Recognizing Predictive Substructures With Subgraph Information Bottleneck

Junchi Yu, Tingyang Xu, Yu Rong, Yatao Bian, Junzhou Huang, and Ran He, *Senior Member, IEEE*

Abstract—The emergence of Graph Convolutional Network (GCN) has greatly boosted the progress of graph learning. However, two disturbing factors, noise and redundancy in graph data, and lack of interpretation for prediction results, impede further development of GCN. One solution is to recognize a predictive yet compressed subgraph to get rid of the noise and redundancy and obtain the interpretable part of the graph. This setting of subgraph is similar to the information bottleneck (IB) principle, which is less studied on graph-structured data and GCN. Inspired by the IB principle, we propose a novel subgraph information bottleneck (SIB) framework to recognize such subgraphs, named IB-subgraph. However, the intractability of mutual information and the discrete nature of graph data makes the objective of SIB notoriously hard to optimize. To this end, we introduce a bilevel optimization scheme coupled with a mutual information estimator for irregular graphs. Moreover, we propose a continuous relaxation for subgraph selection with a connectivity loss for stabilization. We further theoretically prove the error bound of our estimation scheme for mutual information and the noise-invariant nature of IB-subgraph. Extensive experiments on graph learning and large-scale point cloud tasks demonstrate the superior property of IB-subgraph.

Index Terms—Graph convolutional network, subgraph information bottleneck, graph classification

1 Introduction

Many real-world data such as social networks, drug molecules, and point clouds can be viewed as graphs. Classifying the labels or analyzing the underlying properties of graph-structured data is the fundamental problem in the deep graph learning area. Recently, there is a surge of interest to employ Graph Convolutional Network (GCN) [1] to cope with such irregular graphs. Many GCN variants [2], [3], [4] have achieved new state-of-the-art performance in various graph-level tasks. Apart from the tremendous progress in deep graph learning, the main concern is that real-world graphs are likely to contain redundant even noisy structure information [5], [6]. This is detrimental to various GCN variants as they are incapable of recognizing the poisoned structure [7], [8], [9]. Moreover, since GCN variants generally leverage all structure information for prediction [2], [10], the interpretation for their prediction results has lagged compared to the significant gain in performance. These two challenges for the literature of graph learning trigger an interesting idea to recognize a subgraph which is predictive in terms of the graph label and discards unnecessary information.

Recognizing the predictive yet compressed subgraph sheds lights on various tasks as it discovers the vital substructure and filters out irrelevant parts. 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 [11], [12]. 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 [1], [10], [13] and pooling aggregation [2], [4], [14]. In graph attack and defense, it is vital to purify a perturbed graph and mine the robust structures for classification [15]. In 3D object detection and segmentation, viewing point clouds as K-NN graphs, researchers seek for substructures which share the same categories [16], [17], [18], [19], [20].

Along with wide applications, the major difficulty is that it is laborious and time-consuming to obtain explicitly subgraph-level annotations for supervised learning. For example, there are 250 thousand molecules in ZINC250K dataset [21]. It requires extra expertise and experience to label every functional group in each molecule. In the absence of explicit supervision, one is blind to recognize the predictive yet compressed subgraph. Recently, the mechanism of self-attentive aggregation [22], [23] employs attention scores to validate the significance of nodes and somehow discovers a vital substructure at node level with a well-selected threshold. However, the discovered substructure is highly

- *Junchi Yu is with the Center for Research on Intelligent Perception and Computing (CRIPAC), National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100864, China. E-mail: yujunchi2019@ia.ac.cn.*
- *Tingyang Xu, Yu Rong, Yatao Bian, and Junzhou Huang are with the Tencent AI LAB, Shenzhen, Guangdong 518000, China. E-mail: tingyangxu@tencent.com, yu.rong@hotmail.com, {yatao.bian, jzhuang75}@gmail.com.*
- *Ran He is with the Center for Research on Intelligent Perception and Computing (CRIPAC), National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100864, China, and also with the Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Beijing 100864, China. E-mail: rhe@nlpr.ia.ac.cn.*

Manuscript received 11 March 2021; revised 24 July 2021; accepted 9 September 2021. Date of publication 14 September 2021; date of current version 5 February 2024.
(Corresponding author: Ran He.)
Recommended for acceptance by L. Liu, T. Hospedales, Y. LeCun, M. Long, J. Luo, W. Ouyang, M. Pietikäinen, and T. Tuytelaars.
Digital Object Identifier no. 10.1109/TPAMI.2021.3112205

influenced by the threshold. Moreover, this method only identifies isolated important nodes but ignores the topological information at the 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 [24], which aims to juice out a compressed code from the original data that keeps most predictive information of labels or properties. Recently, there is a growing tendency to incorporate IB principle with deep learning due to its capability of extracting informative representation from regular data in the fields of computer vision [25], [26], [27], reinforcement learning [28], [29] and natural language processing [30]. However, current IB methods, like VIB [26], are still incapable for irregular graph data. Another work, Graph Information Bottleneck [31], attempts to bind IB with node representation learning. It follows the Gaussian prior assumption in [26], iteratively sample neighbors and learning node representation at each layer of the network, which is still far from subgraph recognition. Hence, it is still challenging for IB to compress irregular graph data, like a subgraph from an original graph, with a minimum information loss.

To this end, we advance the IB principle for irregular graph data to solve the proposed subgraph recognition problem, which leads to a novel principle, Subgraph Information Bottleneck (SIB). SIB directly recognizes a compressed subgraph which is most predictive of a certain graph label without any subgraph annotations. This idea significantly distinguishes from prior researches in IB in two aspects. First of all, SIB directly reveals the vital substructure at the subgraph level instead of learning an optimal representation of the input data in the hidden space. Second, SIB deals with discrete and irregular graph data rather than regular data. We first i) leverage the mutual information estimator from Deep Variational Information Bottleneck (VIB) [26] for irregular graph data as the SIB 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 SIB 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 SIB to effectively discover the maximally informative but compressed subgraph, which is defined as IB-Subgraph. By optimizing the above SIB objective and connectivity loss, one can recognize the IB-Subgraph only with the input graph and its label/property. On the other hand, iv) SIB is model-agnostic and can be easily plugged into various Graph Neural Networks (GNNs).

We evaluate SIB in four application scenarios: improvement of graph classification, graph interpretation, graph denoising, and 3D relevant structure extraction. Extensive experiments on both synthetic and real-world datasets demonstrate that the information-theoretic IB-Subgraph, recognized by the proposed Subgraph Information Bottleneck, enjoys superior graph properties compared to the subgraphs found by SOTA baselines.

This paper is an extension of the conference work [32]. There are three major improvements over the previous one: *1). We theoretically analyze with provable guarantees that the IB-subgraph, found by SIB framework, is noise-invariant.* In the previous version, we empirically analyze that the significant performance gain in various tasks is because SIB is able to preserve the informative substructure and discard noise and redundancy. In this paper, we further indicate that optimizing SIB objective is equivalent to minimize the upper bound of the mutual information of the IB-subgraph and noise, and thus leads to the noise-invariance nature of IB-subgraph. *2). We provide more theoretical analysis on our optimization scheme.* We give the error bound when estimating the mutual information following the PAC-Bayes framework. *3). We provide more experiment results on computer vision tasks.* Despite of graph interpretation, graph classification and graph denoising, we further evaluate the proposed method on S3DIS point cloud dataset [33]. Experiment results show the proposed method is compatible to the large GNN models and efficiently extracts label-relevant structures on large-scale graph datasets.

2 Related Work

Subgraph Discovery. Traditional subgraph discovery mainly 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 [34], [35]). Frequent subgraph mining is to look for the most common substructure among graphs [36], [37], [38]. Recently, it is popular to select a neighborhood subgraph of a central node to do message passing in node representation learning. DropEdge [39] relieves the over-smoothing phenomenon in deep GCNs by randomly dropping a portion of edges in graph data. Similar to DropEdge, DropNode [40], [41], [42] principle is also widely adopted in node representation learning. FastGCN [40] and ASGCN [42] accelerate GCN training via node sampling. GraphSAGE [41] leverages neighborhood sampling for inductive node representation learning. NeuralSparse [43] select Top-K (K is a hyper-parameter) task-relevant 1-hop neighbors of a central node for robust node classification. Similarly, researchers discover the vital substructure at node level via the attention mechanism [10], [14], [23]. [44] further identifies the important computational graph for node classification. [45] discovers subgraph representations with specific topology given subgraph-level annotation. However, the above methods are far from subgraph recognition, as they are incapable of discovering a compressed yet predictive subgraph in graph data.

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 [1], [13], 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 [2], [4], [14], [46]. When tackling redundant and noisy graphs, these approaches will likely to result in the sub-optimal graph representation. Recently, InfoGraph [47] maximize the

mutual information between graph representations and multi-level local representations to obtain more informative global representations.

Information Bottleneck. Information bottleneck (IB), originally proposed for signal processing, attempts to find a short code of the input signal but preserves maximum information of the code [24]. [26] first bridges the gap between IB and deep learning, and proposed variational information bottleneck (VIB). Nowadays, IB and VIB have been wildly employed in computer vision [25], [27], reinforcement learning [28], [29], natural language processing [30], and speech and acoustics [48] 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. A parallel work, named Graph Information Bottleneck, recently incorporates IB with node representation learning. However, it is still far from directly recognizing a compressed but informative subgraph in graph data.

Point Cloud Segmentation. Point cloud segmentation is to identify the category of each element in the set of points, a type of 3D geometric data. Traditional convolutional network can not consume such irregular data. PointNet [49] first proposes a unified framework to process the point cloud data. Recent breakthroughs in graph learning innovate researchers to view the point cloud as a K-NN graph, thus naturally leading to GCN-based solutions [16], [17], [18], [19], [20], [50]. Beyond that, as it is expensive to label the point cloud, [51], [52] leverage a tiny portion of labeled point for weakly supervised training. Here, in the subgraph recognition phenomenon, one is required to infer the underlying substructure of a certain category with no point-level attribution, which is a huge challenge for existing methods.

3 Notations and Preliminaries

Let $\{(G_1, Y_1), \ldots, (G_N, Y_N)\}$ be a set of N graphs with their real value properties or categories, where G_n refers to the nth graph and Y_n refers to the corresponding properties or labels. We denote by $G_n = (\mathbb{V}, \mathbb{E}, A, X)$ the nth graph of size M_n with node set $\mathbb{V} = \{V_i | i = 1, \ldots, M_n\}$, edge set $\mathbb{E} = \{(V_i, V_j) | i > j; V_i, V_j \text{ is connected}\}$, adjacent matrix $A \in \{0,1\}^{M_n \times M_n}$, and feature matrix $X \in R^{M_n \times d}$ of 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 G_{sub} as a specific subgraph and $\overline{G}_{sub}$ as the complementary structure of G_{sub} in G. Let $f : G \rightarrow R/[0, 1, \ldots, n]$ be the mapping from graphs to the real value property or category, Y. $\mathbb{G}$ is the domain of graphs. $I(X, Y)$ 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 $G = (V, \mathbb{E})$ with node feature X and adjacent matrix A, GCN outputs the node embeddings $X^{'}$ from the following process:

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

where D refers to the diagonal matrix with nodes' degrees and $\hat{A} = A + I$ is the adjacent matrix after adding the self-loop. W refers to the model parameters.

One can simply sum up the node embeddings to get a fixed-length graph embeddings [13]. Recently, researchers attempt to exploit an hierarchical structure of graphs, which leads to various graph pooling methods [2], [3], [14], [22], [46], [53], [54]. [22] 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:

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

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

3.2 Information Bottleneck

Given the input data X and the label Y, the information bottleneck principle aims to discover the latent representation Z which is maximally informative in terms of Y (*Sufficient*) and contains as little information of the input data X as possible (*Minimal*). Formally, the sufficient and minimal representation, denoted as Z_s and Z_m, can be obtained by the following objectives respectively:

$$\begin{aligned} Z_s &= \arg\max_Z I(Z, Y) \\ Z_m &= \arg\min_Z I(Z, X), \end{aligned} \tag{3}$$

where $I(A, B)$ is the mutual information between random variable A and B:

$$I(A, B) = \int_{a \in A} \int_{b \in B} p(a, b) \log \frac{p(a, b)}{p(a)p(b)} dadb. \tag{4}$$

Built upon the above intuition, one can learn the minimally sufficient Z by maximizing the information bottleneck objective:

$$\mathcal{L}_{IB} = I(Z, Y) - \beta I(X, Z), \tag{5}$$

where β refers to a hyper-parameter trading off informativeness and compression. Optimizing this objective will lead to a minimally sufficient Z, which is less prone to over-fitting and less sensitive to noise. However, the IB objective is notoriously difficult to optimize as it is troublesome to compute the mutual information. [26] optimize a tractable lower bound of the IB objective:

$$\begin{aligned} \mathcal{L}_{VIB} = \frac{1}{N} \sum\nolimits_{i=1}^{N} \int p(z|x_i) \log q_\phi(y_i|z) dz \\ - \beta \mathrm{KL}(p(z|x_i) | r(z)), \end{aligned} \tag{6}$$

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. KL is the Kullback-Leibler divergence. 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 Subgraph Information Bottleneck Objective for Subgraph Recognition

In this section, we will elaborate on the proposed method in detail. We first formally define the subgraph information bottleneck and IB-Subgraph. Then, we introduce a novel framework for SIB to effectively find the IB-Subgraph. Moreover, we propose a bi-level optimization scheme and a graph mutual information estimator for SIB optimization. We further elaborate a continuous relaxation strategy with a connectivity loss to stabilize the training.

4.1 Subgraph Information Bottleneck

To begin with, we formally define the *Minimal Subgraph* and the *Sufficient Subgraph*:

Definition 4.1. *G_{sub} is the* Minimal Subgraph *if it contains minimally usable information of G. We denote it as $G^m_{sub} = \arg\min_{G_{sub}} I(G, G_{sub})$.*

Intuitively, the minimal subgraph compresses the information via dropping a portion of the topological structures of the corresponding graph. By the definition of mutual information, we have:

$$\begin{aligned} I(G_{sub}, G) &= H(G_{sub}) - H(G_{sub}|G) \\ &\leq H(G_{sub}) \leq \log |\mathbb{G}_{sub}|, \end{aligned} \tag{7}$$

where $|\cdot|$ denotes the cardinally of a set. Eq. (7) provides a natural bound of $I(G_{sub}, G)$. That is, given a finite set $\mathbb{G}$, one can minimize the number of the elements in the induced subgraph set $\mathbb{G}_{sub}$. One possible way is to discover the most common and frequent substructures among graphs [36], [37], [38]. However, it is troublesome to directly optimize the above bound since it is quite loose and hard to optimize with gradient methods.

Definition 4.2. *G_{sub} is the* Sufficient Subgraph *if it maximizes the mutual information $I(G_{sub}, Y)$, where Y is the label of the original graph G. We denote it as $G^s_{sub} = \arg\max_{G_{sub}} I(Y, G_{sub})$.*

A sufficient subgraph ensures that almost all substructures which are essential to the graph property are preserved. Moreover, when defining a Markov Chain $Y \rightarrow G \rightarrow G_{sub}$, we have $I(Y, G_{sub}) \leq I(G, Y)$ by the data processing inequality. This leads to $I(G^s_{sub}, Y) = I(G, Y)$. As a consequence, we can achieve comparable performance in the downstream tasks when replacing G with G_{sub}. However, there is a trivial solution where $G^s_{sub} = G$. In this case, we are unable to distinguish the essential subgraphs and the noise and redundancy.

To this end, we generalize the information bottleneck principle to recognize minimal sufficient subgraphs in irregular graphs, which leads to the subgraph information bottleneck (SIB) principle.

Definition 4.3 (Subgraph Information Bottleneck). *Given a graph G and its label Y, the Subgraph Information Bottleneck seeks for the most informative yet compressed subgraph G_{sub} by optimizing the following objective:*

$$\max_{G_{sub}} I(Y, G_{sub}) - \beta I(G, G_{sub}). \tag{8}$$

We denote the subgraph G_{sub} induced by Eq. (8) as the *IB-Subgraph*. As is shown in Eq. (8), learning the IB-subgraph needs no subgraph-level annotations. One is supposed to obtain the IB-subgraph only with the input graphs and their labels, thanks to the information-theoretic SIB objective. Statistically, IB-Subgraph is minimal sufficient concerning the original graph. Intuitively, the IB-Subgraphs only preserve label-relevant substructure in original graphs, and thus reduce the effect of noise and redundancy to the downstream tasks. Specifically, let G_n be the noisy substructure which is irrelevant to the graph property Y, we show the IB-Subgraph G_{sub} is invariant to the noise G_n.

Proposition 1 (Noise-invariance). *Suppose the noisy structure G_n is independent of Y, the mutual information $I(G_n, G_{sub})$ is upper bounded by $I(G_{sub}, G) - I(G_{sub}, Y)$:*

$$I(G_n, G_{sub}) \leq I(G, G_{sub}) - I(Y, G_{sub}). \tag{9}$$

Proof. We prove the above proposition following the Markov chain assumption in [56]. Suppose G_n interacts with G_{sub} only through G and G is defined by Y and G_n. We can define the following Markov chain $(Y, G_n) \rightarrow G \rightarrow G_{sub}$. By the Data Processing Inequality, we obtain:

$$\begin{aligned} I(G; G_{sub}) &\geq I(Y, G_n; G_{sub}) \\ &= I(G_n; G_{sub}) + I(Y; G_{sub}|G_n) \\ &= I(G_n; G_{sub}) + H(Y|G_n) - H(Y|G_n; G_{sub}), \end{aligned} \tag{10}$$

since G_n and Y are independent, we obtain $H(Y|G_n) = H(Y)$. Moreover, it is straightforward that $H(Y|G_n; G_{sub}) \leq H(Y|G_{sub})$. Plug the above equality and inequality into Eq. (10), and we derive:

$$\begin{aligned} I(G; G_{sub}) &\geq I(G_n; G_{sub}) + H(Y) - H(Y|G_{sub}) \\ &= I(G_n; G_{sub}) + I(Y; G_{sub}), \end{aligned} \tag{11}$$

thus we obtain $I(G_n, G_{sub}) \leq I(G, G_{sub}) - I(Y, G_{sub})$ and conclude the proof. □

Proposition 1 indicates that optimizing the SIB objective in Eq. (8) is equivalent to minimize the mutual information between the IB-Subgraph and the noisy substructure, leading to the noise-invariance property of the IB-Subgraph. This property is appealing as it provides theoretical guarantees that one can effectively discover the vital substructure which mostly influences the property of the original graph by compressing the information in the IB-Subgraph, even in the absence of explicit subgraph-level annotations. Meanwhile, as the SIB objective also restrict the IB-Subgraph to be predictive, it is sufficient to plug SIB objective into various baseline models to enhance their performances, which shows SIB is model-agnostic.

However, the SIB objective in Eq. (8) 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.

4.2 Bi-Level Optimization for the SIB Objective

The SIB objective in Eq. (8) consists of two parts. We examine the first term $I(Y, G_{sub})$ in Eq. (8). This term measures the relevance between G_{sub} and Y:

$$I(Y, G_{sub}) = \int p(Y, G_{sub}) \log p(Y|G_{sub}) dy Y \; dG_{sub} + \mathrm{H}(Y). \tag{12}$$

$H(Y)$ is the entropy of Y and thus can be ignored. In practice, we approximate $p(Y, G_{sub})$ with an empirical distribution $p(Y, G_{sub}) \approx \frac{1}{N}\sum_{i=1}^{N} \delta_Y(y_i)\delta_{G_{sub}}(G_{sub_i})$, where $\delta()$ is the Dirac function to sample training data. G_{sub_i} and Y_i are the output subgraph and graph label corresponding to i-th training data. By substituting the true posterior $p(Y|G_{sub})$ with a variational approximation $q_{\phi_1}(Y|G_{sub})$, we obtain a tractable lower bound of the first term in Eq. (8):

$$\begin{aligned} I(Y, G_{sub}) &= H(Y|G_{sub}) + H(Y) \\ &= \int p(Y, G_{sub}) \log q_{\phi_1}(Y|G_{sub}) dY \\ &\quad + \mathrm{KL}[p(Y|G_{sub})|q_{\phi_1}(Y|G_{sub}))] \\ &\geq \int p(Y, G_{sub}) \log q_{\phi_1}(Y|G_{sub}) dY \; dG_{sub} \\ &\approx \frac{1}{N}\sum_{i=1}^{N} \log q_{\phi_1}(Y_i|G_{sub_i}) \\ &=: -\mathcal{L}_{cls}(q_{\phi_1}(Y|G_{sub}), Y_{gt}), \end{aligned} \tag{13}$$

where Y_{gt} is the ground truth label of the graph. Eq. (13) indicates that maximizing $I(Y, G_{sub})$ is achieved by the minimization of the classification loss between Y and G_{sub} as $\mathcal{L}_{cls}$. Intuitively, minimizing $\mathcal{L}_{cls}$ encourages the subgraph to be predictive of the graph label, and thus leads to the relevance between G_{sub} and Y. In practice, we choose the cross entropy loss for categorical Y and the mean squared loss for continuous Y, respectively. Then we introduce the PAC-Bayes bound:

Theorem 2. *Suppose G_{sub} and Y take value in $\mathcal{G}$ and $\mathcal{Y}$ respectively. Let $f(Y, G_{sub}) = \log q_{\phi_1}(Y|G_{sub}) \in [-B, B]$, and $\mathcal{F}$ is the family of $f(Y, G_{sub})$. let $(\mathcal{F}, d)$ be a metric space, $d = ||\cdot||_2$. Define $\mathcal{F}_t = \{(f_1, f_2, \ldots, f_n)|\forall f_k \in \mathcal{F}, \exists i \in 1 \ldots, n, d(f_k, f_i) \leq t\}$ is a t-cover of $\mathcal{F}$, and $\mathcal{N}(t, \mathcal{F}, d)$ is the covering number of $\mathcal{F}$. Then, for any $\delta \in (0, 0.5)$, with probability $p \geq 1 - \delta$ we have the following inequality, up to a constant:*

$$\begin{aligned} &|I(Y, G_{sub}) - \frac{1}{N}\sum_{i=1}^{N} \log q_{\phi_1}(Y_i|G_{sub_i})| \\ &\leq B\sqrt{\frac{2\log\frac{1}{\delta}}{N}} + \inf_t 2(t + \sqrt{\frac{\log \mathcal{N}(t, \mathcal{F}, d)}{N}}) \\ &\quad + \mathrm{KL}[p(Y|G_{sub})|q_{\phi_1}(Y|G_{sub}))]. \end{aligned} \tag{14}$$

Theorem 2 indicates that we can reduce the estimating error with large training sets and simple prediction models. Moreover, the KL term decreases when the variational approximation approaches $p(Y|G_{sub})$. This also reduces the estimating error. The proof is in supplementary materials, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2021.3112205. Then, we consider the minimization of $I(G, G_{sub})$ which is the second term of Eq. (8). Remind that [26] introduces a tractable prior distribution $r(Z)$ in Eq. (6), and thus results in a variational upper bound. However, this setting is troublesome as it is hard to find a reasonable prior distribution for $p(G_{sub})$, which is the distribution of graph substructures instead of latent representation. Thus we go for another route. Directly applying the DONSKER-VARADHAN representation [55] of the KL-divergence, we have:

$$\begin{aligned} I(G, G_{sub}) = &\sup_{f_{\phi_2}:\mathbb{G}\times\mathbb{G}\to\mathbb{R}} \mathbb{E}_{G,G_{sub}\in p(G,G_{sub})} f_{\phi_2}(G, G_{sub}) \\ &- \log \mathbb{E}_{G\in p(G), G_{sub}\in p(G_{sub})} e^{f_{\phi_2}(G, G_{sub})}, \end{aligned} \tag{15}$$

where f_{ϕ_2} is the statistics network that maps from the graph set to the set of real numbers. In order to approximate $I(G, G_{sub})$ using Eq. (15), we design a statistics network based on modern GNN architectures as shown by Fig. 1: first, we use a GNN to extract embeddings from both G and G_{sub} (parameter shared with the subgraph generator, which will be elaborated in Section 4.3), then concatenate G and G_{sub} embeddings and feed them into a Multi-Layer Perceptron (MLP), which finally produces the real number. In conjunction with the sampling method to approximate $p(G, G_{sub})$, $p(G)$ and $p(G_{sub})$, we reach the following optimization problem to approximate[1] $I(G, G_{sub})$:

$$\begin{aligned} \max_{\phi_2} \quad \mathcal{L}_{\mathrm{MI}}(\phi_2, G_{sub}) = &\frac{1}{N}\sum_{i=1}^{N} f_{\phi_2}(G_i, G_{sub_i}) \\ &- \log \frac{1}{N}\sum_{i=1, j\neq i}^{N} e^{f_{\phi_2}(G_i, G_{sub_j})}. \end{aligned} \tag{16}$$

With the approximation to the MI in graph data, we combine Eq. (8), (13), and (16), and formulate the optimization process of SIB as a tractable bi-level optimization problem:

$$\min_{G_{sub}, \phi_1} \quad \mathcal{L}(G_{sub}, \phi_1, \phi_2^*) = \mathcal{L}_{cls} + \beta\mathcal{L}_{\mathrm{MI}} \tag{17}$$

$$\text{s.t.} \quad \phi_2^* = \arg\max_{\phi_2} \mathcal{L}_{\mathrm{MI}}. \tag{18}$$

We first derive a sub-optimal ϕ_2 notated as ϕ_2^* by optimizing Eq. (18) for T steps as inner loops. After the T-step optimization of the inner-loop ends, Eq. (16) is a proxy for MI minimization for the SIB objective as an outer loop. Then, the parameter ϕ_1 and the subgraph G_{sub} are optimized to yield IB-Subgraph. However, in the outer loop, the discrete nature of G and G_{sub} hinders applying the gradient-based

1. Notice that the MINE estimator [57] 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 G and G_{sub} since both G and G_{sub} are irregular graph data.

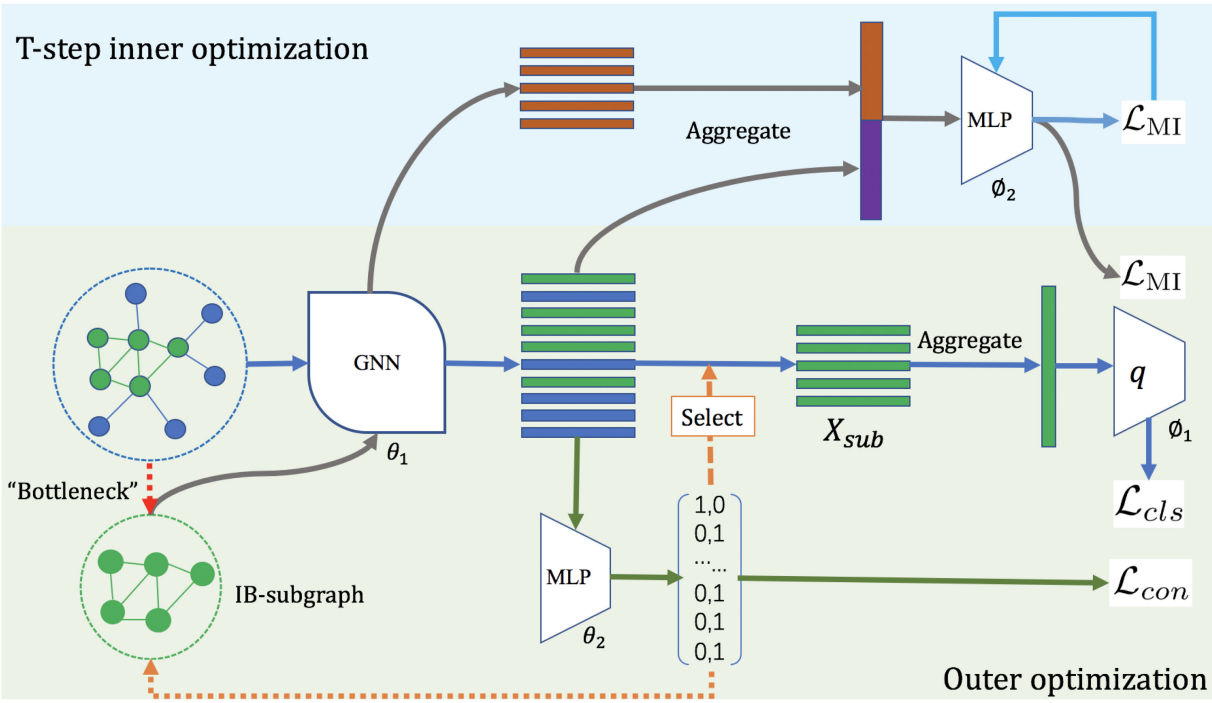


Fig. 1. Illustration of the proposed subgraph information bottleneck (SIB) framework. We employ a bi-level optimization scheme to optimize the SIB objective and thus yielding the IB-Subgraph. In the inner optimization phase, we estimate $I(G, G_{sub})$ by optimizing the statistics network of the DONSKER-VARADHAN representation [55]. Given a good estimation of $I(G, G_{sub})$, in the outer optimization phase, we maximize the SIB objective by optimizing the mutual information, the classification loss $\mathcal{L}_{cls}$ and connectivity loss $\mathcal{L}_{con}$.

method to optimize the bi-level objective and find the IB-Subgraph.

Algorithm 1. Optimizing the Subgraph Information Bottleneck

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

function SIB($G = \{A, X\}, Y, T, N$)
  $\theta \leftarrow \theta^0, \quad \phi_1 \leftarrow \phi_1^0$
  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
  $G_{sub} \leftarrow g(G; \theta^N)$
  return G_{sub}
end Function

4.3 The Subgraph Generator

To alleviate the discreteness in Eq. (17), we propose the continuous relaxation for the subgraph recognition problem and propose a loss to stabilize the training process.

Subgraph Generator. The subgraph generator shown in Fig. 2 generates the IB-Subgraph by learning node assignments. Specifically, it consists of a Graph Neural Network (GNN) and a Multi-layer Perceptron (MLP). The GNN extracts the embedding of each node. The MLP takes the node embeddings as input and outputs the node assignment S which indicates whether the node is in G_{sub} or $\overline{G}_{sub}$. Then, we introduce a continuous relaxation to the node assignment with the probability of nodes belonging to the G_{sub} or $\overline{G}_{sub}$. For example, the ith row of S is a 2-dimensional vector $[p(V_i \in G_{sub}|V_i), p(V_i \in \overline{G}_{sub}|V_i)]$. We first use an l-layer GNN to obtain the node embedding and employ a MLP to output S :

$$\begin{aligned} X^l &= \text{GNN}(A, X^{l-1}; \theta_1), \\ S &= \text{Softmax}(\text{MLP}(X^l; \theta_2)). \end{aligned} \tag{19}$$

Here S is an $n \times 2$ matrix with n as the number of nodes. The added row-wise $\text{Softmax}(\cdot)$ to the output of MLP ensures the nodes are either in or out of the subgraph. For simplicity, we compile the above modules as the subgraph

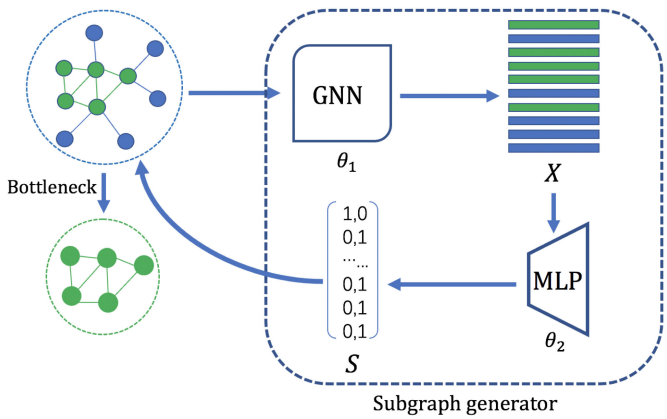


Fig. 2. The subgraph generator. It learns the assignment matrix S which indicates the probability of each node in G_{sub} or $\overline{G}_{sub}$. All the nodes with higher probability in G_{sub} are chosen to form the IB-Subgraph.

generator, denoted as $g(;\theta)$ with $\theta := (\theta_1, \theta_2)$. When S is well-learned, the assignment of nodes is supposed to saturate to 0/1. The representation of G_{sub}, which is employed for predicting the graph label, can be obtained by taking the first row of $S^T X^l$.

Connectivity Loss. When directly generating the IB-subgraph with the above subgraph generator, poor initialization will lead to the assignment $p(V_i \in G_{sub}|V_i)$ and $p(V_i \in \overline{G}_{sub}|V_i)$ of node to be close. Therefore, aggregation of the subgraph embedding by taking the first row of $S^T X^l$ is likely to contain excessive information of $\overline{G}_{sub}$. Moreover, as S outputs the subgraph with a node selection manner, we expect that our model is able to contain an inductive bias to better leverage the topological information. That is, the found subgraph is supposed to be compact. Therefore, we propose the following connectivity loss:

$$\mathcal{L}_{con} = ||\mathrm{Norm}(S^T A S) - I_2||_F, \tag{20}$$

where $\mathrm{Norm}(\cdot)$ is the row-wise normalization, $||\cdot||_F$ is the Frobenius norm, and I_2 is a 2×2 identity matrix. $\mathcal{L}_{con}$ not only leads to distinguishable node assignment but also encourages the subgraph to be compact. Take $(S^T A S)_{1:}$ for example, denote a_{11}, a_{12} the element 1,1 and the element 1,2 of $S^T A S$,

$$\begin{aligned} a_{11} &= \sum_{i,j} A_{ij} p(V_i \in G_{sub}|V_i) p(V_j \in G_{sub}|V_j), a_{12} \\ &= \sum_{i,j} A_{ij} p(V_i \in G_{sub}|V_i) p(V_j \in \overline{G}_{sub}|V_j). \end{aligned} \tag{21}$$

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

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

$$\begin{aligned} \min_{\theta,\phi_1} \quad & \mathcal{L}(\theta, \phi_1, \phi_2^*) = \mathcal{L}_{cls} + \alpha \mathcal{L}_{con} + \beta \mathcal{L}_{MI} \\ \text{s.t.} \quad & \phi_2^* = \arg\max_{\phi_2} \mathcal{L}_{\mathrm{MI}}. \end{aligned} \tag{22}$$

We provide the pseudo code in Algorithm 1 to better illustrate how to optimize the above objective.

5 Experiments

In this section, we evaluate the proposed SIB method on four scenarios, including improvement of graph classification, graph interpretation, graph denoising, and relevant 3D structure extraction.

5.1 Baselines and Settings

Improvement of Graph Classification. We first examine SIB's capability of improving graph classification. Different from prior methods which aggregate all node information for the graph representation, SIB only aggregates the node message in the corresponding IB-subgraph. We plug SIB into various backbones including GCN [1], GAT [10], GIN [13] and GraphSAGE [41]. That is, we first employ the backbones to extract node representations and then yield the graph representation by aggregating the node representation in the IB-Subgraph. We compare the proposed method with the mean/sum aggregation [1], [10], [13], [41] and pooling aggregation [2], [3], [46], [54] in terms of classification accuracy. Moreover, we apply DropEdge [39] to GAT, namely GAT+DropEdge, which randomly drop 30% edges in message-passing at node-level. Similarly, we apply SIB to GAT +DropEdge, resulting in GAT+SIB+DropEdge. This is to examine the flexibility of SIB to the recent proposed dropping-based regularization in node message-passing. For fair comparisons, all the backbones for different methods consist of the same 2-layer GNN with 16 hidden-size.

Graph Interpretation. As the IB-subgraph preserves the most predictive substructure for the graph property, it is interesting to see if we could identify the substructures which best represent some properties of graphs. Therefore, we evaluate SIB on the graph interpretation task. Graph interpretation is to find the substructure with the most similar property to the original graph. In our paper, we focus on interpreting molecules, which is a special type of graph data with atoms and chemical bonds being nodes and edges respectively. The reason is that we can quickly examine the property of molecules and their substructures with toolkits such as RDKIT[2] and iDrug.[3] Bio-chemists can also verify these vital substructures. Moreover, recognizing such vital substructures is the key to new drug discovery [11]. If the substructure is disconnected, we evaluate its largest connected part. We compare SIB with the attention mechanism [22]. That is, we attentively aggregate the node information with the normalized attention scores for graph prediction. The interpretable subgraph is generated by choosing the nodes with top 50% and 70% attention scores, namely Att05 and Att07. SIB outputs the interpretation with the IB-Subgraph, without a manually-selected threshold. 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. Similarly, for fare comparisons, all the backbones for different methods consist of the same 2-layer GNN with 16 hidden-size.

Graph Denoising. We further evaluate the robustness of SIB to the graphs with noisy structures. We translate the permuted graph into the line-graph and use SIB and attention to infer the real structure of the graph. Then, we classify the permuted graph via the inferred structure respectively. We further compare the performance of GCN and DiffPool on the permuted graphs. Similarly, for fare comparisons, all the backbones for different methods consist of the same 2-layer GNN with 16 hidden-size.

Relevant 3D Structure Extraction. Most work on weakly supervised 3D segmentation still requires a tiny portion of labeled points for training [51], [52]. Here we consider a more challenging scenario. That is, to extract relevant 3D structures which belong to the same category without any node-level annotation. This poses a huge challenge for SIB to effectively process large scale graphs. We use the GCN

2. https://www.rdkit.org/
3. https://drug.ai.tencent.com/

TABLE 1
Classification Accuracy

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
GAT + DropEdge	0.743 ± 0.081	0.711 ± 0.043	0.710 ± 0.041	0.717 ± 0.035
GCN+SIB	0.776 ± 0.075	0.748 ± 0.046	0.722 ± 0.039	0.765 ± 0.050
GraphSAGE+SIB	0.760 ± 0.074	0.734 ± 0.043	0.719 ± 0.052	**0.781 ± 0.042**
GIN+SIB	0.839 ± 0.064	**0.749 ± 0.051**	**0.737 ± 0.070**	0.747 ± 0.039
GAT+SIB	0.749 ± 0.097	0.737 ± 0.044	0.729 ± 0.046	0.769 ± 0.040
GAT+SIB+DropEdge	0.754 ± 0.085	0.737 ± 0.037	0.731 ± 0.003	0.776 ± 0.034

The pooling methods yield pooling aggregation while the backbones yield mean aggregation. The proposed SIB method with backbones yields subgraph embedding by aggregating the nodes in subgraphs.

with residual graph connections between layers and GCN with dense graph connections which concatenates representation from previous layers in [50] as our backbones. After each layer, we dynamically updating the constructed K-NN graphs. We refer to these two backbones as ResDyGCN and DenseDyGCN. Since computing the connective loss in large graphs is prone to exceed the GPU memory, we employ Gumbel-Softmax strategy [58] for continuous relaxation in learning assignment matrix. We compare SIB with attention. For fair comparisons, all the backbones for different methods consist of the same 4-layer model with 1024 hidden-size.

5.2 Datasets

Improvement of Graph Classification. We evaluate different methods on the datasets of *MUTAG* [59], *PROTEINS* [60], *IMDB-BINARY* and *DD* [61] datasets.[4] The statistics of the datasets are available in Table 3.[5]

Graph Interpretation. We construct the datasets for graph interpretation on four molecule properties based on ZINC dataset [21]], 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 estimated values of in vitro human and mouse liver microsome metabolic stability (base 10 logarithm 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. The statistics of the datasets are available in Table 4.

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.

Relevant 3D Structure Extraction. We construct the datasets for relevant 3D Structure Extraction based on S3DIS dataset [33]. It contains over 23000 point clouds of 6 indoor areas. Every point cloud consists of 4096 points. Each point is labeled with one of 13 categories, and it has a 9-dimensional feature which indicates the 3D coordinates, normalized 3D coordinates, and RGB color. To enable SIB training, we first choose the point clouds with the percent of main categories greater than 50%, and label the point clouds with the main categories. Then we choose **floor** and **wall** for relevant structure extraction as these two categories contain sufficient training samples. We use Area 1-5 for training and Area 6 for testing, which yield 637 and 2099 training samples and 411 and 1270 testing samples for floor and wall respectively. We train these two categories separately and randomly add the same number of negative samples from the rest 12 categories in training. All point clouds are transformed into K-NN graphs, where K equals 16.

5.3 Results

5.3.1 Improvement of Graph Classification

In Table 1, we comprehensively evaluate the proposed method and baselines on improvement of graph classification. We train SIB on various simple GNN backbones and aggregate the graph representations only from the IB-subgraphs. As is shown, SIB outperforms the backbones by a large margin. This is because SIB can recognize the predictive yet compressed substructures and thus relieves the side-effect of noisy and redundant structures in graph data, which is detrimental to graph classification. Besides, we also compare SIB with many powerful pooling-based methods, which are able to leverage the hierarchical topology information in graphs. SIB also exceeds the pooling methods on most datasets. Notice that SIB still achieves competitive performance with simple GNN backbones on the MUTAG dataset although it littlely under-performs

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

5. The statistics of datasets in improvement of graph classification are collected from http://networkrepository.com

TABLE 2
The Mean and Standard Deviation of Absolute Property Bias Between the Graphs and The Corresponding Subgraphs

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+SIB	**0.38± 0.12**	**0.06± 0.09**	**0.37± 0.30**	**0.72± 0.55**

TABLE 3
Statistics of Datasets in Improvement of Graph Classification

	MUTAG	PROTEINS	IMDB-BINARY	DD
Nodes	97.9K	43.5K	19.8K	334.9K
Edges	202.5K	162.1K	386.1K	1.7M
Density	4.2×10^{-5}	1.7×10^{-4}	2.0×10^{-3}	3.0×10^{-5}
Maximum degree	20	50	540	38
Minimum degree	2	2	4	2
Average degree	4	7	39	10
Number of triangles	2.8K	366K	18.8M	7.1M
Maximum k-core	5	9	117	15
Average number of triangles	0	8	951	21
Maximum number of triangles	12	136	17.8K	160
Average clustering coefficient	0.001965	0.316645	0.831934	0.413379
Fraction of closed triangles	0.003160	0.315106	0.803561	0.410832
Lower bound of Maximum Clique	6	5	18	4

SortPool. Since the pooling methods exploits the graph hierarchically and learn the graph embedding in a coarse-to-fine manner, they are inherently more expressive and more powerful than simple GNNs [3], [4], [22]. Interestingly, the experiment results show that simple GNNs with SIB also achieve competitive results to the pooling methods, which shows the effectiveness of the proposed method.

Recent work shows that regularization on message-passing can lead to better node embeddings [39], [43]. For example, DropEdge relieves over-smoothing phenomenon in deep GNN by randomly dropping a portion of edges in a graph [39], which can be regarded as a random graph sparcification technique. We wonder if such technique can improve graph classification since we generally aggregate node embedding for graph representation. Therefore, we first plug DropEdge into GAT and obtain a variant model GAT+DropEdge. As shown in Table 1, GAT+DropEdge outperforms GAT on MUTAG and DD dataset, while GAT outperforms GAT+DropEdge on the rest datasets. This shows that such sparcification technique may randomly drop important edges for the graph property, and thus probably lead to degenerated results. However, GAT+SIB outperforms GAT on four datasets by recognizing the predictive yet compressed IB-Subgraph for graph classification. We further combine SIB and GAT+DropEdge. Experiment result show SIB can also improve the classification accuracy of GAT+DropEdge, and the performance of GAT+SIB+DropEdge is on par with GAT+SIB. This show the resilience of SIB to the random sparcification.

TABLE 4
Statistics of Datasets in Graph Interpretation

	QED	DRD2	HLM-CLint	MLM-CLint
Number of graphs	35000	3000	25850	16666
Maximum number of nodes	29	66	37	37
Minimum number of nodes	12	13	9	7
Average number of nodes	21.82	27.43	25.14	22.44
Maximum number of edges	34	74	42	42
Minimum number of edges	12	14	9	7
Average number of edges	23.43	30.24	22.23	24.19
Dimension of node features	9	8	9	9

5.3.2 Graph Interpertation

We further examine the capability of IB-subgraph on interpreting the property of molecules. To this end, we first extract the substructures which mostly affect the chemical property of molecules by different models. Then we compare the property of learned substructures with the input molecules. To ensure the chemical validity, when the substructure is disconnected, we evaluate the property of its largest connected part.[6]

Table 2 shows the quantitative performance of different methods on the graph interpretation task. SIB is able to generate precise graph interpretation (IB-Subgraph), as the substructures found by SIB have the most similar property to the input molecules. Moreover, it is noticed that the interpretation found by attention-based method is highly influenced by the manually selected threshold. For example, GCN+Att07 generates subgraphs with more similar properties compared to GCN+att05. However, GCN+Att05 outperforms GCN+Att07 on HLM-CLint and MLM-CLint properties. Therefore, one needs to carefully choose the threshold for different tasks. In contrast, SIB does require a threshold to select the interpretation thanks to the information theoretic objective. In Fig. 3, SIB generates a more compact and reasonable interpretation of the property of molecules confirmed by chemical experts. More visualization results are provided in Fig. 5.

We further compare the compactness of the found subgraphs of different methods as a more compact subgraph has a higher probability to be a functional group and easier for chemists to interpret the property of a molecule. In Table 6, we compare the average number of disconnected substructures per graph. SIB generates more compact subgraphs to better interpret the graph property. Moreover, compared to the baselines, SIB does not require a hyperparameter to control the sizes of subgraphs, thus being more adaptive to different tasks.

5.3.3 Graph Denoising

Table 5 shows the performance of different methods on noisy graph classification. GCN and DiffPool are vulnerable to structure perturbation since they are unable to distinguish the real structure and noise. However, SIB is able to better reveal the real structure of permuted graphs in terms of precision and recall rate of true edges, even in the absence

6. We obtain QED and DRD2 values of molecules with the toolkit on https://www.rdkit.org/. And we evaluate HLM-CLint and MLM-CLint value of molecules on https://drug.ai.tencent.com/

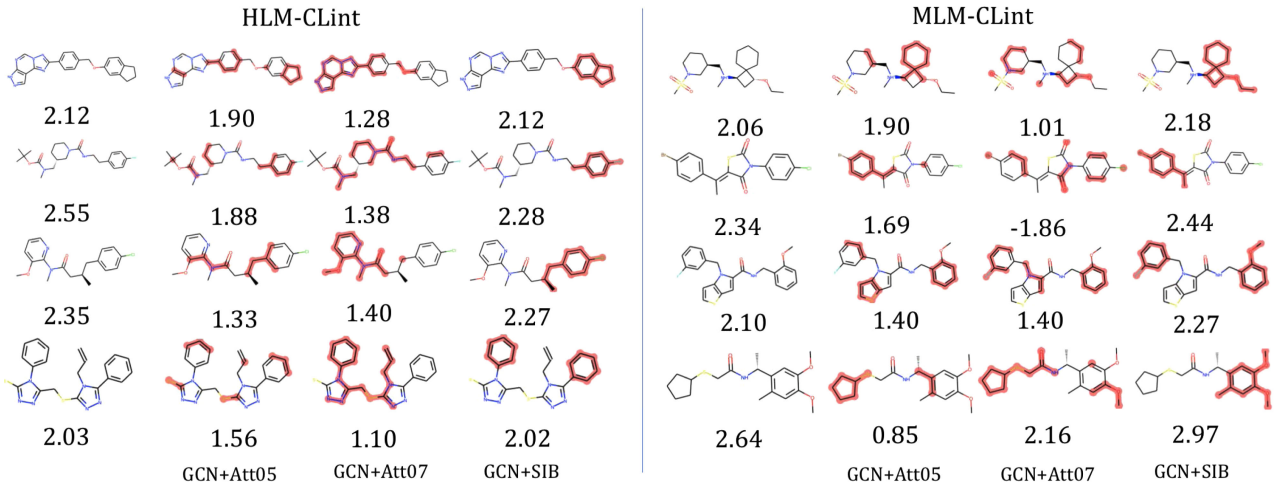


Fig. 3. The molecules with their interpretable subgraphs discovered by different methods. These subgraphs exhibit similar chemical properties compared to the molecules on the left.

of explicit annotations. Therefore, SIB is more robust to perturbations and outperforms the baselines on classification accuracy by a large margin. We provide visualization results in Fig. 6. It is noticed that SIB recognizes more similar structures to the ground truth (not provided in the training process) than other methods.

5.3.4 Relevant 3D Structure Extraction

This task is rather difficult since no point label is provided in the scene point cloud. To the best of our knowledge, few prior methods focus on this extreme scenario. We evaluate different methods in terms of mean intersection over union (mIoU) between the learned substructure and the ground truth[51], [52]. We employ two backbones, namely ResDyGCN and DenseDyGCN, in [50] to extensively compare different methods. As is shown in Table 7, SIB also outperforms the attentive methods since it is capable of effectively recognize relevant structures in large scale point clouds. The results of attentive methods are highly influenced by the threshold. We also notice using ResDyGCN as backbone leads to better results. This is because DenseDyGCN concatenates all previous point representation at each layer, which potentially brings superfluous information. Meanwhile, the residual connection between layers avoids gradient vanishing, and thus leads to more stable training and better performance.

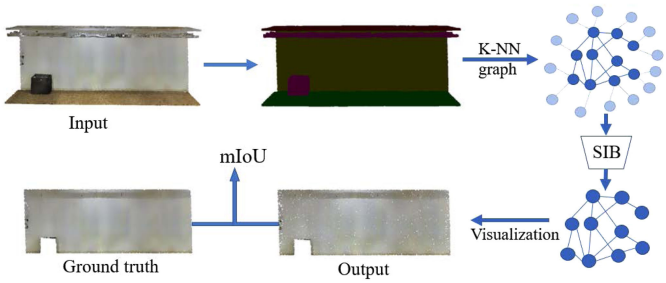


Fig. 4. The pipeline of relevant 3D structure extraction. We label the point cloud with the category of its main component and leverage the color and coordinates as the feature of each point. Then we convert the point cloud into a K-NN graph, where K equals 16. SIB further takes graph data as input and extracts the structure which is most relevant to its main component. We evaluate the result by computing the mIoU of the output and the ground truth (not available in the training procedure) in the testing set. This example is from Area 5 in S3DIS dataset.

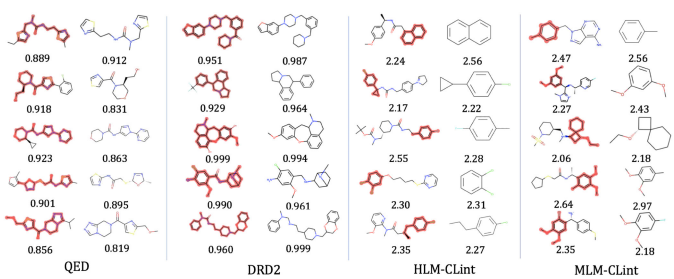


Fig. 5. The molecules with its interpretation found by SIB. These subgraphs exhibit similar chemical properties compared to the molecules on the left.

5.3.5 Ablation Study

To further understand the roles of $\mathcal{L}_{con}$ and $\mathcal{L}_{MI}$, we derive two variants of our method by deleting $\mathcal{L}_{con}$ and $\mathcal{L}_{MI}$,

TABLE 5
Quantitative Results on Graph Denoising

Method	GCN	DiffPool	GCN+Att05	GCN+Att07	**GCN+SIB**
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**

We report the classification accuracy (Acc), the number of real edges over total real edges (Recall), and the number of real edges over total edges in subgraphs (Precision) on the test set.

TABLE 6
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+SIB	**1.57**	**1.08**	**2.29**	**2.06**

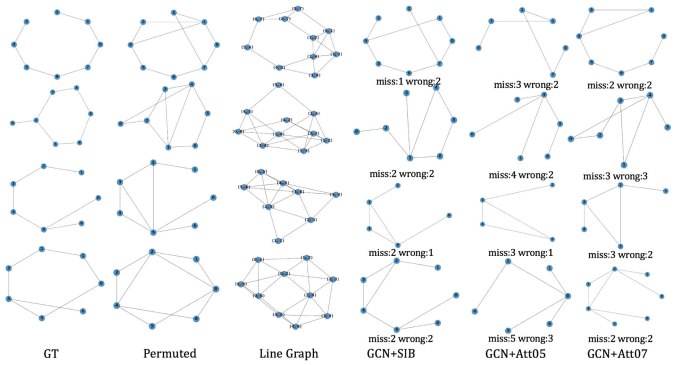


Fig. 6. We show the blindly denoising results on permuted graphs. Each method operates on the line-graphs and tries to recover the true topology by removing the redundant edges. Columns 4,5,6 shows results obtained by different methods, where "miss: m, wrong: n" means missing m edges and there are n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

TABLE 7
We Compare the Capability of Different Methods on Extraction Label-Relevant Substructures in 3D Point Cloud

Backbone	Model	floor		wall	
		mIoU	size	mIoU	size
ResDyGCN	Att05	0.46	2048	0.384	2048
	Att07	0.464	2867	0.465	2867
	SIB	0.519	2819.628	0.492	2032.613
DenseDyGCN	Att05	0.27	2048	0.289	2048
	Att07	0.378	2867	0.358	2867
	SIB	0.397	1978.479	0.419	1784.214

Notice that no point label is available in the training process.

namely SIB w/o $\mathcal{L}_{con}$ and SIB w/o $\mathcal{L}_{MI}$. Note that SIB w/o $\mathcal{L}_{MI}$ is similar to InfoGraph [47] and GNNExplainer [44], as they only consider maximizing MI between latent embedding and global summarization and ignore compression. When adapted to subgraph recognition, it is likely to be $G = G_{sub}$. We evaluate the variants with 2-layer GCN and 16 hidden sizes on graph interpretation. In practice, we find that the training process of SIB w/o $\mathcal{L}_{con}$ is unstable as discussed in Section 4.3. Moreover, we find that SIB w/o $\mathcal{L}_{MI}$ is very likely to output $G_{sub} = G$, as it does not consider compression. Therefore, we try several initiations for SIB w/o $\mathcal{L}_{con}$ and $\mathcal{L}_{MI}$ to get the current results. As shown in Table 8, SIB also outperforms the variants, and thus indicates that every part of our model does contribute to the improvement of performance.

5.3.6 More Discussions

L_{con} is proposed for stabilizing the training process and resulting in compact subgraphs. As it poses regularization for the subgraph generation, we are interested in its potential influence on the sizes of the chosen IB-Subgraphs. Therefore, we show the influence of different hyper-parameters of L_{con} to the sizes of the chosen IB-Subgraphs. We implement the experiments with α varies in $\{1, 3, 5, 10\}$ on QED dataset and compute the mean and deviation of the sizes of IB-Subgraphs (All) and their largest connected parts (Max). As shown in Table 9, we observe that different values of α result in similar sizes of IB-Subgraphs. Therefore, its influence on the size of chosen subgraphs is weak.

As the initialization of our model may potentially influence the final chosen subgraphs, we rerun our model five times on the QED dataset for graph interpretation task. Then, we employ the intersection over union (IoU) to measure the overlap between the subgraphs in 5 different runs and the results reported in Table 2. Similarly, we compute the IoU between the chosen subgraphs and their largest connected parts separately, which refer to IoU_{all} and IoU_{max}. We finally report the mean and standard deviation of IoU_{all}, IoU_{max} on the testing set in Table 10. We notice that different initialization has limited influence on the chosen subgraphs, as all the results of five additional runs have high portions of common nodes with the initial run.

In the graph interpretation task, the hyper-parameter of L_{con}, α, is set to be 5 on four datasets. We show the mean

TABLE 8
Ablation Study on $\mathcal{L}_{con}$ and $\mathcal{L}_{MI}$

Method	QED	DRD2	HLM-CLint	MLM-CLint
SIB w/o $\mathcal{L}_{con}$	0.46± 0.07	0.15± 0.12	0.45± 0.37	1.58± 0.86
SIB w/o $\mathcal{L}_{MI}$	0.43± 0.15	0.21± 0.13	0.48± 0.34	1.20± 0.97
SIB	**0.38± 0.12**	**0.06± 0.09**	**0.37± 0.30**	**0.72± 0.55**

Note that we try several initiations for SIB w/o $\mathcal{L}_{con}$ and $\mathcal{L}_{MI}$ to get the current results due to the instability of optimization process.

TABLE 9
The Influence of the Hyper-Parameter α of L_{con} to the Size of Subgraphs

α	1	3	5	10
All	0.483±0.143	0.496±0.150	0.494±0.147	0.466±0.150
Max	0.387±0.173	0.413±0.169	0.411±0.169	0.391±0.172

TABLE 10
The Overlap Between the Chosen Subgraphs With Different Initialization

Run	1	2	3	4	5
IoU_{all}	0.848±0.163	0.765±0.106	0.784±0.112	0.829±0.166	0.813±0.186
IoU_{max}	0.779±0.330	0.696±0.310	0.742±0.333	0.757±0.335	0.762±0.304

TABLE 11
Size of the Chosen Subgraphs on Four Datasets in Percent

Method	QED	DRD2	HLM-CLint	MLM-CLint
GCN+Att05	43.5±5.4	46.8±3.1	48.1±4.1	45.1±4.1
GCN+Att07	65.8±3.4	66.7±3.0	65.8±5.7	67.6±5.5
GCN+SIB	49.6±15.0	94.8±5.3	47.7±13.7	54.7±17.2

TABLE 12
Size of Largest Connected Parts Used for Graph Interpretation in Percent

Method	QED	DRD2	HLM-CLint	MLM-CLint
GCN+Att05	22.5±9.5	34.7±9.4	23.5±7.2	29.7±7.9
GCN+Att07	43.3±11.8	54.2±13.0	45.0±15.2	41.2±8.4
GCN+SIB	41.3±16.9	92.8±10.4	29.1±10.6	36.9±16.2

TABLE 13
The Performance of SIB With Varied α and β on the QED Dataset

α	β							
	0.01	0.03	0.05	0.1	0.3	0.5	1	5
0.1	0.43+-0.06	0.42+-0.09	0.44+-0.08	0.41+-0.09	0.40+-0.10	0.41+-0.05	0.44+-0.10	0.44+-0.10
0.3	0.42+-0.11	0.42+-0.11	0.42+-0.12	0.39+-0.12	0.41+-0.09	0.40+-0.11	0.45+-0.08	0.42+-0.09
0.5	0.40+-0.08	0.41+-0.12	0.42+-0.09	0.41+-0.10	0.39+-0.11	0.41+-0.07	0.47+-0.09	0.43+-0.11
1	0.41+-0.12	0.38+-0.08	0.39+-0.12	0.38+-0.11	0.38+-0.07	0.40+-0.06	0.43+-0.11	0.42+-0.10
5	0.40+-0.11	0.38+-0.09	0.40+-0.09	0.39+-0.12	0.38+-0.12	0.44+-0.08	0.45+-0.07	0.46+-0.12
10	0.40+-0.09	0.39+-0.11	0.38+-0.11	0.38+-0.12	0.39+-0.09	0.43+-0.07	0.42+-0.01	0.43+-0.09

and standard deviation of the sizes of subgraphs in percent in Tables 11 and 12. Note that the sizes of chosen subgraphs mainly depend on task relevant information. For example, as DRD2 measures the probability of being active against dopamine type 2 receptor, it depends on almost the whole structure of a molecule. In contrast, HLM-CLint measures vitro human microsome metabolic stability, which is greatly influenced by small motifs. As shown in Tables 11 and 12, GCN+SIB can recognize the subgraphs with adaptive sizes on different tasks, leading to better performance. However, in GCN+Att05 and GCN+Att07, the size of subgraphs is explicitly controlled by the hyper-parameter (preserve top 50% or 70 % nodes with the highest attention scores). Therefore, the performances of these methods are limited.

Moreover, we extensively study the sensitivity of SIB to different α and β on the QED dataset, which is shown in Table 13. We observe that a large β will lead to a significant decline in performance. This is because a large penalty on $I(G, G_{sub})$ leads to an over-compressed G_{sub}, which receives insufficient information from G and is less predictive to the label of the original graph. Meanwhile, a small β results in degenerated results, since a small penalty on $I(G, G_{sub})$ makes G_{sub} receive superfluous information from G. We also observe that a large α leads to better results because it leads to a more compact subgraph and distinctive assignment of the assignment matrix S. However, a large α also encourages our model to overemphasize the topological compactness and to ignore the compression and prediction properties of the subgraph. This may lead to an unstable training process. In practice, we choose proper α and β to balance each loss term to achieve the best performance.

6 Conclusion

In this paper, we have studied a subgraph recognition problem to extract a predictive yet compressed subgraph, termed the IB-subgraph. To effectively recognize the IB-subgraph in a weakly supervised manner, we propose a novel subgraph information bottleneck (SIB) framework. Unlike the prior work in information bottleneck, SIB directly recognizes the predictive IB-subgraph for the graph label and operates on the irregular graph-structured data. We further propose a bi-level scheme to efficiently optimize SIB with a mutual information estimator for irregular graph data. We introduce a continuous relaxation to enable training SIB with the gradient-based optimizer and a connectivity loss to stabilize the training process. We evaluate the model-agnostic SIB framework on both graph learning and computer vision scenarios, including the improvement of graph classification, graph interpretation, graph denoising, and relevant 3D structure extraction. Experimental results verify the superior properties of IB-subgraphs.

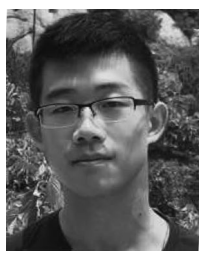
Acknowledgments

The authors would like to thank the associate editor and the reviewers for their valuable comments and advice. This work was supported in part by the Beijing Natural Science Foundation under Grant JQ18017, in part by the National Natural Science Foundation of China under Grants U20A 20223 and 61721004, and in part by the Youth Innovation Promotion Association CAS under Grant Y201929.

References

[1] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representation*, 2017, pp. 1–14.
[2] R. Ying, J. You, C. Morris, X. Ren, W. L. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 4805–4815.
[3] E. Ranjan, S. Sanyal, and P. P. Talukdar, "ASAP: Adaptive structure aware pooling for learning hierarchical graph representations," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 5470–5477.
[4] F. M. Bianchi, D. Grattarola, and C. Alippi, "Spectral clustering with graph neural networks for graph pooling," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 874–883.
[5] L. Franceschi, M. Niepert, M. Pontil, and X. He, "Learning discrete structures for graph neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 1972–1982.
[6] D. Yu, R. Zhang, Z. Jiang, Y. Wu, and Y. Yang, "Graph-revised convolutional network," 2019, *arXiv:1911.07123*.
[7] D. Zagner and S. Gannemann, "Adversarial attacks on graph neural networks via meta learning," in *Proc. Int. Conf. Learn. Representation*, 2019, pp. 1–15.
[8] H. Dai *et al.*, "Adversarial attack on graph structured data," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1115–1124.
[9] H. Chang *et al.*, "A restricted black-box adversarial framework towards attacking graph embedding models," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 04, pp. 3389–3396, 2020.
[10] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lia, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representation*, 2017, pp. 1–12.
[11] W. Jin, R. Barzilay, and T. Jaakkola, "Multi-objective molecule generation using interpretable substructures," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 4849–4859.
[12] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1263–1272.
[13] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–17.
[14] J. Lee, I. Lee, and J. Kang, "Self-attention graph pooling," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 3734–3743.
[15] W. Jin, Y. Ma, X. Liu, X. Tang, S. Wang, and J. Tang, "Graph structure learning for robust graph neural networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 66–74.

[16] J. Chen, B. Lei, Q. Song, H. Ying, D. Z. Chen, and J. Wu, "A hierarchical graph network for 3D object detection on point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 392–401.
[17] H. Lei, N. Akhtar, and A. Mian, "SegGCN: Efficient 3D point cloud segmentation with fuzzy spherical kernel," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11 611–11 620.
[18] Z.-H. Lin, S.-Y. Huang, and Y.-C. F. Wang, "Convolution in the cloud: Learning deformable kernels in 3D graph convolution networks for point cloud analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1800–1809.
[19] W. Shi and R. Rajkumar, "Point-GNN: Graph neural network for 3D object detection in a point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1711–1719.
[20] Q. Xu, X. Sun, C.-Y. Wu, P. Wang, and U. Neumann, "Grid-GCN for fast and scalable point cloud learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5661–5670.
[21] J. J. Irwin and B. K. Shoichet, "Zinc- A free database of commercially available compounds for virtual screening," *J. Chem. Inf. Model.*, vol. 45, no. 1, pp. 177–182, 2005.
[22] J. Li, Y. Rong, H. Cheng, H. Meng, W. Huang, and J. Huang, "Semi-supervised graph classification: A hierarchical graph perspective," in *Proc. World Wide Web Conf.*, 2019, pp. 972–982.
[23] B. Knyazev, G. W. Taylor, and M. R. Amer, "Understanding attention and generalization in graph neural networks," in *Proc. Conf. Neural Inf. Process. Syst.*, 2019, pp. 4204–4214.
[24] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," in *Proc. Annu. Allerton Conf. Commun., Control Comput.*, 1999, pp. 368–377.
[25] X. Peng, A. Kanazawa, S. Toyer, P. Abbeel, and S. Levine, "Variational discriminator bottleneck: Improving imitation learning, inverse RL, and GANs by constraining information flow," in *Proc. Int. Conf. Learn. Representation*, 2019, pp. 1–27.
[26] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," in *Proc. Int. Conf. Learn. Representation*, 2017, pp. 1–19.
[27] Y. Luo, P. Liu, T. Guan, J. Yu, and Y. Yang, "Significance-aware information bottleneck for domain adaptive semantic segmentation," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 6777–6786.
[28] A. Goyal *et al.*, "InfoBot: Transfer and exploration via the information bottleneck," in *Proc. Int. Conf. Learn. Representation*, 2019, pp. 1–13.
[29] M. Igl *et al.*, "Generalization in reinforcement learning with selective noise injection and information bottleneck," in *Proc. Neural Inf. Process. Syst.*, 2019, pp. 13979–13991.
[30] R. Wang, X. He, R. Yu, W. Qiu, B. An, and Z. Rabinovich, "Learning efficient multi-agent communication: An information bottleneck approach," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 9908–9918.
[31] T. Wu, H. Ren, P. Li, and J. Leskovec, "Graph information bottleneck," 2020, *arXiv: 2010.12811*.
[32] J. Yu, T. Xu, Y. Rong, Y. Bian, J. Huang, and R. He, "Graph information bottleneck for subgraph recognition," 2020, *arXiv: 2010.05563*.
[33] I. Armeni *et al.*, "3D semantic parsing of large-scale indoor spaces," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1534–1543.
[34] Y. Fang, K. Yu, R. Cheng, L. V. S. Lakshmanan, and X. Lin, "Efficient algorithms for densest subgraph discovery," *Proc. VLDB Endowment*, vol. 12, no. 11, pp. 1719–1732, 2019.
[35] A. Gionis and C. E. Tsourakakis, "Dense subgraph discovery: KDD 2015 tutorial," in *Proc. Knowl. Discov. Data Mining*, 2015, pp. 2313–2314.
[36] X. Yan and J. Yan, "gSpan: Graph-based substructure pattern mining," in *Proc. IEEE Int. Conf. Data Mining*, 2002, pp. 721–724.
[37] N. S. Ketkar, L. B. Holder, and D. Cook, "Subdue: Compression-based frequent pattern discovery in graph data," in *Proc. Int. Workshop Open Source Data Mining: Frequent Pattern Mining Implementations Knowl. Discov. Data Mining*, 2005, pp. 71–76.
[38] M. J. Zaki, "Efficiently mining frequent embedded unordered trees," *Fundam. Inf.*, vol. 66, no. 1/2, pp. 33–52, 2005.
[39] Y. Rong, W. Huang, T. Xu, and J. Huang, "DropEdge: Towards deep graph convolutional networks on node classification," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–17. [Online]. Available: https://openreview.net/forum?id=Hkx1qkrKPr
[40] J. Chen, T. Ma, and C. Xiao, "FastGCN: Fast learning with graph convolutional networks via importance sampling," 2018, *arXiv: 1801.10247*.
[41] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.
[42] W. Huang, T. Zhang, Y. Rong, and J. Huang, "Adaptive sampling towards fast graph representation learning," 2018, *arXiv: 1809.05343*.
[43] C. Zheng *et al.*, "Robust graph representation learning via neural sparsification," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 11458–11468.
[44] R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "GNNexplainer: Generating explanations for graph neural networks," in *Proc. Neural Inf. Process. Syst.*, 2019, pp. 9240–9251.
[45] E. Alsentzer, S. G. Finlayson, M. M. Li, and M. Zitnik, "Subgraph neural networks," 2020, *arXiv:2006.10538*.
[46] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.
[47] F.-Y. Sun, J. Hoffman, V. Verma, and J. Tang, "InfoGraph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–16.
[48] K. Qian, Y. Zhang, S. Chang, D. Cox, and M. Hasegawa-Johnson, "Unsupervised speech decomposition via triple information bottleneck," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 7836–7846.
[49] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.
[50] G. Li, M. Muller, A. Thabet, and B. Ghanem, "DeepGCNs: Can GCNs go as deep as CNNs?," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9267–9276.
[51] X. Xu and G. H. Lee, "Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13 706–13 715.
[52] X. Shi, X. Xu, K. Chen, L. Cai, C. S. Foo, and K. Jia, "Label-efficient point cloud semantic segmentation: An active learning approach," 2021, *arXiv:2101.06931*.
[53] H. Gao and S. Ji, "Graph u-nets," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2083–2092.
[54] F. Diehl, "Edge contraction pooling for graph neural networks," 2019, *arXiv:1905.10990*.
[55] M. D. Donsker and S. R. S. Varadhan, "Asymptotic evaluation of certain Markov process expectations for large time," *Commun. Pure Appl. Math.*, vol. 36, no. 2, pp. 183–212, 1983.
[56] A. Achille and S. Soatto, "Emergence of invariance and disentanglement in deep representations," *J. Mach. Learn. Res.*, vol. 19, no. 1, pp. 1947–1980, 2018.
[57] M. I. Belghazi *et al.*, "Mutual information neural estimation," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 530–539.
[58] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-Softmax," 2016, *arXiv:1611.01144*.
[59] M. Rupp, A. Tkatchenko, K.-R. Muller, and O. A. von Lilienfeld, "Fast and accurate modeling of molecular atomization energies with machine learning," *Phys. Rev. Lett.*, vol. 108, no. 5, 2012, Art. no. 058301.
[60] K. M. Borgwardt, C. S. Ong, S. Schanauer, S. V. N. Vishwanathan, A. J. Smola, and H.-P. Kriegel, "Protein function prediction via graph kernels," in *Proc. ISMB Suppl. Bioinf.*, 2005, pp. 47–56.
[61] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 4292–4293.



Junchi Yu received the BE degree from Wuhan University, Wuhan, China. He is currently working toward the MS degree with the Institute of Automation, Chinese Academy of Sciences, Beijing, China. He has authored or coauthored several papers in machine learning in top conferences including ICLR and IJCAI. His research interests include generative models, graph neural networks, and graph generations.

Tingyang Xu received the PhD degree from the University of Connecticut in 2017. In July 2017, he joined the Tencent AI Lab, where he is currently a senior researcher with Machine Learning Center, Tencent AI Lab, working on deep graph learning, graph generations, and applying the deep graph learning model to various applications, such as molecular generation and rumor detection. He has authored or coauthored several papers on data mining and machine learning top conferences, including KDD, WWW, NeurIPS, ICLR, CVPR, and ICML. His research interests include social network analysis, graph neural networks, and graph generations, with particular focus on design deep and complex graph learning models for molecular generations.

Yu Rong received the PhD degree from The Chinese University of Hong Kong in 2016. In June 2017, he joined the Tencent AI Lab, where he is currently a senior researcher with Machine Learning Center, working on building the large-scale graph learning framework and applying the deep graph learning model to various applications, such as ADMET prediction and malicious detection. He has authored or coauthored several papers on data mining and machine learning top conferences, including the Proceedings of KDD, WWW, NeurIPS, ICLR, CVPR, and ICCV. His research interests include social network analysis, graph neural networks, and large-scale graph systems.

Yatao Bian received the BScEng and MScEng degrees from Shanghai Jiao Tong University, and the PhD degree from the Institute for Machine Learning, ETH Zurich. Since June 2015, he has been an associated fellow with the Max Planck ETH Center for Learning Systems. In February 2020, he joined the Tencent AI Lab, where he is currently a senior researcher with Machine Learning Center, working on graph neural networks, energy-based learning, and optimization for machine learning and applications such as drug discovery, 3D protein modeling, and social network analysis. He was the recipient of the National Champion in AMD China Accelerated Computing Contest 2011–2012.

Junzhou Huang received the BE degree from the Huazhong University of Science and Technology, Wuhan, China, the MS degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, and the PhD degree in computer science from Rutgers, The State University of New Jersey. He was the director of the Machine Learning Center, Tencent AI Lab. He is currently an associate professor with the Computer Science and Engineering Department, the University of Texas at Arlington. His research interests include machine learning, computer vision, and imaging informatics. He was selected as one of the ten emerging leaders in multimedia and signal processing by the IBM T.J. Watson Research Center in 2010. He was the recipient of the MICCAI Young Scientist Award 2010, the FIMH Best Paper Award 2011, the MICCAI Young Scientist Award Finalist 2011, the STMI Best Paper Award 2012, the NIPS Best Reviewer Award 2013, the MICCAI Best Student Paper Award Finalist 2014, the MICCAI Best Student Paper Award 2015, and the NSF CAREER Award in 2016.

Ran He (Senior Member, IEEE) received the BE and MS degrees in computer science from the Dalian University of Technology in 2001 and 2004, respectively, and the PhD degree in pattern recognition and intelligent systems from CASIA in 2009. Since September 2010, he has been with NLPR where he is currently a full Professor. His research interests include information theoretic learning, pattern recognition, and computer vision. He was an associate editor for *Pattern Recognition* (Elsevier) and was on the program committee of several conferences. He is a fellow of IAPR.