GRAPH INFORMATION BOTTLENECK FOR SUBGRAPH RECOGNITION

Anonymous authors
Paper under double-blind review

ABSTRACT

Given the input graph and its label/property, several key problems of graph learning, such as finding interpretable subgraphs, graph denoising and graph compression, can be attributed to the fundamental problem of recognizing a subgraph of the original one. This subgraph shall be as informative as possible, yet contains less redundant and noisy structure. This problem setting is closely related to the well-known information bottleneck (IB) principle, which, however, has less been studied for the irregular graph data and graph neural networks (GNNs). In this paper, we propose a framework of Graph Information Bottleneck (GIB) for the subgraph recognition problem in deep graph learning. Under this framework, one can recognize the maximally informative yet compressive subgraph, named *IB-subgraph*. However, the GIB objective is notoriously hard to optimize, mostly due to the intractability of the mutual information of irregular graph data and the unstable optimization process. In order to tackle these challenges, we propose: i) a GIB objective based-on a mutual information estimator for the irregular graph data; ii) a bi-level optimization scheme to maximize the GIB objective; iii) a connectivity loss to stabilize the optimization process. We evaluate the properties of the IB-subgraph in three application scenarios: improvement of graph classification, graph interpretation and graph denoising. Extensive experiments demonstrate that the information-theoretic IB-subgraph enjoys superior graph properties.

1 INTRODUCTION

Classifying the underlying labels or properties of graphs is a fundamental problem in deep graph learning with applications across many fields, such as biochemistry and social network analysis. However, real world graphs are likely to contain redundant even noisy information (Franceschi et al., 2019; Yu et al., 2019), which poses a huge negative impact for graph classification. This triggers an interesting problem of recognizing an informative yet compressed subgraph from the original graph. For example, in drug discovery, when viewing molecules as graphs with atoms as nodes and chemical bonds as edges, biochemists are interested in identifying the subgraphs that mostly represent certain properties of the molecules, namely the functional groups (Jin et al., 2020b; Gilmer et al., 2017). In graph representation learning, the predictive subgraph highlights the vital substructure for graph classification, and provides an alternative way for yielding graph representation besides mean/sum aggregation (Kipf & Welling, 2017; Veličković et al., 2017; Xu et al., 2019) and pooling aggregation (Ying et al., 2018; Lee et al., 2019; Bianchi et al., 2020). In graph attack and defense, it is vital to purify a perturbed graph and mine the robust structures for classification (Jin et al., 2020a).

Recently, the mechanism of self-attentive aggregation (Li et al., 2019) somehow discovers a vital substructure at node level with a well-selected threshold. However, this method only identifies isolated important nodes but ignores the topological information at subgraph level. Consequently, it leads to a novel challenge as subgraph recognition: *How can we recognize a compressed subgraph with minimum information loss in terms of predicting the graph labels/properties?*

Recalling the above challenge, there is a similar problem setting in information theory called information bottleneck (IB) principle (Tishby et al., 1999), which aims to juice out a compressed data from the original data that keeps most predictive information of labels or properties. Enhanced with deep learning, IB can learn informative representation from regular data in the fields of computer vision (Peng et al., 2019; Alemi et al., 2017; Luo et al., 2019), reinforcement learning (Goyal et al.,

2019; Igl et al., 2019) and natural language precessing (Wang et al., 2020). However, current IB methods, like VIB (Alemi et al., 2017), is still incapable for irregular graph data. It is still challenging for IB to compress irregular graph data, like a subgraph from an original graph, with a minimum information loss.

Hence, we advance the IB principle for irregular graph data to resolve the proposed subgraph recognition problem, which leads to a novel principle, Graph Information Bottleneck (GIB). Different from prior researches in IB that aims to learn an optimal representation of the input data in the hidden space, GIB directly reveals the vital substructure in the subgraph level. We first i) leverage the mutual information estimator from Deep Variational Information Bottleneck (VIB) (Alemi et al., 2017) for irregular graph data as the GIB objective. However, VIB is intractable to compute the mutual information without knowing the distribution forms, especially on graph data. To tackle this issue, ii) we adopt a bi-level optimization scheme to maximize the GIB objective. Meanwhile, the continuous relaxation that we adopt to approach the discrete selection of subgraph will lead to unstable optimization process. To further stabilize the training process and encourage a compact subgraph, iii) we propose a novel connectivity loss to assist GIB to effectively discover the maximally informative but compressed subgraph, which is defined as *IB-subgraph*. By optimizing the above GIB objective and connectivity loss, one can recognize the IB-subgraph without any explicit subgraph annotation. On the other hand, iv) GIB is model-agnostic and can be easily plugged into various Graph Neural Networks (GNNs).

We evaluate the properties of the IB-subgraph in three application scenarios: improvement of graph classification, graph interpretation, and graph denoising. Extensive experiments on both synthetic and real world datasets demonstrate that the information-theoretic IB-subgraph enjoys superior graph properties compared to the subgraphs found by SOTA baselines.

2 Related Work

Graph Classification. In recent literature, there is a surge of interest in adopting graph neural networks (GNN) in graph classification. The core idea is to aggregate all the node information for graph representation. A typical implementation is the mean/sum aggregation (Kipf & Welling, 2017; Xu et al., 2019), which is to average or sum up the node embeddings. An alternative way is to leverage the hierarchical structure of graphs, which leads to the pooling aggregation (Ying et al., 2018; Zhang et al., 2018; Lee et al., 2019; Bianchi et al., 2020). When tackling with the redundant and noisy graphs, these approaches will likely to result in sub-optimal graph representation.

Information Bottleneck. Information bottleneck (IB), originally proposed for signal processing, attempts to find a short code of the input signal but preserve maximum information of the code (Tishby et al., 1999). (Alemi et al., 2017) firstly bridges the gap between IB and the deep learning, and proposed variational information bottleneck (VIB). Nowadays, IB and VIB have been wildly employed in computer vision (Peng et al., 2019; Luo et al., 2019), reinforcement learning (Goyal et al., 2019; Igl et al., 2019), natural language processing (Wang et al., 2020) and speech and acoustics (Qian et al., 2020) due to the capability of learning compact and meaningful representations. However, IB is less researched on irregular graphs due to the intractability of mutual information.

Subgraph Discovery. Traditional subgraph discovery includes dense subgraph discovery and frequent subgraph mining. Dense subgraph discovery aims to find the subgraph with the highest density (e.g. the number of edges over the number of nodes (Fang et al., 2019; Gionis & Tsourakakis, 2015)). Frequent subgraph mining is to look for the most common substructure among graphs (Yan & Yan, 2002; Ketkar et al., 2005; Zaki, 2005). Recently, researchers discover the vital substructure at node level via the attention mechanism (Veličković et al., 2017; Lee et al., 2019; Knyazev et al., 2019). Ying et al. (2019) further identifies the important computational graph for node classification. Alsentzer et al. (2020) discovers subgraph representations with specific topology given subgraph-level annotation.

3 Notations and Preliminaries

Let $\{(\mathcal{G}_1, Y_1), \ldots, (\mathcal{G}_N, Y_N)\}$ be a set of N graphs with their real value properties or categories, where $\mathcal{G}_n$ refers to the n-th graph and Y_n refers to the corresponding properties or labels. We denote

by $\mathcal{G}_n = (\mathbb{V}, \mathbb{E}, \boldsymbol{A}, \boldsymbol{X})$ the n-th graph of size $\boldsymbol{M}_n$ with node set $\mathbb{V} = \{V_i | i = 1, \ldots, \boldsymbol{M}_n\}$, edge set $\mathbb{E} = \{(V_i, V_j) | i > j; V_i, V_j \text{ is connected}\}$, adjacent matrix $\boldsymbol{A} \in \{0,1\}^{\boldsymbol{M}_n \times \boldsymbol{M}_n}$, and feature matrix $\boldsymbol{X} \in \mathbb{R}^{\boldsymbol{M}_n \times d}$ of $\boldsymbol{V}$ with d dimensions, respectively. Denote the neighborhood of V_i as $\mathcal{N}(V_i) = \{V_j | (V_i, V_j) \in \mathbb{E}\}$. We use $\mathcal{G}_{sub}$ as a specific subgraph and $\overline{\mathcal{G}}_{sub}$ as the complementary structure of $\mathcal{G}_{sub}$ in $\mathcal{G}$. Let $f : \mathbb{G} \to \mathbb{R}/[0, 1, \cdots, n]$ be the mapping from graphs to the real value property or category, $\boldsymbol{Y}$, $\mathbb{G}$ is the domain of the input graphs. $I(\boldsymbol{X}, \boldsymbol{Y})$ refers to the Shannon mutual information of two random variables.

3.1 Graph convolutional network

Graph convolutional network (GCN) is widely adopted to graph classification. Given a graph $\mathcal{G} = (\mathbb{V}, \mathbb{E})$ with node feature $\boldsymbol{X}$ and adjacent matrix $\boldsymbol{A}$, GCN outputs the node embeddings $\boldsymbol{X}^{'}$ from the following process:

$$\boldsymbol{X}^{'} = \mathrm{GCN}(\boldsymbol{A}, \boldsymbol{X}; \boldsymbol{W}) = \mathrm{ReLU}(\boldsymbol{D}^{-\frac{1}{2}} \hat{\boldsymbol{A}} \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{W}), \tag{1}$$

where $\boldsymbol{D}$ refers to the diagonal matrix with nodes' degrees and $\boldsymbol{W}$ refers to the model parameters.

One can simply sum up the node embeddings to get a fixed length graph embeddings (Xu et al., 2019). Recently, researchers attempt to exploit hierarchical structure of graphs, which leads to various graph pooling methods (Li et al., 2019; Gao & Ji, 2019; Lee et al., 2019; Diehl, 2019; Zhang et al., 2018; Ranjan et al., 2020; Ying et al., 2018). Li et al. (2019) enhances the graph pooling with self-attention mechanism to leverage the importance of different nodes contributing to the results. Finally, the graph embedding is obtained by multiplying the node embeddings with the normalized attention scores:

$$\boldsymbol{E} = \mathrm{Att}(\boldsymbol{X}^{'}) = \mathrm{softmax}(\Phi_2 \mathrm{tanh}(\Phi_1 \boldsymbol{X}^{'T}))\boldsymbol{X}^{'}, \tag{2}$$

where Φ_1 and Φ_2 refers to the model parameters of self-attention.

3.2 Optimizing Information bottleneck objective

Given the input signal X and the label Y, the objective of IB is maximized to find the the internal code Z: $\max_Z I(Z, Y) - \beta I(X, Z)$, where β refers to a hyper-parameter trading off informativeness and compression. Optimizing this objective will lead to a compact but informative Z. Alemi et al. (2017) optimize a tractable lower bound of the IB objective:

$$\mathcal{L}_{VIB} = \frac{1}{N} \sum\nolimits_{i=1}^{N} \int p(z|x_i) \log q_\phi(y_i|z) dz - \beta \mathrm{KL}(p(z|x_i) | r(z)), \tag{3}$$

where $q_\phi(y_i|z)$ is the variational approximation to $p_\phi(y_i|z)$ and $r(z)$ is the prior distribution of Z. However, it is hard to estimate the mutual information in high dimensional space when the distribution forms are inaccessible, especially for irregular graph data.

4 Optimizing the Graph Information Bottleneck Objective for Subgraph Recognition

In this section, we will elaborate the proposed method in details. We first formally define the graph information bottleneck and IB-subgraph. Then, we introduce a novel framework for GIB to effectively find the IB-subgraph. We further propose a bi-level optimization scheme and a graph mutual information estimator for GIB optimization. Moreover, we do a continuous relaxation to the generation of subgraph, and propose a novel loss to stabilize the training process.

4.1 Graph information bottleneck

We generalize the information bottleneck principle to learn a informative representation of irregular graphs, which leads to the graph information bottleneck (GIB) principle.

Definition 4.1 (Graph Information Bottleneck)**.** Given a graph $\mathcal{G}$ and its label Y, the GIB seeks for the most informative yet compressed representation Z by optimizing the following objective:

$$\max_Z I(Y, Z) \text{ s.t. } I(\mathcal{G}, Z) \leq I_c. \tag{4}$$

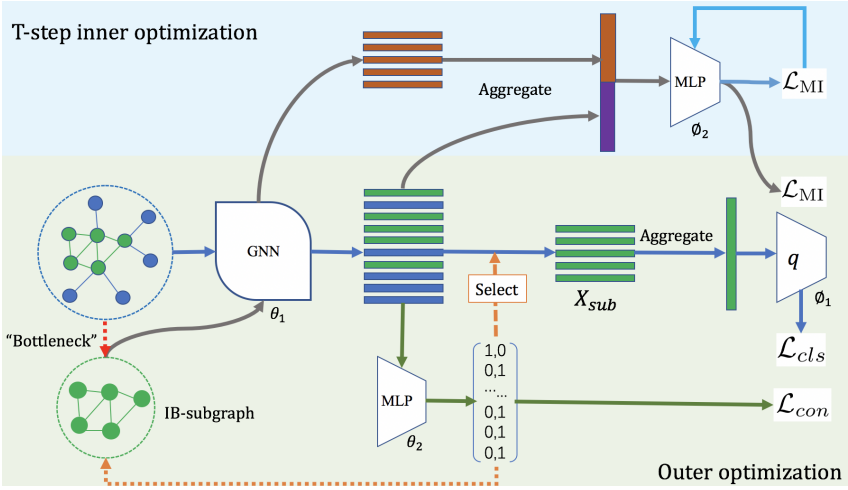

Figure 1: Illustration of the proposed graph information bottleneck (GIB) framework. We employ a bi-level optimization scheme to optimize the GIB objective and thus yielding the IB-subgraph. In the inner optimization phase, we estimate $I(\mathcal{G}, \mathcal{G}_{sub})$ by optimizing the statistics network of the DONSKER-VARADHAN representation (Donsker & Varadhan, 1983). Given a good estimation of $I(\mathcal{G}, \mathcal{G}_{sub})$, in the outer optimization phase, we maximize the GIB objective by optimizing the classification loss $\mathcal{L}_{cls}$ and connectivity loss $\mathcal{L}_{con}$.

where I_c is the information constraint between $\mathcal{G}$ and Z. By introducing a Lagrange multiplier β to Eq. 4, we reach its unconstrained form:

$$\max_{Z} I(Y, Z) - \beta I(\mathcal{G}, Z). \tag{5}$$

Eq. 5 gives a general formulation of GIB. Here, in subgraph recognition, we focus on a subgraph which is compressed with minimum information loss in terms of graph properties.

Definition 4.2 (IB-subgraph)**.** For a graph $\mathcal{G}$, its maximally informative yet compressed subgraph, namely IB-subgraph can be obtained by optimizing the following objective:

$$\max_{\mathcal{G}_{sub} \in \mathbb{G}_{sub}} I(Y, \mathcal{G}_{sub}) - \beta I(\mathcal{G}, \mathcal{G}_{sub}). \tag{6}$$

where $\mathbb{G}_{sub}$ indicates the set of all subgraphs of $\mathcal{G}$.

IB-subgraph enjoys various pleasant properties and can be applied to multiple graph learning tasks such as improvement of graph classification, graph interpretation, and graph denoising. However, the GIB objective in Eq. 6 is notoriously hard to optimize due to the intractability of mutual information and the discrete nature of irregular graph data. We then introduce approaches on how to optimize such objective and derive the IB-subgraph.

4.2 Bi-level optimization for the GIB objective

The GIB objective in Eq. 6 consists of two parts. We examine the first term $I(Y, \mathcal{G}_{sub})$ in Eq. 6, first. This term measures the relevance between $\mathcal{G}_{sub}$ and Y. We expand $I(Y, \mathcal{G}_{sub})$ as:

$$I(Y, \mathcal{G}_{sub}) = \int p(y, \mathcal{G}_{sub}) \log p(y|\mathcal{G}_{sub}) dy \, d\mathcal{G}_{sub} + \mathrm{H}(Y). \tag{7}$$

$H(Y)$ is the entropy of Y and thus can be ignored. In practice, we approximate $p(y, \mathcal{G}_{sub})$ with an empirical distribution $p(y, \mathcal{G}_{sub}) \approx \frac{1}{N} \sum_{i=1}^{N} \delta_{y_i}(y) \delta_{\mathcal{G}_{sub,i}}(\mathcal{G}_{sub})$, where $\mathcal{G}_{sub}$ is the output subgraph and Y is the graph label. By substituting the true posterior $p(y|\mathcal{G}_{sub})$ with a variational approximation $q_{\phi_1}(y|\mathcal{G}_{sub})$, we obtain a tractable lower bound of the first term in Eq. 6:

$$\begin{aligned} I(Y, \mathcal{G}_{sub}) &\geq \int p(y, \mathcal{G}_{sub}) \log q_{\phi_1}(y|\mathcal{G}_{sub}) dy \, d\mathcal{G}_{sub} \\ &\approx \frac{1}{N} \sum_{i=1}^{N} q_{\phi_1}(y_i|\mathcal{G}_{sub_i}) =: -\mathcal{L}_{cls}(q_{\phi_1}(y|\mathcal{G}_{sub}), y_{gt}), \end{aligned} \tag{8}$$

where y_{gt} is the ground truth label of the graph. Eq. 8 indicates that maximizing $I(Y, \mathcal{G}_{sub})$ is achieved by the minimization of the classification loss between Y and $\mathcal{G}_{sub}$ as $\mathcal{L}_{cls}$. Intuitively, minimizing $\mathcal{L}_{cls}$ encourages the subgraph to be predictive of the graph label. In practice, we choose the cross entropy loss for categorical Y and the mean squared loss for continuous Y, respectively. For more details of deriving Eq. 7 and Eq. 8, please refer to Appendix A.1.

Then, we consider the minimization of $I(\mathcal{G}, \mathcal{G}_{sub})$ which is the second term of Eq. 6. Remind that Alemi et al. (2017) introduces a tractable prior distribution $r(Z)$ in Eq. 3, and thus results in a variational upper bound. However, this setting is troublesome as it is hard to find a reasonable prior distribution for $p(\mathcal{G}_{sub})$, which is the distribution of graph substructures instead of latent representation. Thus we go for another route. Directly applying the DONSKER-VARADHAN representation (Donsker & Varadhan, 1983) of the KL-divergence, we have:

$$I(\mathcal{G}, \mathcal{G}_{sub}) = \sup_{f_{\phi_2}:\mathbb{G}\times\mathbb{G}\to\mathbb{R}} \mathbb{E}_{\mathcal{G},\mathcal{G}_{sub}\in p(\mathcal{G},\mathcal{G}_{sub})} f_{\phi_2}(\mathcal{G}, \mathcal{G}_{sub}) - \log \mathbb{E}_{\mathcal{G}\in p(\mathcal{G}),\mathcal{G}_{sub}\in p(\mathcal{G}_{sub})} e^{f_{\phi_2}(\mathcal{G},\mathcal{G}_{sub})}, \tag{9}$$

where f_{ϕ_2} is the statistics network that maps from the graph set to the set of real numbers. In order to approximate $I(\mathcal{G}, \mathcal{G}_{sub})$ using Eq. 9, we design a statistics network based on modern GNN architectures as shown by Figure 1: first we use a GNN to extract embeddings from both $\mathcal{G}$ and $\mathcal{G}_{sub}$ (parameter shared with the subgraph generator, which will be elaborated in Section 4.3), then concatenate $\mathcal{G}$ and $\mathcal{G}_{sub}$ embeddings and feed them into a MLP, which finally produces the real number. In conjunction with the sampling method to approximate $p(\mathcal{G}, \mathcal{G}_{sub})$, $p(\mathcal{G})$ and $p(\mathcal{G}_{sub})$, we reach the following optimization problem to approximate[1] $I(\mathcal{G}, \mathcal{G}_{sub})$:

$$\max_{\phi_2} \quad \mathcal{L}_{\mathrm{MI}}(\phi_2, \mathcal{G}_{sub}) = \frac{1}{N}\sum_{i=1}^{N} f_{\phi_2}(\mathcal{G}_i, \mathcal{G}_{sub,i}) - \log \frac{1}{N}\sum_{i=1,j\neq i}^{N} e^{f_{\phi_2}(\mathcal{G}_i,\mathcal{G}_{sub,j})}. \tag{10}$$

With the approximation to the MI in graph data, we combine Eq. 6 , Eq. 8 and Eq. 10 and formulate the optimization process of GIB as a tractable bi-level optimization problem:

$$\min_{\mathcal{G}_{sub},\phi_1} \quad \mathcal{L}(\mathcal{G}_{sub}, \phi_1, \phi_2^*) = \mathcal{L}_{cls}(q_{\phi_1}(y|\mathcal{G}_{sub}), y_{gt}) + \beta\mathcal{L}_{\mathrm{MI}}(\phi_2^*, \mathcal{G}_{sub}) \tag{11}$$

$$\text{s.t.} \quad \phi_2^* = \arg\max_{\phi_2} \mathcal{L}_{\mathrm{MI}}(\phi_2, \mathcal{G}_{sub}). \tag{12}$$

We first derive a sub-optimal ϕ_2 notated as ϕ_2^* by optimizing Eq. 12 for T steps as inner loops. After the T-step optimization of the inner-loop ends, Eq. 10 is a proxy for MI minimization for the GIB objective as an outer loop. Then, the parameter ϕ_1 and the subgraph $\mathcal{G}_{sub}$ are optimized to yield IB-subgraph. However, in the outer loop, the discrete nature of $\mathcal{G}$ and $\mathcal{G}_{sub}$ hinders applying the gradient-based method to optimize the bi-level objective and find the IB-subgraph.

4.3 THE SUBGRAPH GENERATOR AND CONNECTIVITY LOSS

To alleviate the discreteness in Eq. 11, we propose the continuous relaxation to the subgraph recognition and propose a loss to stabilize the training process.

Subgraph generator: For the input graph $\mathcal{G}$, we generate its IB-subgraph with the node assignment $\boldsymbol{S}$ which indicates the node is in $\mathcal{G}_{sub}$ or $\overline{\mathcal{G}}_{sub}$. Then, we introduce a continuous relaxation to the node assignment with the probability of nodes belonging to the $\mathcal{G}_{sub}$ or $\overline{\mathcal{G}}_{sub}$. For example, the i-th row of $\boldsymbol{S}$ is 2-dimensional vector $[p(V_i \in \mathcal{G}_{sub}|V_i), p(V_i \in \overline{\mathcal{G}}_{sub}|V_i)]$. We first use an l-layer GNN to obtain the node embedding and employ a multi-layer perceptron (MLP) to output $\boldsymbol{S}$:

$$\boldsymbol{X}^l = \mathrm{GNN}(\boldsymbol{A}, \boldsymbol{X}^{l-1}; \theta_1), \quad \boldsymbol{S} \ = \mathrm{MLP}(\boldsymbol{X}^l; \theta_2). \tag{13}$$

$\boldsymbol{S}$ is a $n\times 2$ matrix, where n is the number of nodes. For simplicity, we compile the above modules as the subgraph generator, denoted as $g(;\theta)$ with $\theta := (\theta_1, \theta_2)$. When $\boldsymbol{S}$ is well-learned, the assignment of nodes is supposed to saturate to 0/1. The representation of $\mathcal{G}_{sub}$, which is employed for predicting the graph label, can be obtained by taking the first row of $\boldsymbol{S}^T\boldsymbol{X}^l$.

[1]Notice that the MINE estimator (Belghazi et al., 2018) straightforwardly uses the DONSKER-VARADHAN representation to derive an MI estimator between the regular input data and its vectorized representation/encoding. It cannot be applied to estimate the mutual information between $\mathcal{G}$ and $\mathcal{G}_{sub}$ since both of $\mathcal{G}$ and $\mathcal{G}_{sub}$ are irregular graph data.

Table 1: Classification accuracy in percent. The pooling methods yield pooling aggregation while the backbones yield mean aggregation. The proposed GIB method with backbones yields subgraph embedding by aggregating the nodes in subgraphs.

Method	MUTAG	PROTEINS	IMDB-BINARY	DD
SortPool	**0.844 ± 0.141**	0.747 ± 0.044	0.712 ± 0.047	0.732 ± 0.087
ASAPool	0.743 ± 0.077	0.721 ± 0.043	0.715 ± 0.044	0.717 ± 0.037
DiffPool	0.839 ± 0.097	0.727 ± 0.046	0.709 ± 0.053	0.778 ± 0.030
EdgePool	0.759 ± 0.077	0.723 ± 0.044	0.728 ± 0.044	0.736 ± 0.040
AttPool	0.721 ± 0.086	0.728 ± 0.041	0.722 ± 0.047	0.711 ± 0.055
GCN	0.743±0.110	0.719±0.041	0.707 ± 0.037	0.725 ± 0.046
GraphSAGE	0.743±0.077	0.721 ± 0.042	0.709 ± 0.041	0.729 ± 0.041
GIN	0.825±0.068	0.707 ± 0.056	0.732 ± 0.048	0.730 ± 0.033
GAT	0.738 ± 0.074	0.714 ± 0.040	0.713 ± 0.042	0.695 ± 0.045
GCN+GIB	0.776 ± 0.075	0.748 ± 0.046	0.722 ± 0.039	0.765 ± 0.050
GraphSAGE+GIB	0.760 ± 0.074	0.734 ± 0.043	0.719 ± 0.052	**0.781 ± 0.042**
GIN+GIB	0.839 ± 0.064	**0.749 ± 0.051**	**0.737 ± 0.070**	0.747 ± 0.039
GAT+GIB	0.749 ± 0.097	0.737 ± 0.044	0.729 ± 0.046	0.769 ± 0.040

Connectivity loss: However, poor initialization will cause $p(V_i \in \mathcal{G}_{sub}|V_i)$ and $p(V_i \in \overline{\mathcal{G}}_{sub}|V_i)$ to be close. This will either lead the model to assign all nodes to $\mathcal{G}_{sub}$ / $\overline{\mathcal{G}}_{sub}$, or result that the representations of $\mathcal{G}_{sub}$ contain much information from the redundant nodes. These two scenarios will cause the training process to be unstable. On the other hand, we suppose our model to have an inductive bias to better leverage the topological information while $\boldsymbol{S}$ outputs the subgraph at a node-level. Therefore, we propose the following connectivity loss:

$$\mathcal{L}_{con} = ||\mathrm{Norm}(\boldsymbol{S}^T\boldsymbol{A}\boldsymbol{S}) - \boldsymbol{I}_2||_F, \tag{14}$$

where $\mathrm{Norm}(\cdot)$ is the row-wise normalization, $||\cdot||_F$ is the Frobenius norm, and $\boldsymbol{I}_2$ is a 2×2 identity matrix. $\mathcal{L}_{con}$ not only leads to distinguishable node assignment, but also encourage the subgraph to be compact. Take $(\boldsymbol{S}^T\boldsymbol{A}\boldsymbol{S})_{1:}$ for example, denote a_{11}, a_{12} the element 1,1 and the element 1,2 of $\boldsymbol{S}^T\boldsymbol{A}\boldsymbol{S}$,

$$\begin{aligned} a_{11} &= \sum_{i,j} A_{ij} p(V_i \in \mathcal{G}_{sub}|V_i) p(V_j \in \mathcal{G}_{sub}|V_j), \\ a_{12} &= \sum_{i,j} A_{ij} p(V_i \in \mathcal{G}_{sub}|V_i) p(V_j \in \overline{\mathcal{G}}_{sub}|V_j). \end{aligned} \tag{15}$$

Minimizing $\mathcal{L}_{con}$ results in $\frac{a_{11}}{a_{11}+a_{12}} \to 1$. This occurs if V_i is in $\mathcal{G}_{sub}$, the elements of $\mathcal{N}(V_i)$ have a high probability in $\mathcal{G}_{sub}$. Minimizing $\mathcal{L}_{con}$ also encourages $\frac{a_{12}}{a_{11}+a_{12}} \to 0$. This encourages $p(V_i \in \mathcal{G}_{sub}|V_i) \to 0/1$ and less cuts between $\mathcal{G}_{sub}$ and $\overline{\mathcal{G}}_{sub}$. This also holds for $\overline{\mathcal{G}}_{sub}$ when analyzing a_{21} and a_{22}.

In a word, $\mathcal{L}_{con}$ encourages distinctive $\boldsymbol{S}$ to stabilize the training process and a compact topology in the subgraph. Therefore, the overall loss is:

$$\begin{aligned} \min_{\theta,\phi_1} \quad & \mathcal{L}(\theta, \phi_1, \phi_2^*) = \mathcal{L}_{con}(g(\mathcal{G};\theta)) + \mathcal{L}_{cls}(q_{\phi_1}(g(\mathcal{G};\theta)), y_{gt}) + \beta\mathcal{L}_{\mathrm{MI}}(\phi_2^*, \mathcal{G}_{sub}) \\ \text{s.t.} \quad & \phi_2^* = \arg\max_{\phi_2} \mathcal{L}_{\mathrm{MI}}(\phi_2, \mathcal{G}_{sub}). \end{aligned} \tag{16}$$

We provide the pseudo code in the Appendix to better illustrate how to optimize the above objective.

5 EXPERIMENTS

In this section, we evaluate the proposed GIB method on three scenarios, including improvement of graph classification, graph interpretation and graph denoising.

Table 2: The mean and variance of absolute property bias between the graphs and the corresponding subgraphs. Note that we try several initiations for GCN+GIB w/o $\mathcal{L}_{con}$ and $\mathcal{L}_{MI}$ to get the current results due to the instability of optimization process.

Method	QED	DRD2	HLM-CLint	MLM-CLint
GCN+Att05	0.48± 0.07	0.20± 0.13	0.90± 0.89	0.92± 0.61
GCN+Att07	0.41± 0.07	0.16± 0.11	1.18± 0.60	1.69± 0.88
GCN+GIB w/o $\mathcal{L}_{con}$	0.46± 0.07	0.15± 0.12	0.45± 0.37	1.58± 0.86
GCN+GIB w/o $\mathcal{L}_{MI}$	0.43± 0.15	0.21± 0.13	0.48± 0.34	1.20± 0.97
GCN+GIB	**0.38± 0.12**	**0.06± 0.09**	**0.37± 0.30**	**0.72± 0.55**

5.1 BASELINES AND SETTINGS

Improvement of graph classification: For improvement of graph classification, GIB generates graph representation by aggregating the subgraph information. We plug GIB into various backbones including GCN (Kipf & Welling, 2017), GAT (Veličković et al., 2017), GIN (Xu et al., 2019) and GraphSAGE (Hamilton et al., 2017). We compare the proposed method with the mean/sum aggregation (Kipf & Welling, 2017; Veličković et al., 2017; Hamilton et al., 2017; Xu et al., 2019) and pooling aggregation (Zhang et al., 2018; Ranjan et al., 2020; Ying et al., 2018; Diehl, 2019) in terms of classification accuracy.

Graph interpretation: The goal of graph interpretation is to find the substructure which shares the most similar property to the molecule. If the substructure is disconnected, we evaluate its largest connected part. We compare GIB with the attention mechanism (Li et al., 2019). That is, we attentively aggregate the node information for graph prediction. The interpretable subgraph is generated by choosing the nodes with top 50% and 70% attention scores, namely Att05 and Att07. GIB outputs the interpretation with the IB-subgraph. Then, we evaluate the absolute property bias (the absolute value of the difference between the property of graph and subgraph) between the graph and its interpretation. For fare comparisons, we adopt the same GCN as the backbone for different methods.

Graph denoising: We translate the permuted graph into the line-graph and use GIB and attention to 1) infer the real structure of graph, 2) classify the permuted graph via the inferred structure. We further compare the performance of GCN and DiffPool on the permuted graphs.

5.2 DATASETS

Improvement of graph classification: We evaluate different methods on the datasets of **MUTAG** (Rupp et al., 2012), **PROETINS** (Borgwardt et al., 2005), **IMDB-BINARY** and **DD** (Rossi & Ahmed, 2015) datasets. [2].

Graph interpretation: We construct the datasets for graph interpretation on four molecule properties based on ZINC dataset, which contains 250K molecules. **QED** measures the drug likeness of a molecule, which is bounded within the range $(0, 1.0)$. **DRD2** measures the probability that a molecule is active against dopamine type 2 receptor, which is bounded with $(0, 1.0)$. **HLM-CLint** and **MLM-CLint** are estimate values of in vitro human and mouse liver microsome metabolic stability (base 10 logrithm of mL/min/g). We sample the molecules with QED ≥ 0.85, DRD2 ≥ 0.50, HLM-CLint ≥ 2, MLM-CLint ≥ 2 for each task. We use 85% of these molecules for training, 5% for validating and 10% for testing.

Graph denoising: We generate a synthetic dataset by adding 30% redundant edges for each graph in **MUTAG** dataset. We use 70% of these graphs for training, 5% for validating and 25% for testing.

5.3 RESULTS

Improvement of Graph Classification: In Table 1, we comprehensively evaluate the proposed method and baselines on improvement of graph classification. We train GIB on various backbones

[2]We follow the protocol in https://github.com/rusty1s/pytorch_geometric/tree/master/benchmark/kernel

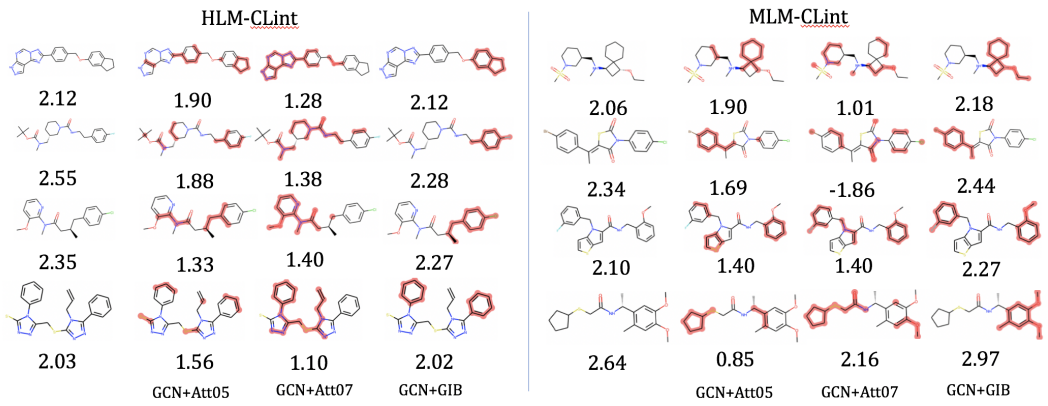

Figure 2: The molecules with their interpretable subgraphs discovered by different methods. These subgraphs exhibit similar chemical properties compared to the molecules on the left.

Table 3: Quantitative results on graph denoising. We report the classification accuracy (Acc), number of real edges over total real edges (Recall) and number of real edges over total edges in subgraphs (Precision) on the test set

Method	GCN	DiffPool	GCN+Att05	GCN+Att07	**GCN+GIB**
Recall	-	-	0.226±0.047	0.324± 0.049	**0.493± 0.035**
Precision	-	-	0.638± 0.141	0.675± 0.104	**0.692 ±0.061**
Acc	0.617	0.658	0.649	0.667	**0.684**

and aggregate the graph representations only from the subgraphs. We compare the performance of our framework with the mean/sum aggregation and pooling aggregation. This shows that GIB improves the graph classification by reducing the redundancies in the graph structure.

Graph interpretation: Table 2 shows the quantitative performance of different methods on the graph interpretation task. GIB is able to generate precise graph interpretation (IB-subgraph), as the substructures found by GIB has the most similar property to the input molecules. Then we derive two variants of our method by deleting $\mathcal{L}_{con}$ and $\mathcal{L}_{MI}$. GIB also outperforms the variants, and thus indicates that every part of our model does contribute to the improvement of performance. In practice, we observe that removing $\mathcal{L}_{con}$ will lead to unstable training process due to the continuous relaxation to the generation of subgraph. In Fig. 2, GIB generates more compact and reasonable interpretation to the property of molecules confirmed by chemical experts. More results are provided in the Appendix. In Table 4, we compare the average number of disconnected substructures per graph since a compact subgraph should preserve more topological information. GIB generates more compact subgraphs to better interpret the graph property.

Table 4: Average number of disconnected substructures per graph selected by different methods

Method	QED	DRD2	HLM	MLM
GCN+Att05	3.38	1.94	3.11	5.16
GCN+Att07	2.04	1.76	2.75	3.00
GCN+GIB	**1.57**	**1.08**	**2.29**	**2.06**

Graph denoising: Table 3 shows the performance of different methods on noisy graph classification. GIB outperforms the baselines on classification accuracy by a large margin due to the superior property of IB-subgraph. Moreover, GIB is able to better reveal the real structure of permuted graphs in terms of precision and recall rate of true edges.

6 Conclusion

In this paper, we have studied a subgraph recognition problem to infer a maximally informative yet compressed subgraph. We define such a subgraph as IB-subgraph and propose the graph information bottleneck (GIB) framework for effectively discovering an IB-subgraph. We derive the GIB objective from a mutual information estimator for irregular graph data, which is optimized by a bi-level learning scheme. A connectivity loss is further used to stabilize the learning process. We evaluate our GIB framework in the improvement of graph classification, graph interpretation and graph denoising. Experimental results verify the superior properties of IB-subgraphs.

REFERENCES

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *The International Conference on Representation Learning*, 2017.

Emily Alsentzer, Samuel G. Finlayson, Michelle M. Li, and Marinka Zitnik. Subgraph neural networks. 2020. URL `http://https://arxiv.org/abs/2006.10538`.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, R. Devon Hjelm, and Aaron C. Courville. Mutual information neural estimation. In *International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 530–539, 2018.

Filippo Maria Bianchi, Daniele Grattarola, and Cesare Alippi. Spectral clustering with graph neural networks for graph pooling. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Karsten M. Borgwardt, Cheng Soon Ong, Stefan Schönauer, S. V. N. Vishwanathan, Alexander J. Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. In *ISMB (Supplement of Bioinformatics)*, pp. 47–56, 2005.

Frederik Diehl. Edge contraction pooling for graph neural networks. *CoRR*, abs/1905.10990, 2019.

M. D. Donsker and S. R. S. Varadhan. Asymptotic evaluation of certain markov process expectations for large time. *Communications on Pure and Applied Mathematics*, 36(2):183–212, 1983.

Yixiang Fang, Kaiqiang Yu, Reynold Cheng, Laks V. S. Lakshmanan, and Xuemin Lin. Efficient algorithms for densest subgraph discovery. *Proceedings of VLDB Endowment*, 12(11):1719–1732, 2019.

Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. Learning discrete structures for graph neural networks. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1972–1982. PMLR, 2019.

Hongyang Gao and Shuiwang Ji. Graph u-nets. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2083–2092. PMLR, 2019.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. *Proceedings of the 34th International Conference on Machine Learning*, 70:1263–1272, 2017.

Aristides Gionis and Charalampos E. Tsourakakis. Dense subgraph discovery: Kdd 2015 tutorial. In *Knowledge Discovery and Data Mining*, pp. 2313–2314. ACM, 2015.

Anirudh Goyal, Riashat Islam, Daniel Strouse, Zafarali Ahmed, Matthew Botvinick, Hugo Larochelle, Yoshua Bengio, and Sergey Levine. Infobot: Transfer and exploration via the information bottleneck. In *The International Conference on Representation Learning*, 2019.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pp. 1024–1034, 2017.

Maximilian Igl, Kamil Ciosek, Yingzhen Li, Sebastian Tschiatschek, Cheng Zhang, Sam Devlin, and Katja Hofmann. Generalization in reinforcement learning with selective noise injection and information bottleneck. In *Advances in neural information processing systems*, 2019.

Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. *CoRR*, abs/2005.10203, 2020a.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Multi-objective molecule generation using interpretable substructures. In *International Conference on Machine Learning*, 2020b.

Nikhil S Ketkar, Lawrence Bruce Holder, and Diane Cook. Subdue: compression-based frequent pattern discovery in graph data. In *Knowledge Discovery and Data Mining*, 2005.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *The International Conference on Representation Learning*, 2017.

Boris Knyazev, Graham W. Taylor, and Mohamed R. Amer. Understanding attention and generalization in graph neural networks. In *NeurIPS*, pp. 4204–4214, 2019.

Junhyun Lee, Inyeop Lee, and Jaewoo Kang. Self-attention graph pooling. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Jia Li, Yu Rong, Hong Cheng, Helen Meng, Wenbing Huang, and Junzhou Huang. Semi-supervised graph classification: A hierarchical graph perspective. In *The World Wide Wed Conference*, 2019.

Yawei Luo, Ping Liu, Tao Guan, Junqing Yu, and Yi Yang. Significance-aware information bottleneck for domain adaptive semantic segmentation. In *ICCV*, pp. 6777–6786. IEEE, 2019.

XueBin Peng, Angjoo Kanazawa, Sam Toyer, Pieter Abbeel, and Sergey Levine. Variational discriminator bottleneck: Improving imitation learning, inverse rl, and gans by constraining information flow. In *The International Conference on Representation Learning*, 2019.

Kaizhi Qian, Yang Zhang, Shiyu Chang, David Cox, and Mark Hasegawa-Johnson. Unsupervised speech decomposition via triple information bottleneck. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Ekagra Ranjan, Soumya Sanyal, and Partha Pratim Talukdar. Asap: Adaptive structure aware pooling for learning hierarchical graph representations. In *AAAI*, 2020.

Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015.

Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, and O. Anatole von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.*, 108(5):058301, January 2012.

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

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representation*, 2017.

Rundong Wang, Xu He, Runsheng Yu, Wei Qiu, Bo An, and Zinovi Rabinovich. Learning efficient multi-agent communication: An information bottleneck approach. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How Powerful are Graph Neural Networks? In *Proceedings of the 7th International Conference on Learning Representations*, ICLR '19, pp. 1–17, 2019.

Xifeng Yan and Jiawei Yan. gspan: graph-based substructure pattern mining. In *IEEE International Conference on Data Mining*, pp. 721–724, 2002.

Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Advances in neural information processing systems*, 2018.

Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. In *Advances in neural information processing systems*, 2019.

Donghan Yu, Ruohong Zhang, Zhengbao Jiang, Yuexin Wu, and Yiming Yang. Graph-revised convolutional network. *CoRR*, abs/1911.07123, 2019.

Mohammed Javeed Zaki. Efficiently mining frequent embedded unordered trees. *Fundamenta Information*, 66(1-2):33–52, 2005.

Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

A APPENDIX

A.1 MORE DETAILS ABOUT EQ. 7 AND EQ. 8

Here we provide more details about how to yield Eq. 7 and Eq. 8.

$$
\begin{aligned}
I(Y, \mathcal{G}_{sub}) &= \int p(y, \mathcal{G}_{sub}) \log p(y|\mathcal{G}_{sub}) dy \, d\mathcal{G}_{sub} - \int p(y, \mathcal{G}_{sub}) \log p(y) dy \, d\mathcal{G}_{sub} \\
&= \int p(y, \mathcal{G}_{sub}) \log p(y|\mathcal{G}_{sub}) dy \, d\mathcal{G}_{sub} + \mathrm{H}(Y) \\
&\geq \int p(y, \mathcal{G}_{sub}) \log q_{\phi_1}(y|\mathcal{G}_{sub}) dy \, d\mathcal{G}_{sub} + \mathrm{KL}(p(y|\mathcal{G}_{sub}) | q_{\phi_1}(y|\mathcal{G}_{sub})) \\
&\geq \int p(y, \mathcal{G}_{sub}) \log q_{\phi_1}(y|\mathcal{G}_{sub}) dy \, d\mathcal{G}_{sub} \\
&\approx \frac{1}{N} \sum_{i=1}^{N} q_{\phi_1}(y_i|\mathcal{G}_{sub_i}) \\
&= -\mathcal{L}_{cls}(q_{\phi_1}(y|\mathcal{G}_{sub}), y_{gt})
\end{aligned}
\tag{17}
$$

A.2 CASE STUDY

To understand the bi-level objective to MI minimization in Eq. 11, we provide a case study in which we optimize the parameters of distribution to reduce the mutual information between two variables. Consider $p(x) = sign(\mathcal{N}(0, 1)), p(y|x) = \mathcal{N}(y; x, \sigma^2)^3$. The distribution of Y is:

$$
\begin{aligned}
p(y) &= \int p(y|x)p(x)dx \\
&= \sum_i p(y|x_i)p(x_i) \\
&= p(y|x = 1)p(x = 1) + p(y|x = -1)p(x = -1) \\
&= 0.5(\mathcal{N}(y; 1, \sigma^2) + \mathcal{N}(y; -1, \sigma^2))
\end{aligned}
\tag{18}
$$

We optimize the parameter σ^2 to reduce the mutual information between X and Y. For each epoch, we sample 20000 data points from each distribution, denoted as $X = \{x_1, x_2, \cdots, x_{20000}\}, Y = \{y_1, y_2, \cdots, y_{20000}\}$. The inner-step is set to be 150. After the inner optimization ends, the model yields a good mutual information approximator and optimize σ^2 to reduce the mutual information by minimizing L_{MI}. We compute the mutual information with the traditional method and compare it with L_{MI}:

$$
\begin{aligned}
I(X, Y) &= \int p(x, y) \log \frac{p(y|x)}{p(y)} dxdy \\
&\approx \frac{1}{20000} \sum_{i=1}^{20000} \log \frac{p(y_i|x_i)}{p(y_i)}
\end{aligned}
\tag{19}
$$

As is shown in Fig .9, the mutual information decreases as $\mathcal{L}_{MI}$ descends. The advantage of such bi-level objective to MI minimization in Eq. 11 is that it only requires samples instead of forms of distribution. Moreover, it needs no tractable prior distribution for variational approximation. The drawback is that it needs additional computation in the inner loop.

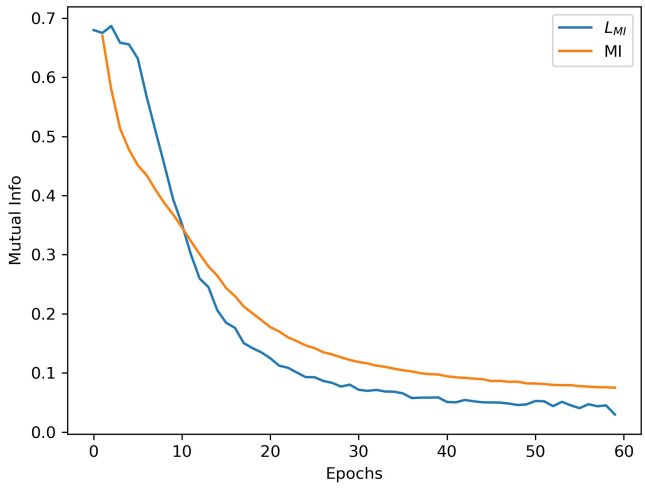

Figure 3: We use the bi-level objective to minimize the mutual information of two distributions. The MI is consistent with the loss as $\mathcal{L}_{MI}$ declines.

Algorithm 1 Optimizing the graph information bottleneck.

Input: Graph $\mathcal{G} = \{\boldsymbol{A}, \boldsymbol{X}\}$, graph label Y, inner-step T, outer-step N.
Output: Subgraph $\mathcal{G}_{sub}$

function GIB($\mathcal{G} = \{\boldsymbol{A}, \boldsymbol{X}\}, Y, T, N$)
  $\theta \leftarrow \theta^0, \quad \phi_1 \leftarrow \phi_1^0$
  for $i = 0 \rightarrow N$ **do**
    $\phi_2 \leftarrow \phi_2^0$
    for $t = 0 \rightarrow T$ **do**
      $\phi_2^{t+1} \leftarrow \phi_2^t + \eta_1 \nabla_{\phi_2^t} \mathcal{L}_{\text{MI}}$
    end for
    $\theta^{i+1} \leftarrow \theta^i - \eta_2 \nabla_{\theta^i} \mathcal{L}(\theta^i, \phi_1^i, \phi_2^T)$
    $\phi_1^{i+1} \leftarrow \phi_1^i - \eta_2 \nabla_{\phi_1^i} \mathcal{L}(\theta^i, \phi_1^i, \phi_2^T)$
  end for
  $\mathcal{G}_{sub} \leftarrow g(\mathcal{G}; \theta^N)$
  return $\mathcal{G}_{sub}$
end function

A.3 ALGORITHM

A.4 MORE RESULTS ON GRAPH INTERPRETATION

In Fig. 4, we show the distribution of absolute bias between the property of graphs and subgraphs. GIB is able to generate such subgraphs with more similar properties to the original graphs.

In Fig. 5, we provide more results of four properties on graph interpretation.

A.5 MORE RESULTS ON NOISY GRAPH CLASSIFICATION

We provide qualitative results on noisy graph classification in Fig. 6.

[3] We use the toy dataset from https://github.com/mzgubic/MINE

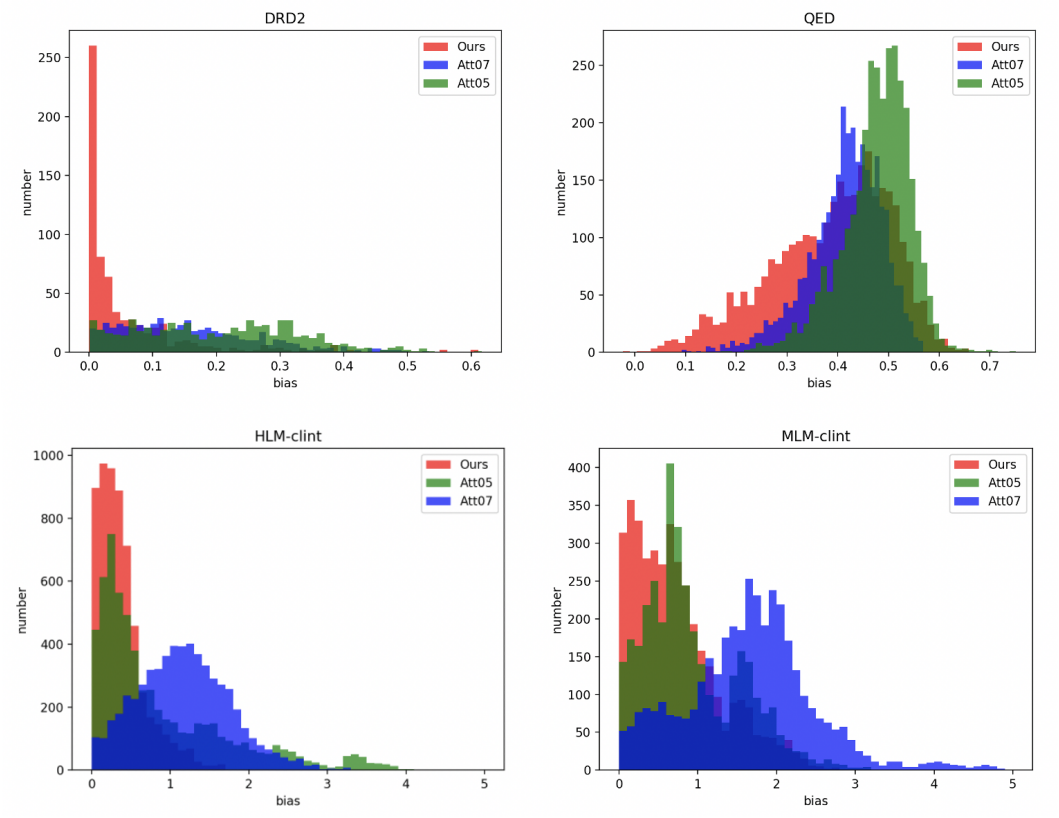

Figure 4: The histgram of absolute bias between the property of graphs and subgraphs.

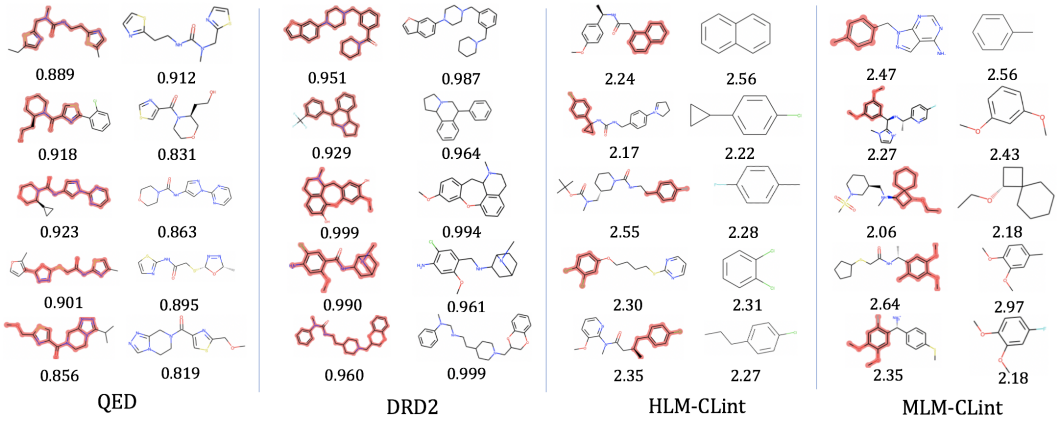

Figure 5: The molecules with its interpretation found by GIB. These subgraphs exhibit similar chemical properties compared to the molecules on the left.

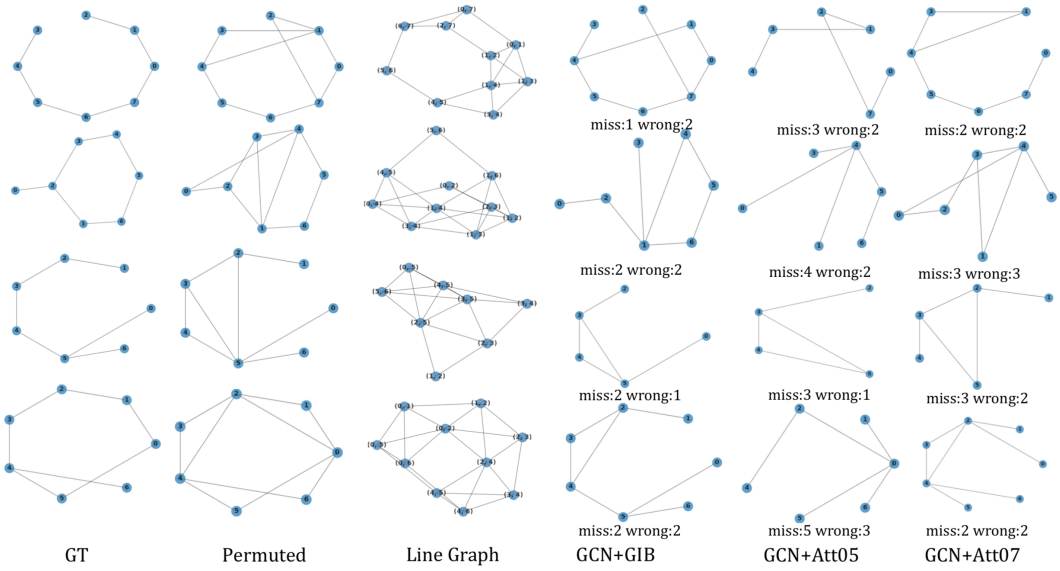

Figure 6: We show the blindly denoising results on permuted graphs. Each method operates on the line-graphs and tries to recover the true topology by removing the redundant edges. Columns 4,5,6 shows results obtained by different methods, where "miss: m, wrong: n" means missing m edges and there are n wrong edges in the output graph. GIB always recognizes more similar structure to the ground truth (not provided in the training process) than other methods.