TOWARDS A MORE STABLE AND GENERAL SUBGRAPH INFORMATION BOTTLENECK

Hongzhi Liu[1] *Kaizhong Zheng*[1] *Shujian Yu*[2*] *Badong Chen*[1*†]

[1]Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
[2]Department of Computer Science, Vrije Universiteit Amsterdam

ABSTRACT

Graph Neural Networks (GNNs) have been widely applied to graph-structured data. However, the lack of interpretability impedes its practical deployment especially in high-risk areas such as medical diagnosis. Recently, the Information Bottleneck (IB) principle has been extended to GNNs to identify a compact subgraph that is most informative to class labels, which significantly improves the interpretability on decision. However, existing Graph Information Bottleneck (GIB) models are either unstable during the training (due to the difficulty of mutual information estimation) or only focus on a special kind of graph (e.g., brain networks) that suffer from poor generalization to general graph datasets with varying graph sizes. In this work, we extend the recently developed Brain Information Bottleneck (BrainIB) to general graphs by introducing matrix-based Rényi's α-order mutual information to stablize the training; and by designing a novel mask strategy to deal with varying graph sizes such that the new method can also be used for social networks, molecules, etc. Extensive experiments on different types of graph datasets demonstrate the superior stability and generality of our model.

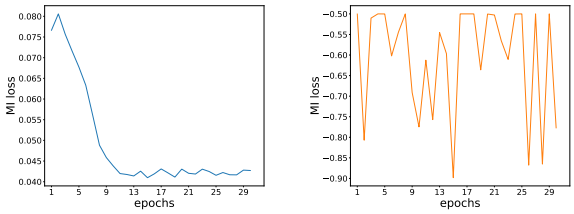
***Index Terms*—** Graph Information Bottleneck, Generalization, Stability

1. INTRODUCTION

Graph Neural Networks (GNNs), are becoming dominating tools in modern artificial intelligence applications for analyzing graph-structured data, such as fMRI-based brain networks [1], social networks [2], chemical molecules [3], traffic trajectories [4], etc. Despite the appealing performance of GNNs, they lack humanly comprehensible explanations [5]. Thus, it is difficult or even dangerous to apply GNNs to high-risk real-world applications [1].

So far, GNN explanation has gained tremendous attention. According to a recent survey [6], perturbation-based approach is the most popular category which employs distinct subgraph generators to sample important structure from input graph. Notable examples in this category include GNNExplainer [7], PGExplainer [8], and the recently developed DIR-GNN [9] that leverages the ideas in causality.

Recently, the famed Information Bottleneck [10] principle has been applied to GNNs to improve their interpretability and generalization [11]. Yu et al. [12] propose Subgraph Information Bottleneck (SIB), which identifies predictive subgraphs. However, their model is prone to unstable training due to mutual information estimation based on variational lower bounds. BrainIB [1] applies the matrix-based Rényi's α-order mutual information to address this issue, but it is specific to networks in which the number of nodes (i.e., cortical or subcortical regions in brain template) is fixed and the edge attribute remains the same. This hinders the application of BrainIB to other kinds of graphs with varying sizes or have specific edge attributes.



(a) Matrix-based Rényi's α-order mutual information (b) Mutual Information Neural Estimator (MINE)

Fig. 1. Curves of $I(G; G_{\text{sub}})$ over epochs on MUTAG. Curves in blue and orange show $I(G; G_{\text{sub}})$ based on matrix-based Rényi's α-order mutual information [13] and Mutual Information Neural Estimator (MINE) [14], respectively.

In this paper, we aim to close the performance gap between graph information bottleneck and wide-spread graph classification applications. Our main contributions include:

- Instead of using variational approximation, we use the matrix-based Rényi's α-order mutual information to stabilize the training (see Fig. 1).
- We transform the problem of discovering informative subgraph into a edge selection task, and design a novel mask strategy to deal with arbitrary graph sizes.

*To whom correspondence should be addressed (yusj9011@gmail.com; chenbd@mail.xjtu.edu.cn).

†This work was funded by the National Natural Science Foundation of China with grant numbers (U21A20485, 61976175).

- Extensive experiments on molecular, social network and brain disorder datasets show improvements on classification performance and interpretability.

2. RELATED WORK

2.1. Graph Neural Networks and Graph Classification

Graph Neural Networks (GNNs) can perform classification tasks such as nodes, edges and graphs classification [15]. Given a graph $G = \{\mathbb{V}; \mathbb{E}\}$, in which $\mathbb{V} = \{V_i \mid i \in \{1, ..., n\}\}$, $\mathbb{E} = \{(V_i, V_j) \mid V_i, V_j \in \mathbb{V}\}$ denote nodes and edges set of G separately and n represents the number of nodes. The main task of GNN is obtaining the prediction of the graph obtained by this graph feature. Message passing process can be expressed by the following formula:

$$\mathbf{x}_i^{(k)} = \gamma^{(k)} \left(\mathbf{x}_i^{(k-1)}, f_{j \in \mathcal{N}(i)} \left(\phi^{(k)} \left(\mathbf{x}_i^{(k-1)}, \mathbf{x}_j^{(k-1)}, \mathbf{e}_{j,i} \right) \right) \right), \tag{1}$$

where $\mathbf{x}_i^{(k)}$ denotes node embeddings of node V_i in layer k , $e_{j,i}$ denotes edge features from node V_j to node V_i and $\mathcal{N}(i)$ denotes neighborhood set of nodes i. $\gamma^{(k)}(.)$ and $\phi^{(k)}(.)$ are differentiable functions meanly using MLPs, $f(.)$ denotes differentiable and permutation invariant function such as sum [16], mean [17] or max [18].

In the problem of graph classification, we need to learn the representation g of entire graph G from node embeddings $\mathbf{x}_i^{(k)}$ by READOUT function:

$$g = \text{READOUT} \left(\left\{ x_i^{(k)} \mid V_i \in \mathbb{V} \right\} \right), \tag{2}$$

Typical READOUT functions include extracting information from nodes directly [19, 20] and hierarchically [21, 22, 23].

2.2. Information Bottleneck Meets GNN

For a fixed but unknown distribution $p(x, y)$, the main idea of the information bottleneck principle is to extract from the input X a latent representation Z that contains the maximum information of the output result Y under the condition that the compression rate is specified [10]. This process can be described by the following formula:

$$\min_{p(z|x)} -I(Y; Z) + \beta I(X; Z), \tag{3}$$

where $I(\cdot)$ denotes mutual information, and β denotes Lagrange multiplier that controls the trade-off between the minimality of the representation to the input (as measured by $I(X; Z)$) and the maximality of the representation Z to the prediction of label Y (as evaluated by $I(Y; Z)$).

Subgraph Information Bottleneck (SIB) [12] aims to obtain the most informative subgraph by optimizing Eq. (3) as:

$$\min_{G_{\text{sub}}} -I(Y; G_{\text{sub}}) + \beta I(G; G_{\text{sub}}). \tag{4}$$

SIB shows that $-I(Y; G_{\text{sub}})$ equals to minimizing the cross entropy and identifies the predictive subgraph through discarding redundant nodes. Furthermore, GSAT [24] leverages stochastic attention mechanism to obtain the faithful subgraph. These Graph Information Bottlenecks (GIBs) use the Mutual Information Neural Estimator (MINE) [14] to evaluate $I(G; G_{\text{sub}})$, which leads to high instability during the training. Zheng et al. use the matrix-based Rényi's α-order mutual information [13] rather than MINE to estimate $I(G; G_{\text{sub}})$, which make training stable. However, BrainIB is proposed for the brain disease datasets where the number of nodes are fixed and edge attributes remain the same. This impedes the generalization of BrainIB to different graph datasets.

3. PROPOSED MODEL

3.1. Overall Structure of BrainIB

Zheng et al. [1] propose BrainIB which generates subgraphs based on edges. There are three main modules in the BrainIB framework: ***graph encoder*** obtains node and graph embeddings from input graph G; ***mutual information estimator*** calculates mutual information $I(G; G_{\text{sub}})$ between graph G and subgraph G_{sub} and ***subgraph generator*** obtains the interpretation subgraph G_{sub} by sampling edges from input graph G. The relationship among three modules in the framework is briefly shown in the Fig. 2.

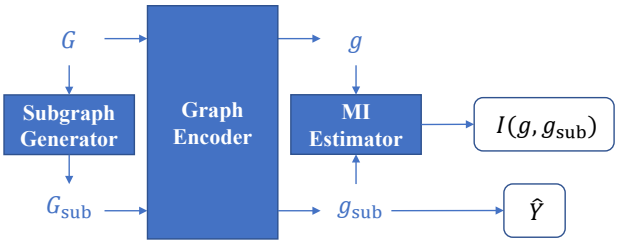


Fig. 2. Architecture of BrainIB consists of three part: ***subgraph generator***, ***graph encoder*** and ***mutual information estimator***.

3.2. Our Model

Our approach takes inspiration from BrainIB's framework and refines it so that it can be applied to a wider range of datasets. We describe the specific implementation of each part as follows.

3.2.1. Graph Encoder

Graph encoder aims to encode graph-structured data $G = \{X, A\}$ into a vectorized embedding, where X is node feature and A is the graph adjacency matrix. Node embedding

$X^{(L)}$ are firstly updated from input node features X by the L-th layer of GCN encoder [25], which can be described as:

$$X^{(L)} = (\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}})^{L} X^{(k)}\Theta_1 ... \Theta_L, \quad (5)$$

where $\tilde{D}$ denotes diagonal degree matrix, and $\tilde{A}$ denotes the adjacency matrix with inserted self-loops. Then we use bi-linear mapping second-order pooling ($\text{SOPOOL}_{\text{bi-linear}}$) [26] and flatten into a vectorized embedding g_n:

$$g_n = \text{flatten}\left(\text{SOPOOL}_{\text{bi-linear}}\left(X^{(L)}\right)\right), \quad (6)$$

3.2.2. *Mutual Information Estimator*

Mutual information estimator is used to evaluate $I(G; G_{\text{sub}})$. Previous works apply MINE to estimate $I(G; G_{\text{sub}})$ [12, 24], which is defined as:

$$\begin{aligned} I(G, G_{\text{sub}}) = &\sup_{f(G,G_{sub})} \mathbb{E}_{G,G_{sub}\sim p(G,G_{\text{sub}})} f(G, G_{\text{sub}}) \\ &- \log \mathbb{E}_{G\sim p(G), G_{\text{sub}}\sim p(G_{\text{sub}})} e^{f(G,G_{\text{sub}})}, \end{aligned} \quad (7)$$

where $f(\cdot)$ is a linear network. Due to the application of f, $I(G; G_{\text{sub}})$ is unstable during the training or even lead to negative mutual information values [27]. Thus, we use the matrix-based Rényi's α-order mutual information proposed by Yu et al. [13] which is mathematically well-defined and computationally efficient to address this issue. Specifically, this estimator can be easily calculated without any auxiliary neural network, which suits well for deep learning applications [1]. To calculate $I(G; G_{\text{sub}})$, we first extract the graph embeddings g and g_{sub} from G and G_{sub} through the graph encoder. Due to the sufficient encoder assumption [28], we approximate $I(G; G_{\text{sub}})$ with $I(g; g_{\text{sub}})$ in this study. Thus, the matrix-based information $I(g; g_{\text{sub}})$ in analogy of Shannon's could be formulated as:

$$I(g, g_{\text{sub}}) = H_\alpha(g) + H_\alpha(g_{\text{sub}}) - H_\alpha(g, g_{\text{sub}}). \quad (8)$$

where the calculation processes of $H_\alpha(g)$, $H_\alpha(g_{\text{sub}})$ and $H_\alpha(g, g_{\text{sub}})$ are provided in the supplementary material (`https://github.com/SJYuCNEL/brain-and-Information-Bottleneck`).

3.2.3. *Subgraph Generator*

Due to the limitation of edge assignment extracted by BrainIB, we propose a novel subgraph generator architecture generalizing to more graph datasets as shown in the Fig. 3.

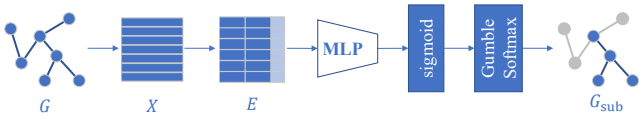


Fig. 3. Subgraph generator structure. Subgraph G_{sub} is generated from the input graph G with the edge selection strategy. Here, X represents the node feature and E is edge feature.

We learn a probability p_{ij} over edge e_{ij} that connects node i and node j, where p_{ij} refers to edge selection probability. Specifically, we use GumbleSoftmax [29] and a linear network MLP to obtain the probability p_{ij} between node i and j [30]:

$$p_{ij} = \text{GumbleSoftmax}\left(\text{sigmoid}\left(\text{MLP}\left([x_i; x_j; e_{ij}]\right)\right)\right), \quad (9)$$

where $[\cdot;\cdot;\cdot]$ is concatenation operation, x_i, x_j are node embeddings obtained from graph encoder, e_{ij} is edge attribute of the input graph. Note that e_{ij} can be any size, when $e_{ij} \in \emptyset$, edge features can be updated as $[x_i; x_j]$, which makes our model generalizing to more graph datasets.

4. EXPERIMENT

4.1. Experimental Datasets and Baselines

Three molecular datasets (MUTAG [31], PROTEINS [32] and NCI-1 [33]), one social networks dataset (IMDB-BINARY [34]) and one brain disorder dataset (ABIDE I) (`http://fcon_1000.projects.nitrc.org/indi/abide`) are chose in the experiments. In addition, we compared our model with baseline: Graph Convolutional Network (GCN) [25], Graph Isomorphic Network (GIN) [16] and Graph ATtention network (GAT) [17] and three state-of-the-art (SOTA) GNN explainers: Subgraph Information Bottleneck (SIB) [12], GNNExplainer [7] and PGExplainer [8]. Details of the used datasets, the competing methods, and the hyperparameter tuning are specified in the supplementary material.

4.2. Performance Evaluation

We present the average results and variance of accuracy in ten-fold validation shown in the Table 1. It can be seen that our model outperforms baselines on both molecular (MUTAG, PROTEINS and NCI-1), social network (IMDB-BINARY) and brain (ABIDE I) datasets. This demonstrates that our improvement successfully generalizes to versatile graph data with different properties. Compared to the SOTA GNN explainers (SIB, GNNExplainer and PGExplainer) we are able to get at least 5.9%, 4.2%, 7.3%, 3.9% and 3.8% performance improvement in four datasets separately.

Method	DATASET				
	MUTAG	PROTEINS	NCI-1	IMDB-BINARY	ABIDE I
GCN	0.830	0.774	0.761	0.766	0.666
GIN	**0.920**	0.768	0.729	0.562	0.638
GAT	0.825	0.766	0.732	0.761	0.668
SIB	0.798	0.721	0.700	0.720	0.601
GNNExplainer	0.820	0.732	0.684	0.720	0.633
PGExplainer	0.830	0.733	0.645	0.737	0.624
Ours	0.889	**0.775**	**0.773**	**0.776**	**0.671**

Table 1. 10-fold validation result. The standard deviation (STD) is omitted, since all methods have STD less than 0.005.

4.3. Interpretability Exploration

In order to access the interpretability of our model, we validate the interpretable performance of the model on MUTAG and interpretable subgraph on MUTAG and ABIDE-I.

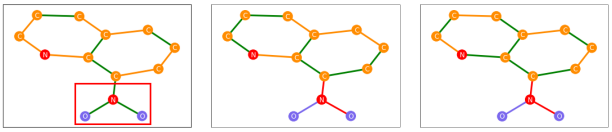
	acc(%)					AUC
k	0.3	0.4	0.5	0.6	0.7	
GNNExplainer	68.83	70.22	67.57	69.21	72.94	0.7069
PGExplainer	68.80	68.62	68.74	70.15	72.14	0.7083
Ours	**70.93**	**72.48**	**73.36**	**74.54**	**75.37**	**0.7253**

Table 2. Interpretability performance in MUTAG when value of k changes from 0.3 to 0.7. acc shows the mean accuracy and AUC shows the area under the $k - acc$ curve.

Table 2 shows the comparison of interpretable performance between our model and SOTA GNN explainers including GNNExplainer and PGExplainer. We use the interpretability accuracy acc as evaluating indicator, which could be formulated as:

$$acc = \frac{N_{\text{correct}}}{min\{N, N_{\text{groundtruth}}\}}, \tag{10}$$

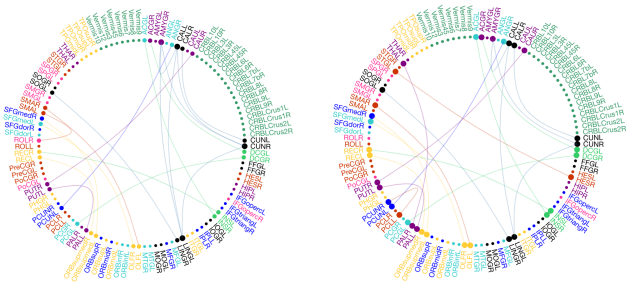
where N_{correct} is the number of correctly selected edges; $N = \lfloor k \cdot |G| \rfloor$ is the edges number in the subgraph controlled by hyperparameter $k \in \{0.1, 0.2, ..., 0.9\}$, $\lfloor \cdot \rfloor$ denotes round down operation; $N_{\text{groundtruth}}$ is the edges number of the groundtruth. It can be seen from the table 2, SG-SIB outperforms the other two *post-hoc* explainable methods when k is between 0.3-0.7. These results indicate that our model could successfully identify the ground truth (carbon ring with NO_2) which determines the label. The results of AUC also confirm this conclusion, which shows that our model can achieve better interpretable performance in the process of k value change.



(a) our method (b) GNNExplainer (c) PGExplainer

Fig. 4. Interpretation result on MUTAG dataset under the setting of $k = 0.5$ ($N = 7$). (a), (b), (c) shows ours, GNNExplainer and PGExplainer interpretable subgraph separately. The interpretable subgraph is marked as green. Our model is the only one that can discover the NO2 group.

To further verify what exactly our model explains, we visualize MUTAG and ABIDE I interpretability result of three models shown in the Fig. 4 and 5 separately. Fig. 4 compares our model with explainable subgraphs between GNNExplainer and PGExplainer on a mutagenic molecule (label 0). Our model could successfully identify the NO_2 group, but



(a) ASD (b) TC

Fig. 5. Interpretation result on ABIDE I dataset. (a), (b) denotes the combined results on ASD individuals and TC individuals, respectively. The colors of brain neural systems are described as: Visual Network (VN), SomatoMotor Network (SMN), Dorsal Attention Network (DAN), Ventral Attention Network (VAN), LImbic Network (LIN), Fronto-Parietal Network (FPN), Default Mode Network (DMN), CereBelLum (CBL) and SuBcortial Network (SBN). (Please refer to supplementary material `https://github.com/SJYuCNEL/brain-and-Information-Bottleneck` for zoomed-in plot.)

neither GNNExplainer nor PGExplainer recognize it, which shows that GNNExplainer and PGExplainer could lead to unfaithful prediction. Fig. 5 shows our interpretable subgaph obtained on ABIDE I. As we can see from Fig. 5(a), connections between ventral DMN and left fronto-parietal network in patients with ASD are stronger than those of TC. This experimental findings is consistent with the previous literature of wang et al. [35] , in which evaluated functional connectivity between ventral DMN and other resting-state network, such as left FPN is reported in the patients with ASD. Fig. 5(b) demonstrates that the sparser connections within the subcortical network are observed in the patients with ASD compared with TC. These findings are in line with a previous study [36], where the decreased functional connectivity between left caudate and right thalamus within subcoritical network are related to ASD social score.

5. CONCLUTION

Experiments show that we have successfully extended the model to more general datasets with robustness. our proposed framework can reasonably extract interpretable subgraphs and reached the SOTA of GNNs in the classification performance. As a *built-in* interpretable method, the interpretation performance and classification performance are obviously better than the baseline of the interpretable model. Through the demonstration in the paper, we prove the advanced nature and universality of our framework.

In the future, we will apply our framework to other graph tasks such as node classification, unsupervised or semi-supervised graph clustering.

6. REFERENCES

[1] K. Zheng, S. Yu, B. Li, R. Jenssen, and B. Chen, "Brainib: Interpretable brain network-based psychiatric diagnosis with graph information bottleneck," *arXiv preprint arXiv:2205.03612*, 2022.

[2] X. Liao, D. Zheng, and X. Cao, "Coronavirus pandemic analysis through tripartite graph clustering in online social networks," *Big Data Mining and Analytics*, vol. 4, no. 4, pp. 242–251, 2021.

[3] S. Liu, M. F. Demirel, and Y. Liang, "N-gram graph: Simple unsupervised representation for graphs, with applications to molecules," *NeurIPS*, vol. 32, 2019.

[4] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *T-ITS*, vol. 21, no. 9, pp. 3848–3858, 2019.

[5] C. Rudin, "Please stop explaining black box models for high stakes decisions," *Stat*, vol. 1050, p. 26, 2018.

[6] H. Yuan, H. Yu, S. Gui, and S. Ji, "Explainability in graph neural networks: A taxonomic survey," *TPAMI*, 2022.

[7] Z. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "Gnnexplainer: Generating explanations for graph neural networks," *NeurIPS*, vol. 32, 2019.

[8] D. Luo, W. Cheng, D. Xu, W. Yu, B. Zong, H. Chen, and X. Zhang, "Parameterized explainer for graph neural network," *NeurIPS*, vol. 33, pp. 19 620–19 631, 2020.

[9] 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.

[10] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," *arXiv preprint physics/0004057*, 2000.

[11] T. Wu, H. Ren, P. Li, and J. Leskovec, "Graph information bottleneck," *NeurIPS*, vol. 33, pp. 20 437–20 448, 2020.

[12] J. Yu, T. Xu, Y. Rong, Y. Bian, J. Huang, and R. He, "Recognizing predictive substructures with subgraph information bottleneck," *TPAMI*, 2021.

[13] S. Yu, L. G. S. Giraldo, R. Jenssen, and J. C. Principe, "Multivariate extension of matrix-based rényi's α -order entropy functional," *TPAMI*, 2020.

[14] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, and R. D. Hjelm, "Mine: mutual information neural estimation," *ICML*, 2018.

[15] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[16] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks," *CVPR*, 2018.

[17] P. Veliković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," *ICLR*, 2017.

[18] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *NeurIPS*, vol. 30, 2017.

[19] O. Vinyals, S. Bengio, and M. Kudlur, "Order matters: Sequence to sequence for sets," *ICLR*, 2015.

[20] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in *AAAI*, vol. 32, no. 1, 2018.

[21] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," *NeurIPS*, vol. 31, 2018.

[22] Y. Ma, S. Wang, C. C. Aggarwal, and J. Tang, "Graph convolutional networks with eigenpooling," in *KDD '19*, 2019, pp. 723–731.

[23] J. Lee, I. Lee, and J. Kang, "Self-attention graph pooling," in *ICML*. PMLR, 2019, pp. 3734–3743.

[24] S. Miao, M. Liu, and P. Li, "Interpretable and generalizable graph learning via stochastic attention mechanism," in *ICML*. PMLR, 2022, pp. 15 524–15 543.

[25] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *ICLR*, 2016.

[26] Z. Wang and S. Ji, "Second-order pooling for graph neural networks," *TPAMI*, 2020.

[27] J. Yu, J. Cao, and R. He, "Improving subgraph recognition with variational graph information bottleneck," in *CVPR*, 2022, pp. 19 396–19 405.

[28] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, "What makes for good views for contrastive learning?" *NeurIPS*, vol. 33, pp. 6827–6839, 2020.

[29] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *ICLR*, 2016.

[30] Z. Zhang, Q. Liu, H. Wang, C. Lu, and C. Lee, "Protgnn: Towards self-explaining graph neural networks," in *AAAI*, vol. 36, no. 8, 2022, pp. 9127–9135.

[31] A. K. Debnath, L. de Compadre Rl, G. Debnath, A. J. Shusterman, and C. Hansch, "Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity." *Journal of Medicinal Chemistry*, 1991.

[32] K. M. Borgwardt, C. S. Ong, S. Schönauer, S. Vishwanathan, A. J. Smola, and H.-P. Kriegel, "Protein function prediction via graph kernels," *Bioinformatics*, vol. 21, no. suppl_1, pp. i47–i56, 2005.

[33] N. Wale, I. A. Watson, and G. Karypis, "Comparison of descriptor spaces for chemical compound retrieval and classification," *Knowledge and Information Systems*, vol. 14, no. 3, pp. 347–375, 2008.

[34] C. Cai and Y. Wang, "A simple yet effective baseline for non-attributed graph classification," *ICLR Workshop: Representation Learning on Graphs and Manifolds*, 2019.

[35] K. Wang, K. Li, and X. Niu, "Altered functional connectivity in a triple-network model in autism with co-occurring attention deficit hyperactivity disorder," *Frontiers in psychiatry*, vol. 12, 2021.

[36] C. Zuo, D. Wang, F. Tao, and Y. Wang, "Changes in the development of subcortical structures in autism spectrum disorder," *Neuroreport*, vol. 30, no. 16, pp. 1062–1067, 2019.