is generated from a Gumbel$(0,1)$ distribution:

$$g_i = -\log(-\log u_i), \quad u_i \sim \text{Uniform}(0,1). \tag{15}$$

Note that, although we use the Gumbel-Softmax to ease the optimization, Graph-PRI itself has analytical gradient (Theorem 3). The detailed algorithm of Graph-PRI is elaborated in Appendix E. We also provide a PyTorch example therein.

Theorem 3. *The gradient of Eq. (10) w.r.t. edge selection vector* $\mathbf{w}$ *is:*

$$\nabla_w \mathcal{J}_{\textit{Graph-PRI}} = Ug \tag{16}$$

where $\bar{\sigma}_w = \frac{1}{2}\left(\tilde{\sigma}_w + \tilde{\rho}\right) = \frac{1}{2} B\,\mathrm{diag}\left(\tilde{\mathbf{w}} + \tilde{\mathbf{1}}_M\right) B^T$ *and* $g = -\mathrm{diag}\left(B^T\left[(1-\beta)\ln\tilde{\sigma}_w + \beta \ln \bar{\sigma}_w\right] B\right)$ *and* $U = \{u_{ij}\}, u_{ij} = -\frac{\tilde{w}_j}{1-\tilde{w}_i}, \forall ij | i \neq j, u_{ii} = 1$.

3.4 APPROXIMATION AND CONNECTIVITY CONSTRAINT

The computation of von Neumann entropy requires the eigenvalue decomposition of a trace normalized SPS matrix, which usually takes $\mathcal{O}(N^3)$ time. In this paper, based on the tight bound in Theorem 2, we simply approximate $S_{\text{vN}}(L_G)$ with the Shannon discrete entropy on the normalized degree of nodes $H(G)$, which immediately reduces the computational complexity to $\mathcal{O}(N)$.

On the other hand, when the connectivity of the subgraph is preferred, one can simply add another regularization on the degree of the nodes [Kalofolias, 2016]:

$$\min_{\mathbf{w}} S(\tilde{\sigma}_{\mathbf{w}}) + \beta D(\tilde{\sigma}_{\mathbf{w}} || \tilde{\rho}) - \alpha \mathbf{1}^T \log\left(\mathrm{diag}\left(\sigma\right)\right), \tag{17}$$

where the hyper-parameter $\alpha > 0$. This Logarithm barrier forces the degree to be positive and improves the connectivety of graph without comprising sparsity. Unless otherwise specified, we select $\alpha = 0.005$ throughout this work.

4 EXPERIMENTAL EVALUATION

In this section, we demonstrate the effectiveness and versatility of our Graph-PRI in multiple graph-related machine learning tasks. Our experimental study is guided by the following three questions:

Q1 What kind of structural property or information does our method preserves?

Q2 How well does our method compare against popular and competitive graph sparsification baselines?

Q3 How to use the Graph-PRI in practical machine learning problems; and what are the performance gains?

The selected competing methods include 1 baseline and 3 state-of-the-art (SOTA) ones: 1) the Random Sampling (RS) that randomly prunes a percentage of edges; 2) the Local Degree (LD) [Hamann et al., 2016] that only preserves

the top $|\text{degree}(v)^\alpha|$ ($0 \leq \alpha \leq 1$) neighbors (sorted by degree in descending order) for each node v; 3) the Local Similarity (LS) [Satuluri et al., 2011]) that applies Jaccard similarity function on nodes v and u's neighborhoods to quantify the score of edge (u, v); 4) the Effective Resistance (ER) [Spielman and Srivastava, 2011]. We implement RS, LD, LS by NetworKit[1], and ER by PyGSP[2].

4.1 GRAPH SPARSIFICATION

We use 2 synthetic data and 4 real-world network data from KONECT network datasets[3] for evaluation. They are, **G1**: a k-NN ($k = 10$) graph with 20 nodes that constitute a global circle structure; **G2**: a stochastic block model (SBM) with four distinct communities (30 nodes in each community, and intra- and inter-community connection probabilities of 2^{-2} and 2^{-7}, respectively); **G3**: the most widely used Zachary karate club network (34 nodes and 78 edges); **G4**: a network contains contacts between suspected terrorists involved in the train bombing of Madrid on March 11, 2004 (64 nodes and 243 edges); **G5**: a network of books about US politics published around the time of the 2004 presidential election and sold by the online bookseller Amazon.com (105 nodes and 441 edges); and **G6**: a collaboration network of Jazz musicians (198 nodes and $2,742$ edges).

We expect Graph-PRI to preserve two essential properties associated with the original graph: 1) the spectral similarity (due to the divergence term); and 2) the graph centrality (due to the entropy term). We empirically justify our claims with two metrics. They are, the geodesic distance $d_{\vec{x}}(\rho, \sigma)$ [Bravo Hermsdorff and Gunderson, 2019]:

$$d_{\vec{x}}(\rho, \sigma) = \operatorname{arccosh}\left(1 + \frac{\|(\rho - \sigma)\vec{x}\|_2^2 \|\vec{x}\|_2^2}{2(\vec{x}^T \rho \vec{x})(\vec{x}^T \sigma \vec{x})}\right), \quad (18)$$

in which we select $\vec{x}$ to be the smallest non-trivial eigenvector of the original Laplacian ρ, as it encodes the global structure of a graph; and the graph centralization measure by C_D [Freeman, 1978]):

$$C_D = \frac{\sum_{i=1}^{N} \max(d_j) - d_i}{N^2 - 3N + 2}, \quad (19)$$

in which $\max(d_j)$ refers to the maximum node degree.

We demonstrate in Fig. 4 and Fig. 5 respectively the values of $d_{\vec{x}}(\rho, \sigma)$ and C_D with respect to different edge preserving ratio (i.e., $|E_s|/|E|$) for different sparsification methods. As can be seen, our Graph-PRI always achieves the 2nd best performance across different graphs. Although LD has advantages on preserving spectral distance and graph centrality, it does not have compelling performance in practical applications as will be demonstrated in the next subsection.

[1]https://networkit.github.io/

[2]https://github.com/epfl-lts2/pygsp

[3]http://konect.cc/networks/

4.2 GRAPH-REGULARIZED MULTI-TASK LEARNING

In traditional multi-task learning (MTL), we are given a group of T related tasks. In each task we have access to a training set $\mathcal{D}_t$ with N_t data instances $\{(\mathbf{x}_t^i, y_t^i) : i = 1, \cdots, N_t, t = 1, \cdots, T\}$. In this section, we focus on the regression setup in which $\mathbf{x}_t^i \in \mathcal{X}_t \subseteq \mathbb{R}^d$ and $y_t^i \in \mathbb{R}$. Multi-task learning aims to learn from each training set $\mathcal{D}_t$ a prediction model $f_t(\mathbf{w}_t, \cdot) : \mathcal{X}_t \to \mathbb{R}$ with parameter $\mathbf{w}_t$ such that the task relatedness is taken into consideration and the overall generalization error is small.

In what follows, we assume a linear model in each task, i.e., $f_t(\mathbf{w}_t, \mathbf{x}) = \mathbf{w}_t^T \mathbf{x}$. The multi-task regression problem with a regularization Ω on the model parameters $W = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_T]$ can thus be defined as:

$$\min_W \sum_{t=1}^{T} \|\mathbf{w}_t^T \mathbf{x}_t - y_t\|_2^2 + \gamma \Omega(W). \quad (20)$$

Graph is a natural way to establish the relationship over multiple tasks: each node refers to a single task; if two tasks are strongly correlated to each other, there is an edge to connect them. In this sense, the objective for multi-task regression learning regularized with a graph adjacency matrix A can be formulated as [He et al., 2019]:

$$\min_W \sum_{t=1}^{T} \|\mathbf{w}_t^T \mathbf{x}_t - y_t\|_2^2 + \gamma \sum_{i=1}^{T} \sum_{j \in \mathcal{N}_i} A_{ij} \|\mathbf{w}_i - \mathbf{w}_j\|_2^2, \quad (21)$$

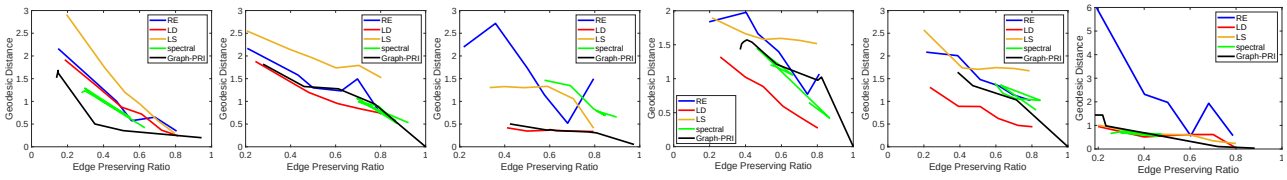
where $\mathcal{N}_i$ is the set of neighbors of i-th task.

Usually, a dense graph G is estimated at first to fully characterize task relatedness [Chen et al., 2010, He et al., 2019]. Here, we are interesting in: 1) sparsifying G to reduce redundant or less-important connections (edges) between tasks; and 2) validating if the sparsified graph can further reduce the generalization error.

To this end, we exemplify our motivation with the recently proposed Convex Clustering Multi-Task regression Learning (CCMTL) [He et al., 2019] that optimizes Eq. (21) with the Combinatorial Multigrid (CMG) solver [Koutis et al., 2011], and test its performance on two benchmark MTL datasets[4]: 1) a synthetic dataset Gonçalves et al. [2016] with 20 tasks in which tasks 1-10 are mutually related and tasks 11-20 are mutually related; 2) a real-world Parkinson's disease dataset[5] which contains biomedical voice measurements from 42 patients. We view each patient as a single task and aim to predict the motor Unified Parkinson's Disease Rating Scale (UPDRS) score based 19-dimensional features such as age, gender, and jitter and shimmer voice measurements.

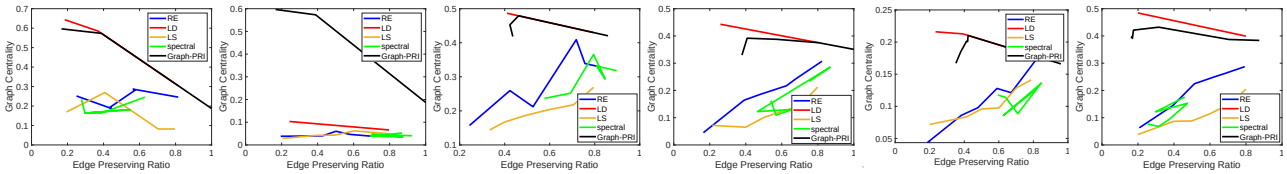
[4]See Appendix C on details of datasets in sections 4.2 and 4.3.

[5]https://archive.ics.uci.edu/ml/datasets/parkinsons+telemonitoring



(a) k-NN (b) SBM (c) Karate club (d) Train bombing (e) US political books (f) Jazz musicians

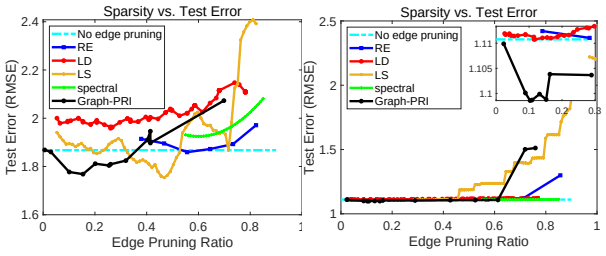
Figure 4: Spectral distance $d_{\vec{x}}(\rho, \sigma)$ (the smaller the better).



(a) k-NN (b) SBM (c) Karate club (d) Train bombing (e) US political books (f) Jazz musicians

Figure 5: Graph centrality C_D (the larger the higher degree of centrality).

We evaluate the test performance with the root mean squared error (RMSE) and demonstrate the values of RMSE with respect to different edge pruning ratio (i.e., $1 - |E_s|/|E|$) of different methods in Fig. 6. In synthetic data, only Graph-PRI and LS are able to further reduce test error. For Graph-PRI, this phenomenon occurs at the beginning of pruning edges, which indicates that our method begins to remove less-informative or spurious connections in an early stage. In Parkinson's data, most of methods obtain almost similar performances to "no edge pruning" (with Graph-PRI performs slightly better as shown in the zoomed plot), which suggests the existence of redundant task relationships. One should note that, the performance of Graph-PRI becomes worse if we remove large amount of edges. One possible reason is that when $|E_s|$ is small, our subgraph tends to have a high graph centrality or star shape, such that one task dominates. Note however that, in MTL, keeping a very sparse relationship is usually not the goal. Because it may lead to weak collaboration between tasks, which violates the motivation of MTL.



(a) Synthetic data (b) Parkinsons's disease data

Figure 6: The RMSE with respect to the degree of sparsity (defined as $1 - |E_s|/|E|$) of the resulting subgraph for all competing methods. Black dashed line indicates performance without any edge pruning. Our method is able to drop out redundant or less-important edges to further reduce generalization error.

4.3 fMRI-DERIVED BRAIN NETWORK CLASSIFICATION AND INTERPRETABILITY

Brain networks are complex graphs with anatomic brain regions of interest (ROIs) represented as nodes and functional connectivity (FC) between brain ROIs as links. For resting-state functional magnetic resonance imaging (rs-fMRI), the Pearson's correlation coefficient between blood-oxygen-level-dependent (BOLD) signals associated with each pair of ROIs is the most popular way to construct FC network [Farahani et al., 2019].

In the problem of brain network classification, the identification of predictive subnetworks or edges is perhaps one of the most important tasks, as it offers a mechanistic understanding of neuroscience phenomena [Wang et al., 2021]. Traditionally, this is achieved by treating all the connections (i.e., the Pearson's correlation coefficients) of FC as a long feature vector, and applying feature selection techniques, such as LASSO [Tibshirani, 1996] and two-sample t-test, to determine if one edge connection is significantly different in different groups (e.g., patients with Alzheimer's disease with respect to normal control members).

In this section, we develop a new graph neural networks (GNNs) framework for interpretable brain network classification that can infer brain network categories and identify the most informative edge connections, in a joint end-to-end learning framework. We follow the motivation of [Cui et al., 2021] and aim to learn a global shared edge mask M to highlight decision-specific prominent brain network connections. The final explanation for an input graph G_i is generated by the element-wise product of A_i and $\sigma(M)$, i.e., $A_i \odot \sigma(M)$, in which A_i is the adjacency matrix of G_i, σ refers to the sigmoid activation function that maps M to $[0, 1]^{N \times N}$. Obviously, $\sigma(M)$ in our GNN also plays a similar role to the edge selection vector $\mathbf{w}$ in Graph-PRI.

Problem definition. Given a weighted brain network $G = (V, E, W)$, where $V = \{v_i\}_{i=1}^N$ is the node set of size N defined by the ROIs, E is the edge set, and $W \in \mathbb{R}^{N\times N}$ is the weighted adjacency matrix describing FC strengths between ROIs, the model outputs a prediction label y. In brain network analysis, N remains the same across subjects.

Experimental data. We evaluate our method on two benchmark real-world brain network datasets. The first one is the eyes open and eyes closed (EOEC) dataset [Zhou et al., 2020], which includes 96 brain networks with the goal to predict either eyes open or eyes closed states. The second one is from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database[6]. We use the brain networks generated by [Kuang et al., 2019], with the task of distinguishing mild cognitive impairment (MCI)[7] group (38 patients) from normal control (NC) subjects (37 in total). Details on brain network construction are elaborated in Appendix C.

Methodology and objective. Following [Cui et al., 2021], we provide interpretability by learning an edge mask $M \in \mathbb{R}^{N\times N}$ that is shared across all subjects to highlight the disease-specific prominent ROI connections. Motivated by the functionality of PRI to prune redundant or less informative edges as demonstrated in previous sections, we train M such that the resulting subgraph $G' = G \odot \sigma(M)$ and the original graph G meets the PRI constraint, i.e., Eq. (8). Therefore, the final objective of our interpretable GNN can be formulated as:

$$\mathcal{L}_{\text{CE}} + \lambda\, \mathbb{E}_{G\sim p(G)} \left\{ S(G') + \beta D(G' \| G) \right\}, \qquad (22)$$

in which $\mathcal{L}_{\text{CE}}$ refers to the supervised cross-entropy loss for label prediction, λ is the hyperparameter that balances the trade-off between $\mathcal{L}_{\text{CE}}$ and PRI constraint.

Empirical results. We summarize the classification accuracy (%) with different methods over 10 independent runs in Table 1, in which Graph-PRI* refers to our objective implemented by approximating von Neumann entropy with Shannon discrete entropy functional on the normalized degree of nodes (see Section 3.4). As can be seen, our method achieves compelling or higher accuracy in both datasets.

To evaluate the interpretability of our method, we visualize the edges been frequently selected for MCI patients and NC group in Fig. 7. We observed that the interactions within sensorimotor cortex (colored blue) for MCI patients are stronger than that of NC group. This result is consistent with the findings in [Ferreri et al., 2016, Niskanen et al., 2011] which observed that the motor cortex excitability is enhanced in AD and MCI from the early stages. We also observed that the interactions within the frontoparietal network (colored yellow) of patients are significantly less than that of NC group, which is in line with previous studies [Neufang et al., 2011, Zanchi et al., 2017] stated that decreased activation in FPN is associated with subtle cognitive deficits.

[6]http://adni.loni.usc.edu/

[7]MCI is a transitional stage between AD and NC.

Table 1: Classification accuracy (%) and standard deviation with different methods over 10 independent runs. The best and second-best performances are in bold and underlined, respectively.

Method	EOEC	ADNI
SVM + t-test	71.79 ± 7.80	60.61 ± 10.52
SVM + LASSO	72.08 ± 7.29	54.67 ± 12.88
GCN [Kipf and Welling, 2017]	68.42 ± 8.59	$\mathbf{66.67} \pm 2.48$
GAT [Veličković et al., 2018]	73.68 ± 8.60	$\mathbf{66.67} \pm 9.43$
Graph-PRI	$\mathbf{80.70} \pm 9.60$	$\mathbf{66.67} \pm 6.67$
Graph-PRI*	$\underline{78.95} \pm 4.30$	$\underline{64.44} \pm 3.14$

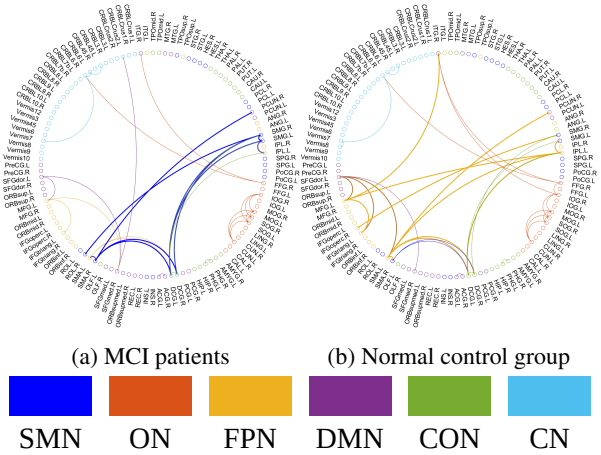


Figure 7: The contributing functional connectivity links for (a) MCI patients; and (b) normal control group. We visualize edges with a probability of more than 50% been selected by our generated edges. The colors of neural systems are described as: sensorimotor network (SMN), occipital network (ON), fronto-parietal network (FPN), default mode network (DMN), cingulo-opercular network (CON), and cerebellum network (CN), respectively. See Appendix F for a zoomed plot.

5 CONCLUSIONS

We present a first study on extending the Principle of Relevant Information (PRI) - a less well-known but promising unsupervised information-theoretic principle - to network analysis and graph neural networks (GNNs). Our Graph-PRI preserves spectral similarity well, while also encouraging the resulting subgraph to have higher graph centrality. Moreover, our Graph-PRI is easy to optimize. It can be flexibly integrated with either multi-task learning or GNNs to improve not only the quantitative accuracy but also interpretability.

In the future, we will explore more unknown properties behind Graph-PRI, including a full understanding to the physical meaning of von Neumann entropy on graphs. We will also investigate more downstream applications of Graph-PRI on GNNs such as node representation learning.

References

Raman Arora and Jalaj Upadhyay. On differentially private graph sparsification and applications. *NeurIPS*, 32:13399–13410, 2019.

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

András A Benczúr and David R Karger. Randomized approximation schemes for cuts and flows in capacitated graphs. *SIAM Journal on Computing*, 44(2):290–319, 2015.

Gecia Bravo Hermsdorff and Lee Gunderson. A unifying framework for spectrum-preserving graph sparsification and coarsening. *NeurIPS*, 32:7736–7747, 2019.

Pin-Yu Chen, Lingfei Wu, Sijia Liu, and Indika Rajapakse. Fast incremental von neumann graph entropy computation: Theory, algorithm, and applications. In *ICML*, 2019.

Xi Chen, Seyoung Kim, Qihang Lin, Jaime G Carbonell, and Eric P Xing. Graph-structured multi-task regression and an efficient optimization method for general fused lasso. *arXiv preprint arXiv:1005.3579*, 2010.

Yi-Ling Chen, Ming-Syan Chen, and S Yu Philip. Ensemble of diverse sparsifications for link prediction in large-scale networks. In *ICDM*, pages 51–60. IEEE, 2015.

Hejie Cui, Wei Dai, Yanqiao Zhu, Xiaoxiao Li, Lifang He, and Carl Yang. Brainnnexplainer: An interpretable graph neural network framework for brain network based disease analysis. *arXiv preprint arXiv:2107.05097*, 2021.

Manlio De Domenico, Vincenzo Nicosia, Alexandre Arenas, and Vito Latora. Structural reducibility of multilayer networks. *Nature communications*, 6:6864, 2015.

Farzad V Farahani, Waldemar Karwowski, and Nichole R Lighthall. Application of graph theory for identifying connectivity patterns in human brain networks: a systematic review. *frontiers in Neuroscience*, 13:585, 2019.

Florinda Ferreri et al. Sensorimotor cortex excitability and connectivity in alzheimer's disease: A tms-eeg co-registration study. *Human brain mapping*, 37(6):2083–2096, 2016.

Linton C Freeman. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239, 1978.

Arpita Ghosh and Stephen Boyd. Growing well-connected graphs. In *CDC*, pages 6605–6611. IEEE, 2006.

André R Gonçalves, Fernando J Von Zuben, and Arindam Banerjee. Multi-task sparse structure learning with gaussian copula models. *The Journal of Machine Learning Research*, 17(1):1205–1234, 2016.

Ian Goodfellow et al. Generative adversarial nets. *NeurIPS*, 27, 2014.

Michael Hamann, Gerd Lindner, Henning Meyerhenke, Christian L Staudt, and Dorothea Wagner. Structure-preserving sparsification methods for social networks. *Social Network Analysis and Mining*, 6(1):22, 2016.

Lin Han, Francisco Escolano, Edwin R Hancock, and Richard C Wilson. Graph characterizations from von neumann entropy. *Pattern Recognition Letters*, 33(15): 1958–1967, 2012.

Xiao He, Francesco Alesiani, and Ammar Shaker. Efficient and scalable multi-task regression on massive number of tasks. In *AAAI*, volume 33, pages 3763–3770, 2019.

J Hoyos-Osorio et al. Relevant information undersampling to support imbalanced data classification. *Neurocomputing*, 436:136–146, 2021.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2016.

Robert Jenssen, Jose C Principe, Deniz Erdogmus, and Torbjørn Eltoft. The cauchy–schwarz divergence and parzen windowing: Connections to graph theory and mercer kernels. *Journal of the Franklin Institute*, 343(6):614–629, 2006.

Vassilis Kalofolias. How to learn a graph from smooth signals. In *AISTATS*, pages 920–929. PMLR, 2016.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

Ioannis Koutis, Gary L Miller, and David Tolliver. Combinatorial preconditioners and multilevel solvers for problems in computer vision and image processing. *CVIU*, 115(12): 1638–1646, 2011.

Liqun Kuang et al. A concise and persistent feature to study brain resting-state network dynamics: Findings from the alzheimer's disease neuroimaging initiative. *Human brain mapping*, 40(4):1062–1081, 2019.

Pedro W Lamberti, Ana P Majtey, A Borras, Montserrat Casas, and A Plastino. Metric character of the quantum jensen-shannon divergence. *Physical Review A*, 77(5): 052311, 2008.

Yanjun Li, Shujian Yu, Jose C Principe, Xiaolin Li, and Dapeng Wu. Pri-vae: principle-of-relevant-information variational autoencoders. *arXiv preprint arXiv:2007.06503*, 2020.

Xuecheng Liu, Luoyi Fu, and Xinbing Wang. Bridging the gap between von neumann graph entropy and structural information: Theory and applications. In *Proceedings of the Web Conference 2021*, pages 3699–3710, 2021.

Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. In *NeurIPS*, volume 33, pages 19620–19631, 2020.

C Maddison, A Mnih, and Y Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *ICLR*, 2017.

Bojan Mohar. Some applications of laplace eigenvalues of graphs. In *Graph symmetry*, pages 225–275. Springer, 1997.

Susanne Neufang et al. Disconnection of frontal and parietal areas contributes to impaired attention in very early alzheimer's disease. *Journal of Alzheimer's Disease*, 25 (2):309–321, 2011.

Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.

Eini Niskanen et al. New insights into alzheimer's disease progression: a combined tms and structural mri study. *PLoS One*, 6(10):e26113, 2011.

Peer Nowack, Jakob Runge, Veronika Eyring, and Joanna D Haigh. Causal networks for climate model evaluation and constrained projections. *Nature communications*, 11(1): 1–11, 2020.

Filippo Passerini and Simone Severini. The von neumann entropy of networks. *arXiv:0812.2597*, 2008.

Jose C Principe. *Information theoretic learning: Renyi's entropy and kernel perspectives*. Springer Science & Business Media, 2010.

Alfréd Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 547–561, 1961.

Mikail Rubinov and Olaf Sporns. Complex network measures of brain connectivity: uses and interpretations. *Neuroimage*, 52(3):1059–1069, 2010.

Veeru Sadhanala, Yu-Xiang Wang, and Ryan Tibshirani. Graph sparsification approaches for laplacian smoothing. In *AISTATS*, pages 1250–1259, 2016.

Venu Satuluri, Srinivasan Parthasarathy, and Yiye Ruan. Local graph sparsification for scalable clustering. In *ACM SIGMOD*, pages 721–732, 2011.

David E Simmons, Justin P Coon, and Animesh Datta. The von neumann theil index: characterizing graph centralization using the von neumann index. *Journal of Complex Networks*, 6(6):859–876, 2018.

Daniel A Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011.

Daniel A Spielman and Shang-Hua Teng. Spectral sparsification of graphs. *SIAM Journal on Computing*, 40(4): 981–1025, 2011.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

Petar Veličković et al. Graph attention networks. In *ICLR*, 2018.

Dániel Virosztek. The metric property of the quantum jensen-shannon divergence. *Advances in Mathematics*, 380:107595, 2021.

Lu Wang, Feng Vankee Lin, Martin Cole, and Zhengwu Zhang. Learning clique subgraphs in structural brain network classification with application to crystallized cognition. *Neuroimage*, 225:117493, 2021.

Ryan Wickman, Xiaofei Zhang, and Weizi Li. Sparrl: Graph sparsification via deep reinforcement learning. *arXiv preprint arXiv:2112.01565*, 2021.

Hang-Yang Wu and Yi-Ling Chen. Graph sparsification with generative adversarial network. In *ICDM*, pages 1328–1333. IEEE, 2020.

Tailin Wu, Hongyu Ren, Pan Li, and Jure Leskovec. Graph information bottleneck. In *NeurIPS*, 2020.

Junchi Yu, Tingyang Xu, Yu Rong, Yatao Bian, Junzhou Huang, and Ran He. Graph information bottleneck for subgraph recognition. In *ICLR*, 2020.

Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473, 1977.

Davide Zanchi et al. Decreased fronto-parietal and increased default mode network activation is associated with subtle cognitive deficits in elderly controls. *Neurosignals*, 25 (1):127–138, 2017.

Zhen Zhou et al. A toolbox for brain network construction and classification (brainnetclass). *Human Brain Mapping*, 41(10):2808–2826, 2020.

A PROOFS

A.1 PROOF TO COROLLARY 1

Before proving Corollary 1, we first present Lemma 1.

Lemma 1. *Let $\boldsymbol{\lambda} = \{\lambda_i\}$ be the eigenvalues of (trace normalized) Graph Laplacian $\tilde{L}$. Suppose the i-th eigenvalue λ_i has a minor and negligible increase and the remaining eigenvalues decrease proportionately to their existing values, such that $\sum_i \lambda_i = 1$. Then, the total derivative of von Neumann entropy $S(\boldsymbol{\lambda}) = -\sum_{i=1}^{N} \lambda_i \log_2 \lambda_i$ with respect to λ_i is given by:*

$$\frac{dS}{d\lambda_i} = \frac{\partial S}{\partial \lambda_i} + \sum_{i \neq j} \frac{\partial S}{\partial \lambda_j} \frac{d\lambda_j}{d\lambda_i} = -\frac{S(\boldsymbol{\lambda}) + \log_2 \lambda_i}{1 - \lambda_i}. \quad (23)$$

Proof. For simplicity, suppose we increase the element λ_i up to the value $\lambda_i + d\lambda_i$ for some infinitesimally small value $d\lambda_i$. As we increase this element, we decrease all the other elements proportionately to their values so that the constraint holds. Thus, for some infinitesimal value δ we update $\boldsymbol{\lambda}$ as:

$$\begin{aligned} \lambda_1 &\mapsto \lambda_1(1-\delta) \\ &\vdots \\ \lambda_{i-1} &\mapsto \lambda_{i-1}(1-\delta) \\ \lambda_{i+1} &\mapsto \lambda_{i+1}(1-\delta) \\ &\vdots \\ \lambda_N &\mapsto \lambda_N(1-\delta) \end{aligned}$$

or in general $\lambda_j \mapsto \lambda_j(1-\delta), j \neq i$. Due to the constraint $\sum_i \lambda_i = 1$, or

$$\begin{aligned} \sum_{j \neq i} \lambda_j(1-\delta) + \lambda_i + d\lambda_i &= 1 \\ (1-\lambda_i)(1-\delta) + \lambda_i + d\lambda_i &= 1 \\ 1 - \lambda_i - (1-\lambda_i)\delta + \lambda_i + d\lambda_i &= 1 \\ -(1-\lambda_i)\delta + d\lambda_i &= 0 \end{aligned}$$

thus we have:

$$\delta = d\lambda_i/(1-\lambda_i). \quad (24)$$

Then,

$$\frac{d\lambda_j}{d\lambda_i} = \frac{-\lambda_j \delta}{d\lambda_i} = -\frac{\lambda_j}{1-\lambda_i}, \quad j \neq i. \quad (25)$$

Therefore, the total derivative of von Neumann entropy $S(\boldsymbol{\lambda}) = -\sum_{i=1}^{N} \lambda_i \log_2 \lambda_i$ with respect to λ_i is given by:

$$\begin{aligned} \frac{dS}{d\lambda_i} &= \frac{\partial S}{\partial \lambda_i} + \sum_{i \neq j} \frac{\partial S}{\partial \lambda_j} \frac{d\lambda_j}{d\lambda_i} \\ &= -(\frac{1}{\ln 2} + \log_2 \lambda_i) + \frac{1}{1-\lambda_i} \sum_{i \neq j} \lambda_j \left(\frac{1}{\ln 2} + \log_2 \lambda_j \right) \\ &= -\frac{(1-\lambda_i)\log_2 \lambda_i}{1-\lambda_i} + \frac{1}{1-\lambda_i} \sum_{i \neq j} \lambda_j \log_2 \lambda_j \\ &= -\frac{1}{1-\lambda_i} \{ \sum_{i=1}^{N} -\lambda_i \log_2 \lambda_i + \log_2 \lambda_i \} \\ &= -\frac{1}{1-\lambda_i} \{ S(\boldsymbol{\lambda}) + \log_2 \lambda_i \} \end{aligned} \quad (26)$$

□

Now, we present proof to Corollary 1.

Proof. Suppose the addition of an edge makes the i-th eigenvalue λ_i has a minor change $\Delta\lambda_i$. By first-order approximation, we have:

$$S_{\text{vN}} \left(\frac{L_G + L_{G'_S}}{2} \right) - S_{\text{vN}} \left(\frac{L_G + L_{G_S}}{2} \right) \approx \frac{1}{2} \frac{dS_{\text{vN}}(L_{\bar{G}})}{d\lambda_i} \Delta\lambda_i, \quad (27)$$

and

$$S_{\text{vN}}(L_{G'_S}) - S_{vN}(L_{G_S}) \approx \frac{dS_{\text{vN}}(L_{G_s})}{d\lambda_i} \Delta\lambda_i, \quad (28)$$

in which $\frac{dS}{d\lambda_i}$ is the total derivative of von Neumann entropy $S(\boldsymbol{\lambda}) = -\sum_{i=1}^{N} \lambda_i \log_2 \lambda_i$ with respect to λ_i.

By Assumption 1, we have $S_{\text{vN}}(L_{\bar{G}}) \geq S_{\text{vN}}(L_{G_s})$. Thus, according to Lemma 1, we obtain:

$$\frac{dS_{\text{vN}}(L_{\bar{G}})}{d\lambda_i} \leq \frac{dS_{\text{vN}}(L_{G_s})}{d\lambda_i}. \quad (29)$$

Combining Eqs. (27) to (29), we get:

$$\begin{aligned} &S_{\text{vN}} \left(\frac{L_G + L_{G'_S}}{2} \right) - S_{\text{vN}} \left(\frac{L_G + L_{G_S}}{2} \right) \\ &\leq \frac{1}{2} \left(S_{\text{vN}}(L_{G'_S}) - S_{\text{vN}}(L_{G_S}) \right). \end{aligned} \quad (30)$$

Thus,

$$\begin{aligned} &S_{\text{vN}} \left(\frac{L_G + L_{G'_S}}{2} \right) - \frac{1}{2} \left(S_{\text{vN}}(L_{G'_S}) + S_{vN}(L_G) \right) \\ &\leq S_{\text{vN}} \left(\frac{L_G + L_{G_S}}{2} \right) - \frac{1}{2} \left(S_{\text{vN}}(L_{G_S}) + S_{vN}(L_G) \right), \end{aligned} \quad (31)$$

which completes the proof.

□

A.2 PROOF TO THEOREM 3

Proof of Theorem 3. Theorem 3 follows by definition of Eq. (10) and substituting the definition of Eq. (10) and the use of result from Theorem 4. The total derivative of the cost function w.r.t. to the normalized selection vector $\tilde{\mathbf{w}}$, is given by $\frac{dJ}{dw_i} = \frac{\partial J}{\partial w_i} + \sum_{i \neq j} \frac{\partial J}{\partial w_j} \frac{dw_j}{dw_i}$. With the normalized selector vector, we have that $\sum_k w_k = 1$ before and after the change. If we consider, $w_i \to w_i + \delta$ and $w_j \to w_j(1-\gamma), j \neq i$, then $\gamma = \frac{\delta}{1-w_i}$ and $\frac{dw_j}{dw_i} = -\frac{\gamma w_j}{\delta} = -\frac{w_j}{1-w_i}$ □

Theorem 4. *The gradient of the von Neumann entropy w.r.t. the edge selection vector* $\mathbf{w}$ *is*

$$\nabla_{\mathbf{w}} S(\sigma_{\mathbf{w}}) = -\operatorname{diag}\left(B^T \log(B \operatorname{diag}(\mathbf{w}) B^T) B\right) - \mathbf{1}_M, \tag{32}$$

where $S(\sigma) = -\operatorname{tr}(\sigma \log \sigma)$ *and* $\sigma_w = B \operatorname{diag}(\mathbf{w}) B^T$.

Proof of Theorem 4. Theorem 4 follows from $\nabla_\sigma S(\sigma) = -\log \sigma - I$ and the use of gradient of the trace of a function of a matrix. Here we used the un-normalized Laplacian matrix. □

B PRINCIPLE OF RELEVANT INFORMATION (PRI) FOR SCALAR RANDOM VARIABLES

In information theory, a natural extension of the well-known Shannon's entropy is the Rényi's α-entropy [Rényi, 1961]. For a random variable $\mathbf{X}$ with PDF $f(x)$ in a finite set $\mathcal{X}$, the α-entropy of $H(\mathbf{X})$ is defined as:

$$H_\alpha(f) = \frac{1}{1-\alpha} \log \int_{\mathcal{X}} f^\alpha(x) dx. \tag{33}$$

On the other hand, motivated by the famed Cauchy-Schwarz (CS) inequality:

$$\left| \int f(x) g(x) dx \right|^2 \leq \int \mid f(x) \mid^2 dx \int \mid g(x) \mid^2 dx, \tag{34}$$

with equality if and only if $f(x)$ and $g(x)$ are linearly dependent (e.g., $f(x)$ is just a scaled version of $g(x)$), a measure of the "distance" between the PDFs can be defined, which was named the CS divergence [Jenssen et al., 2006], with:

$$\begin{aligned} D_{cs}(f \| g) &= -\log(\int fg)^2 + \log(\int f^2) + \log(\int g^2) \\ &= 2H_2(f;g) - H_2(f) - H_2(g), \end{aligned} \tag{35}$$

the term $H_2(f;g) = -\log \int f(x) g(x) dx$ is also called the quadratic cross entropy [Principe, 2010].

Combining Eqs. (33) and (35), the PRI under the 2-order Rényi entropy can be formulated as:

$$\begin{aligned} f_{\text{opt}} &= \arg\min_f H_2(f) + \beta(2H_2(f;g) - H_2(f) - H_2(g)) \\ &\equiv \arg\min_f (1-\beta) H_2(f) + 2\beta H_2(f;g), \end{aligned} \tag{36}$$

the second equation holds because the extra term $\beta H_2(g)$ is a constant with respect to f.

As can be seen, the objective of naïve PRI for $i.i.d.$ random variables (i.e., Eq. (36)) resembles its new counterpart on graph data (i.e., Eq. (11)). The big difference is that we replace $H_2(f)$ with $S_{\text{vN}}(\tilde{\sigma})$ and $H_2(f;g)$ with $S_{\text{vN}}\left(\frac{\tilde{\sigma}+\tilde{\rho}}{2}\right)$ to capture structure information.

If we estimate $H_2(f)$ and $H_2(f;g)$ with the Parzen-window density estimator and optimize Eq. (36) by gradient descent. Fig. 8 demonstrates the structure learned from an original intersect data by different values of β.

Interestingly, when $\beta = 0$, we obtained a single point, very similar to what happens for Graph-PRI that learns a nearly star graph such that edges concentrates on one node. Similarly, when $\beta \to \infty$, both naïve PRI and Graph-PRI get back to the original input as the solution.

C DETAILS OF USED DATASETS IN SECTION 4.2 AND SECTION 4.3

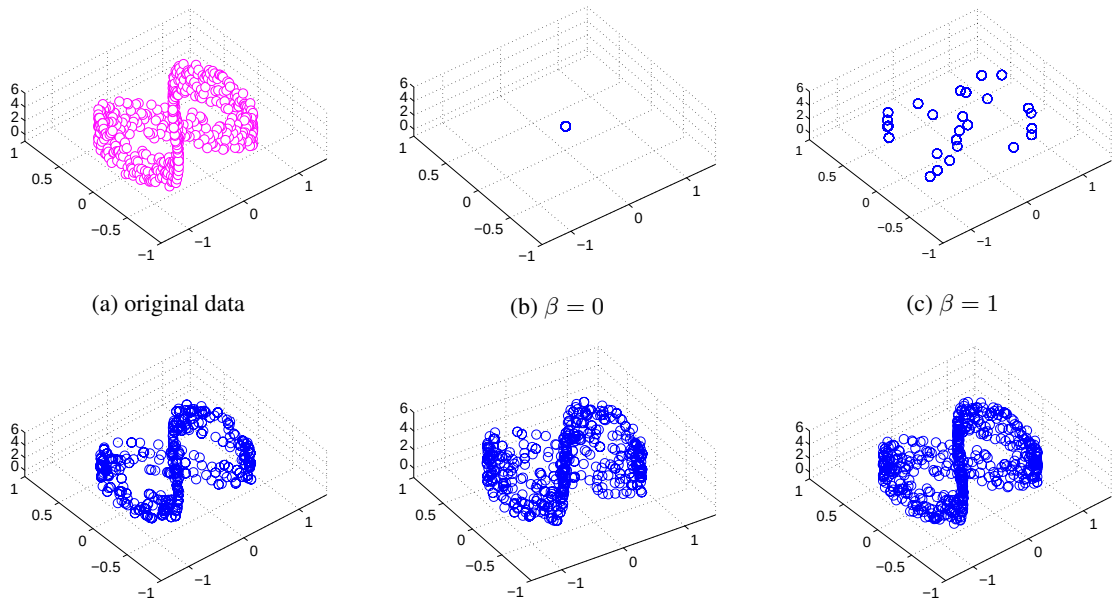
C.1 MULTI-TASK LEARNING

Synthetic data. This dataset consists of 20 regression tasks with 100 samples each. Each task is a 30-dimensional linear regression problem in which the last 10 variables are independent of the output variable y. The 20 tasks are related in a group-wise manner: the first 10 tasks form a group and the remaining 10 tasks belong to another group. Tasks' coefficients in the same group are completely related to each other, while totally unrelated to tasks in another group.

Tasks' data are generated as follows: weight vectors corresponding to tasks 1 to 10 are $\mathbf{w}_k = \mathbf{w}_a \odot \mathbf{b}_k + \xi$, where $\odot$ is the element-wise Hadamard product; and tasks 11 to 20 are $\mathbf{w}_k = \mathbf{w}_b \odot \mathbf{b}_k + \xi$, where $\xi \sim \mathcal{N}(\mathbf{0}, 0.2\mathbf{I}_{20})$. Vectors $\mathbf{w}_a$ and $\mathbf{w}_b$ are generated from $\mathcal{N}(\mathbf{0}, \mathbf{I}_{20})$, while $\mathbf{b}_k \sim \mathcal{U}(0,1)$ are uniformly distributed 20-dimensional random vectors.

Input and output variables for the t-th ($t = 1, \cdots, 20$) task, X_t and y_t, are generated as $X_t' \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{20})$ and $y_t = X_t' \mathbf{w}_t + \mathcal{N}(0,1)$. 10-dimensional unrelated variables $X_t'' \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{10})$ are then concatenated to X_t' to form the final input data $X_t = [X_t' \quad X_t'']$.

Parkinsons's disease dataset. This is a benchmark multi-task regression data set, comprising a range of biomedical voice measurements taken from 42 patients with earlystage Parkinson's disease. For each patient, the goal is to predict



(a) original data (b) $\beta = 0$ (c) $\beta = 1$

(d) $\beta = 3$ (e) $\beta = 6$ (f) $\beta = 100$

Figure 8: Illustration of the structures revealed by the naïve PRI for (a) Intersect data set. As the values of β increase the solution passes through (b) a single point, (c) modes, (d) and (e) principal curves at different dimensions, and in the extreme case of (f) $\beta \to \infty$ we get back the data themselves as the solution.

the motor Unified Parkinson's Disease Rating Scale (UPDRS) score based 18-dimensional record: age, gender, and 16 jitter and shimmer voice measurements. For the categorical variable "gender", we applied label encoding that converts genders into a numeric representation. We treat UPDRS prediction for each patient as a task, resulting in 42 tasks and $5,875$ observations in total.

C.2 BRAIN NETWORK CLASSIFICATION

For both datasets, the Automated Anatomical Labeling (AAL) template was used to extract ROI-averaged time series from the 116 ROIs. Meanwhile, to construct the initial brain network topology (i.e., the adjacency matrix A), we only keep edge if its weight (i.e., the absolute correlation coefficient) is among the top 20% of all absolute correlation coefficients in the network.

As for the node features, we only use the correlation coefficients for simplicity. That is, the node feature for node i can be represented as $\mathbf{x}_i = [\rho_{i1}, \rho_{i2}, \cdots, \rho_{in}]^T$, in which ρ_{ij} is the Pearson's correlation coefficient for node i and node j. One can expect performance gain by incorporating more discriminative network property features such as the local clustering coefficient [Rubinov and Sporns, 2010], although this is not the main scope of our work.

The first one is the eyes open and eyes closed (EOEC) dataset [Zhou et al., 2020], which contains the rs-fMRI data of 48 (22 females) college students (aged 19-31 years) in both eyes open and eyes closed states. The task is to predict two states based on brain network FC.

The second one is from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database[8]. We use the rs-fMRI data collected and preprocessed in [Kuang et al., 2019] which includes 31 AD patients aged 60–90 years. They were matched by age, gender, and education to mild cognitive impairment (MCI)[9] and 37 normal control (NC) subjects, together comprising 106 participants been selected. In this work, we only focus on distinguishing MCI group from NC group.

EOEC is publicly available from https://github.com/zzstefan/BrainNetClass/.

ADNI preprocessed by [Kuang et al., 2019] is publicly available from http://gsl.lab.asu.edu/software/ipf/.

D NETWORK ARCHITECTURE AND HYPERPARAMETER TUNING

D.1 FMRI-BASED BRAIN NETWORK CLASSIFICATION

The classification problem is solved using graph neural networks composed of two graph convolutional networks of

[8] http://adni.loni.usc.edu/

[9] MCI is a transitional stage between AD and NC.

size 32 and with relu activation function. We also use node feature drop with probability 10^{-1}. The node pooling is the sum of the node features, while the node classification minimizes the cross entropy loss. Hyper parameter search is applied to all method with time budget of $3'000$ seconds, over 3 runs. The learning rates, λ, β and the softmax temperature are optimized using early pruning. Each graph neural network is fed with graphs generated from the full correlation matrix by selecting edges among the strongest 20% absolute correlation values. For the Graph-PRI method, we used the GCN [Kipf and Welling, 2017] as graph classification network.

For SVM, we use the Gaussian kernel and set kernel size equals to 1. For LASSO, we set the hyperparameter as 0.1. For t-test, we set the significance level as 0.05.

E MINIMAL IMPLEMENTATION OF GRAPH-PRI IN PYTORCH

Algorithm 1 PRI for Graph Sparsification

Input: $\rho = BB^T$, β, learning rate η, number of samples S
Output: $\sigma_{\mathbf{w}}$

$B \leftarrow$ incident matrix of ρ;
Initialize $\theta = \{\theta_1, \theta_2, \cdots, \theta_M\}$;
while not converged **do**
  $L = 0$;
  for $i = 1, 2, \cdots, S$ **do**
    $\mathbf{w}^i \leftarrow \text{GumbelSoftmax}(\theta)$;
    $\sigma_{\mathbf{w}^i} = B\,\text{diag}\,(\mathbf{w}^i)B^T$;
    $L = L + \mathcal{J}_\beta(\rho, \sigma_{\mathbf{w}^i})$;
  end for
  $L = \frac{1}{S}L$
  $\theta \leftarrow \theta - \eta\nabla_\theta L$;
end while
$\mathbf{w} \leftarrow \text{GumbelSoftmax}(\theta)$;
return $\sigma_{\mathbf{w}} = B\,\text{diag}\,(\mathbf{w})B^T$;

We additional provide PyTorch implementation of Graph-PRI.

```python

import torch
import networkx as nx
import numpy as np

def vn_entropy(k, eps=1e-20):

    k = k / torch.trace(k)
    eigv = torch.abs(torch.symeig(k, eigenvectors=True)[0])
    entropy = -torch.sum(eigv[eigv>0] * torch.log(eigv[eigv>0] + eps))
    return entropy

def entropy_loss(sigma, rho, beta):

    assert(beta>=0), "beta shall be >=0"
    if beta > 0:
        return 0.5 * (1 - beta) / beta * vn_entropy(sigma) + vn_entropy(0.5 * (sigma + rho))
    else:
        return vn_entropy(sigma)

def sparse(G, tau, n_samples, max_iteration, lr, beta):
  '''
  Args:
    G: networkx Graph
    n_samples: number of samples for gumbel softmax
  '''

  E = nx.incidence_matrix(g1, oriented=True)
    E = E.todense().astype(np.double)
    E = torch.from_numpy(E)

    rho = E @ E.T

    m, n = G.number_of_edges(), G.number_of_nodes()
    theta = torch.randn(m, 2, requires_grad=True)
    optimizer = torch.optim.Adam([theta], lr=lr)

    for itr in range(max_iteration):
        cost = 0
        for sample in range(n_samples):
            # Sampling
            z = F.gumbel_softmax(theta, tau, hard = True)
            w = z[:, 1].squeeze()
            sigma = E @ torch.diag(w) @ E.T
            _loss = entropy_loss(sigma, rho, beta)
            cost = cost + _loss

        cost = cost / n_samples
        cost.backward()
        optimizer.step()
        optimizer.zero_grad()

    z = F.gumbel_softmax(theta, tau, hard=True)
    w = z[:,1].squeeze()

    sigma = E @ torch.diag(w) @ E.T # sparse laplacian

    return sigma, w
```

Listing 1: Graph-PRI PyTorch

F ZOOMED PLOT OF FIG. 7

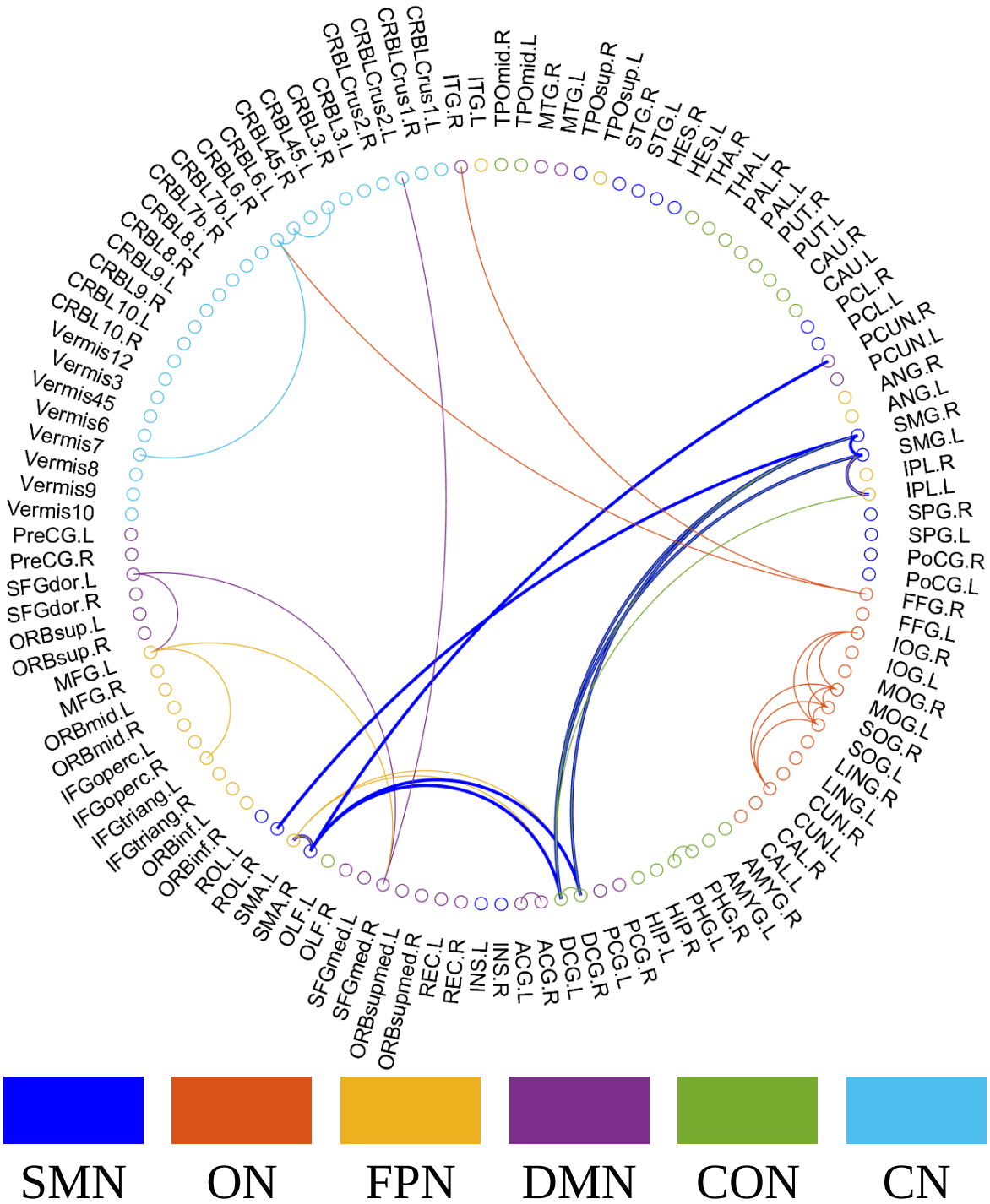


Figure 9: The contributing functional connectivity links for MCI patients. We visualize edges with a probability of more than 50% been selected by our generated edges. The colors of neural systems are described as: sensorimotor network (SMN), occipital network (ON), fronto-parietal network (FPN), default mode network (DMN), cingulo-opercular network (CON), and cerebellum network (CN), respectively.

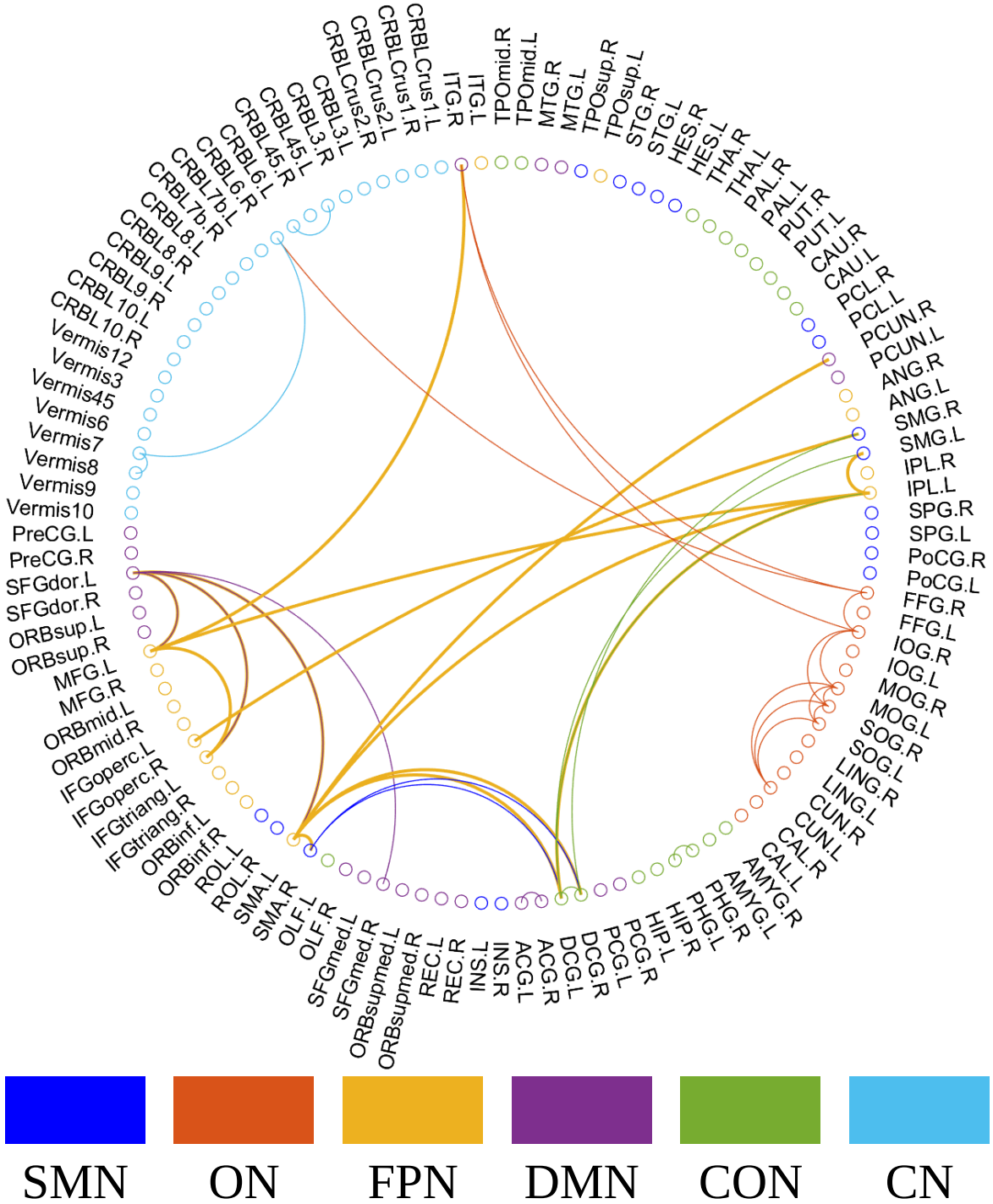


Figure 10: The contributing functional connectivity links for normal control group. We visualize edges with a probability of more than 50% been selected by our generated edges. The colors of neural systems are described as: sensorimotor network (SMN), occipital network (ON), fronto-parietal network (FPN), default mode network (DMN), cingulo-opercular network (CON), and cerebellum network (CN), respectively.