PRELIMINARY PREPRINT VERSION: DO NOT CITE
The AAAI Digital Library will contain the published
version some time after the conference.

Graph Structure Learning with Variational Information Bottleneck

Qingyun Sun[123], **Jianxin Li**[12], **Hao Peng**[1], **Jia Wu**[4], **Xingcheng Fu**[1], **Cheng Ji**[1], **Philip S. Yu**[5]
[1] Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing 100191, China
[2] School of Computer Science and Engineering, Beihang University, Beijing 100191, China
[3] Shenyuan Honors College, Beihang University, Beijing 100191, China
[4] School of Computing, Macquarie University, Sydney, Australia
[5] Department of Computer Science, University of Illinois at Chicago, Chicago, USA
{sunqy, lijx, penghao, fuxc, jicheng}@act.buaa.edu.cn, jia.wu@mq.edu.au, psyu@uic.edu

Abstract

Graph Neural Networks (GNNs) have shown promising results on a broad spectrum of applications. Most empirical studies of GNNs directly take the observed graph as input, assuming the observed structure perfectly depicts the accurate and complete relations between nodes. However, graphs in the real-world are inevitably noisy or incomplete, which could even exacerbate the quality of graph representations. In this work, we propose a novel **V**ariational **I**nformation **B**ottleneck guided **G**raph **S**tructure **L**earning framework, namely **VIB-GSL**, in the perspective of information theory. VIB-GSL is the first attempt to advance the Information Bottleneck (IB) principle for graph structure learning, providing a more elegant and universal framework for mining underlying task-relevant relations. VIB-GSL learns an informative and compressive graph structure to distill the actionable information for specific downstream tasks. VIB-GSL deduces a variational approximation for irregular graph data to form a tractable IB objective function, which facilitates training stability. Extensive experimental results demonstrate that the superior effectiveness and robustness of VIB-GSL.

1 Introduction

Recent years have seen a significant growing amount of interest in graph representation learning (Zhang et al. 2018; Tong et al. 2021), especially in efforts devoted to developing more effective graph neural networks (GNNs) (Zhou et al. 2020). Despite GNNs' powerful ability in learning graph representations, most of them directly take the observed graph as input, assuming the observed structure perfectly depicts the accurate and complete relations between nodes. However, these raw graphs are naturally admitted from network-structure data (e.g., social network) or constructed from the original feature space by some pre-defined rules, which are usually independent of the downstream tasks and lead to the gap between the raw graph and the optimal graph for specific tasks. Moreover, most of graphs in the real-word are noisy or incomplete due to the error-prone data collection (Chen, Wu, and Zaki 2020), which could even exacerbate the quality of representations produced by GNNs (Zügner, Akbarnejad, and Günnemann 2018; Sun et al. 2018). It's also found that the properties of a graph are mainly determined by some critical structures rather than the whole graph (Sun et al. 2021; Peng et al. 2021). Furthermore, many graph enhanced applications (e.g., text classification (Li et al. 2020) and vision navigation (Gao et al. 2021)) may only have data without graph-structure and require additional graph construction to perform representation learning. The above issues pose a great challenge for applying GNNs to real-world applications, especially in some risk-critical scenarios. Therefore, *learning a task-relevant graph structure is a fundamental problem for graph representation learning*.

To adaptively learn graph structures for GNNs, many graph structure learning methods (Zhu et al. 2021; Franceschi et al. 2019; Chen, Wu, and Zaki 2020) are proposed, most of which optimize the adjacency matrix along with the GNN parameters toward downstream tasks with assumptions (e.g., community) or certain constraints (e.g., sparsity, low-rank, and smoothness) on the graphs. However, these assumptions or explicit certain constraints may not be applicable to all datasets and tasks. There is still a lack of a general framework that can mine underlying relations from the essence of representation learning.

Recalling the above problems, the key of structure learning problem is learning the underlying relations invariant to task-irrelevant information. Information Bottleneck (IB) principle (Tishby, Pereira, and Bialek 2000) provides a framework for constraining such task-irrelevant information retained at the output by trading off between prediction and compression. Specifically, the IB principle seeks for a representation Z that is maximally informative about target Y (i.e., maximize mutual information $I(Y;Z)$) while being minimally informative about input data X (i.e., minimize mutual information $I(X;Z)$). Based on the IB principle, the learned representation is naturally more robust to data noise. IB has been applied to representation learning (Kim et al. 2021; Jeon et al. 2021; Pan et al. 2020; Bao 2021; Dubois et al. 2020) and numerous deep learning tasks such as model ensemble (Sinha et al. 2020), fine-tuning (Mahabadi, Belinkov, and Henderson 2021), salient region discovery (Zhmoginov, Fischer, and Sandler 2020).

In this paper, we advance the IB principle for graph to solve the graph structure learning problem. We propose a novel **V**ariational **I**nformation **B**ottleneck guided **G**raph **S**tructure **L**earning framework, namely **VIB-GSL**. VIB-

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

GSL employs the irrelevant feature masking and structure learning method to generate a new IB-Graph G_{IB} as a bottleneck to distill the actionable information for the downstream task. VIB-GSL consists of three steps: (1) the IB-Graph generator module learns the IB-graph G_{IB} by masking irrelevant node features and learning a new graph structure based on the masked feature; (2) the GNN module takes the IB-graph G_{IB} as input and learns the distribution of graph representations; (3) the graph representation is sampled from the learned distribution with a reparameterization trick and then used for classification. The overall framework can be trained efficiently with the supervised classification loss and the distribution KL-divergence loss for the IB objective. The main contributions are summarized as follows:

- VIB-GSL advances the Information Bottleneck principle for graph structure learning, providing an elegant and universal framework in the perspective of information theory.
- VIB-GSL is model-agnostic and has a tractable variational optimization upper bound that is easy and stable to optimize. It is sufficient to plug existing GNNs into the VIB-GSL framework to enhance their performances.
- Extensive experiment results in graph classification and graph denoising demonstrate that the proposed VIB-GSL enjoys superior effectiveness and robustness compared to other strong baselines.

2 Background and Problem Formulation

2.1 Graph Structure Learning

Graph structure learning (Zhu et al. 2021) targets jointly learning an optimized graph structure and corresponding representations to improving the robustness of GNN models. In this work, we focus on graph structure learning for graph-level tasks.

Let $G \in \mathbb{G}$ be a graph with label $Y \in \mathbb{Y}$. Given a graph $G = (X, A)$ with node set V, node feature matrix $X \in \mathbb{R}^{|V| \times d}$, and adjacency matrix $A \in \mathbb{R}^{|V| \times |V|}$, or only given a feature matrix X, the graph structure learning problem we consider in this paper can be formulated as producing an optimized graph $G^* = (X^*, A^*)$ and its corresponding node/graph representations $Z^* = f(G^*)$, with respect to the downstream graph-level tasks.

2.2 Information Bottleneck

The Information Bottleneck (Tishby, Pereira, and Bialek 2000) seeks the balance between data fit and generalization using the mutual information as both cost function and regularizer. We will use the following standard quantities in the information theory (Cover 1999) frequently: Shannon entropy $H(X) = \mathbb{E}_{X \sim p(X)}[-\log p(X)]$, cross entropy $H(p(X), q(X)) = \mathbb{E}_{X \sim p(X)}[-\log q(X)]$, Shannon mutual information $I(X;Y) = H(X) - H(X|Y)$, and Kullback Leiber divergence $\mathcal{D}_{\mathrm{KL}}(p(X)||q(X) = \mathbb{E}_{X \sim p(X)} \log \frac{p(X)}{q(X)}$. Following standard practice in the IB literature (Tishby, Pereira, and Bialek 2000), given data X, representation Z of X and target Y, (X, Y, Z) are following the Markov Chain $< Y \rightarrow X \rightarrow Z >$.

Definition 1 (Information Bottleneck)**.** *For the input data X and its label Y, the **Information Bottleneck** principle aims to learn the minimal sufficient representation Z:*

$$Z = \arg\min_{Z} -I(Z;Y) + \beta I(Z;X), \tag{1}$$

where β is the Lagrangian multiplier trading off sufficiency and minimality.

Deep VIB (Alemi et al. 2016) proposed a variational approximation to the IB objective by parameterizing the distribution via a neural network:

$$\begin{aligned}\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \int dZ p(Z|X_i) \log q(Y_i|Z) \\ + \beta \mathcal{D}_{\mathrm{KL}} \left(p(Z|X_i), r(Z) \right),\end{aligned} \tag{2}$$

where $q(Y_i|Z)$ is the variational approximation to $p(Y_i|Z)$ and $r(Z)$ is the variational approximation of $p(Z)$.

The IB framework has received significant attention in machine learning and deep learning (Alemi et al. 2016; Saxe et al. 2019). As for irregular graph data, there are some recent works (Wu et al. 2020; Yu et al. 2020; Yang et al. 2021; Yu et al. 2021) introducing the IB principle to graph learning. GIB (Wu et al. 2020) extends the general IB to graph data with regularization of the structure and feature information for robust node representations. SIB (Yu et al. 2020, 2021) was proposed for the subgraph recognition problem. HGIB (Yang et al. 2021) was proposed to implement the consensus hypothesis of heterogeneous information networks in an unsupervised manner. We illustrate the difference between related graph IB methods and our method in Section 3.3.

3 Variational Information Bottleneck Guided Graph Structure learning

In this section, we elaborate the proposed VIB-GSL, a novel variational information bottleneck principle guided graph structure learning framework. First, we formally define the IB-Graph and introduce a tractable upper bound for IB objective. Then, we introduce the graph generator to learn the optimal IB-Graph as a bottleneck and give the overall framework of VIB-GSL. Lastly, we compare VIB-GSL with two graph IB methods to illustrate its difference and properties.

3.1 Graph Information Bottleneck

In this work, we focus on learning an optimal graph $G_{\mathrm{IB}} = (X_{\mathrm{IB}}, A_{\mathrm{IB}})$ named IB-Graph for G, which is compressed with minimum information loss in terms of G's properties.

Definition 2 (IB-Graph)**.** *For a graph $G = (X, A)$ and its label Y, the optimal graph $G_{\mathrm{IB}} = (X_{\mathrm{IB}}, A_{\mathrm{IB}})$ found by Information Bottleneck is denoted as **IB-Graph**:*

$$G_{\mathrm{IB}} = \arg\min_{G_{\mathrm{IB}}} -I(G_{\mathrm{IB}};Y) + \beta I(G_{\mathrm{IB}};G), \tag{3}$$

where X_{IB} is the task-relevant feature set and A_{IB} is the learned task-relevant graph adjacency matrix.

Intuitively, the first term $-I(G_{\mathrm{IB}};Y)$ is the *prediction* term, which encourages that essential information to the graph property is preserved. The second term $I(G_{\mathrm{IB}};G)$ is the *compression* term, which encourages that label-irrelevant information in G is dropped. And the Lagrangian multiplier β indicates the degree of information compression, where larger β indicates more information in G was retained to G_{IB}. Suppose $G_n \in \mathbb{G}$ is a task-irrelevant nuisance in G, the learning procedure of G_{IB} follows the Markov Chain $< (Y, G_n) \rightarrow G \rightarrow G_{\mathrm{IB}} >$. IB-Graph only preserves the task-relevant information in the observed graph G and is invariant to nuisances in data.

Lemma 1 (**Nuisance Invariance**). *Given a graph $G \in \mathbb{G}$ with label $Y \in \mathbb{Y}$, let $G_n \in \mathbb{G}$ be a task-irrelevant nuisance for Y. Denote G_{IB} as the IB-Graph learned from G, then the following inequality holds:*

$$I(G_{\mathrm{IB}};G_n) \leq I(G_{\mathrm{IB}};G) - I(G_{\mathrm{IB}};Y) \tag{4}$$

Please refer to the Technical Appendix for the detailed proof. Lemma 1 indicates that optimizing the IB objective in Eq. (3) is equivalent to encourage G_{IB} to be less related to task-irrelevant information in G, leading to the nuisance-invariant property of IB-Graph.

Due to the non-Euclidean nature of graph data and the intractability of mutual information, the IB objective in Eq. (3) is hard to optimize directly. Therefore, we introduce two tractable variational upper bounds of $-I(G_{\mathrm{IB}};Y)$ and $I(G_{\mathrm{IB}};G)$, respectively. First, we examine the prediction term $-I(G_{\mathrm{IB}};Y)$ in Eq. (3), which encourages G_{IB} is informative of Y. Please refer to Technical Appendix for the detailed proof of Proposition 1.

Proposition 1 (**Upper bound of** $-I(G_{\mathrm{IB}};Y)$). *For graph $G \in \mathbb{G}$ with label $Y \in \mathbb{Y}$ and IB-Graph G_{IB} learned from G, we have*

$$-I(Y;G_{\mathrm{IB}}) \leq -\iint p(Y,G_{\mathrm{IB}}) \log q_\theta(Y|G_{\mathrm{IB}}) dY\, dG_{\mathrm{IB}} + H(Y), \tag{5}$$

where $q_\theta(Y|G_{\mathrm{IB}})$ is the variational approximation of the true posterior $p(Y|G_{\mathrm{IB}})$.

Then we examine the compression term $I(G_{\mathrm{IB}};G)$ in Eq. (3), which constrains the information that G_{IB} receives from G. Please refer to the Technical Appendix for the detailed proof of Proposition 2.

Proposition 2 (**Upper bound of** $I(G_{\mathrm{IB}};G)$). *For graph $G \in \mathcal{G}$ and IB-Graph G_{IB} learned from G, we have*

$$I(G_{\mathrm{IB}};G) \leq \iint p(G_{\mathrm{IB}},G) \log \frac{p(G_{\mathrm{IB}}|G)}{r(G_{\mathrm{IB}})} dG_{\mathrm{IB}} dG, \tag{6}$$

where $r(G_{\mathrm{IB}})$ is the variational approximation to the prior distribution $p(G_{\mathrm{IB}})$ of G_{IB}.

Finally, plug Eq. (5) and Eq. (6) into Eq. (3) to derive the following objective function, which we try to minimize:

$$\begin{aligned} &-I(G_{\mathrm{IB}};Y) + \beta I(G_{\mathrm{IB}};G) \\ &\leq -\iint p(Y,G_{\mathrm{IB}}) \log q_\theta(Y|G_{\mathrm{IB}}) dY\, dG_{\mathrm{IB}} \\ &\quad + \beta \iint p(G_{\mathrm{IB}},G) \log \frac{p(G_{\mathrm{IB}}|G)}{r(G_{\mathrm{IB}})} dG_{\mathrm{IB}} dG. \end{aligned} \tag{7}$$

3.2 Instantiating the VIB-GSL Framework

Following the theory discussed in Section 3.1, we first obtain the graph representation Z_{IB} of G_{IB} to optimize the IB objective in Eq. (7). We assume that there is no information loss during this process, which is the general practice of mutual information estimation (Tian et al. 2020). Therefore, we have $I(G_{\mathrm{IB}};Y) \approx I(Z_{\mathrm{IB}};Y)$ and $I(G_{\mathrm{IB}};G) \approx I(Z_{\mathrm{IB}};G)$. In practice, the integral over G_{IB} and G can be approximated by Monte Carlo sampling (Shapiro 2003) on all training samples $\{G_i \in \mathbb{G}, Y_i \in \mathbb{Y}, i = 1, \ldots, N\}$.

$$\begin{aligned} &-I(G_{\mathrm{IB}};Y) + \beta I(G_{\mathrm{IB}};G) \approx -I(Z_{\mathrm{IB}};Y) + \beta I(Z_{\mathrm{IB}};G) \\ &\leq \frac{1}{N}\sum_{i=1}^{N} \left\{ -\log q_\theta(Y_i|Z_{\mathrm{IB}i}) + \beta p(Z_{\mathrm{IB}i}|G_i) \log \frac{p(Z_{\mathrm{IB}i}|G_i)}{r(Z_{\mathrm{IB}})} \right\}. \end{aligned} \tag{8}$$

As shown in Figure 1, VIB-GSL consists of three steps:

Step-1: Generate IB-Graph G_{IB}.
We introduce an IB-Graph generator to generate the IB-graph G_{IB} for the input graph G. Following the assumption that nuisance information exists in both irrelevant feature and structure, the generation procedure consists of feature masking and structure learning.

Feature Masking. We first use a feature masking scheme to discretely drop features that are irrelevant to the downstream task, which is formulated as:

$$X_{\mathrm{IB}} = \{X_i \odot M, i = 1, 2, \cdots, |V|\}, \tag{9}$$

where $M \in \mathbb{R}^d$ is a learnable binary feature mask and $\odot$ is the element-wise product. Intuitively, if a particular feature is not relevant to task, the corresponding weight in M takes value close to zero. We can reparameterize X_{IB} using the reparameterization trick (Kingma and Welling 2013) to backpropagate through a d-dimensional random variable:

$$X_{\mathrm{IB}} = X_r + (X - X_r) \odot M, \tag{10}$$

where X_r is a random variable sampled from the empirical distribution of X.

Structure Learning. We model all possible edges as a set of mutually independent Bernoulli random variables parameterized by the learned attention weights π:

$$A_{\mathrm{IB}} = \bigcup_{u,v \in V} \{a_{u,v} \sim \mathrm{Ber}(\pi_{u,v})\}. \tag{11}$$

For each pair of nodes, we optimized the edge sampling probability π jointly with the graph representation learning. $\pi_{u,v}$ describes the task-specific quality of edge (u, v) and smaller $\pi_{u,v}$ indicates that the edge (u, v) is more likely to

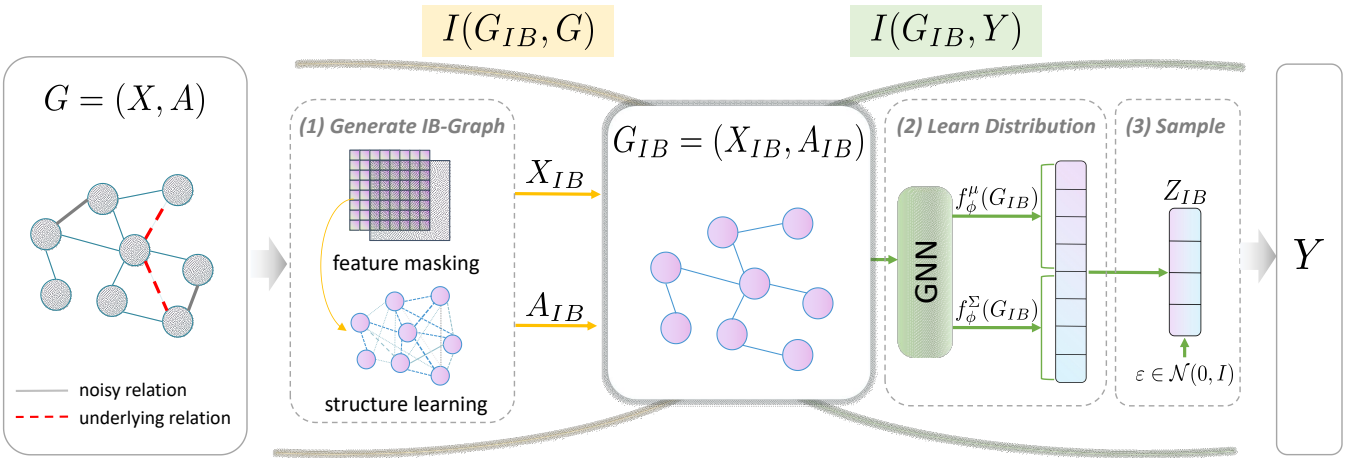


Figure 1: Overview of VIB-GSL. Given G as input, VIB-GSL consists of the following three steps: (1) Generate IB-Graph: the IB-Graph generator learns an IB-Graph G_{IB} by masking irrelevant features and learning a new structure; (2) Learn distribution of IB-Graph representation: the GNN module learns the distribution of IB-Graph representation Z_{IB}; (3) Sample IB-Graph representation: Z_{IB} is sampled from the learned distribution by a reparameterization trick for classification.

be noise and should be assigned small weight or even be removed. For a pair of nodes (u, v), the edge sampling probability $\pi_{u,v}$ is calculated by:

$$\begin{aligned} Z(u) &= \mathbf{NN}\left(X_{\mathrm{IB}}(u)\right), \\ \pi_{u,v} &= \text{sigmoid}\left(Z(u)Z(v)^{\mathrm{T}}\right), \end{aligned} \tag{12}$$

where $\mathbf{NN}(\cdot)$ denotes a neural network and we use a two-layer perceptron in this work. One issue is that A_{IB} is not differentiable with respect to π as Bernoulli distribution. We thus use the concrete relaxation (Jang, Gu, and Poole 2017) of the Bernoulli distribution to update π:

$$\mathrm{Ber}(\pi_{u,v}) \approx \text{sigmoid}\left(\frac{1}{t}\left(\log\frac{\pi_{u,v}}{1-\pi_{u,v}} + \log\frac{\epsilon}{1-\epsilon}\right)\right), \tag{13}$$

where $\epsilon \sim \mathrm{Uniform}(0, 1)$ and $t \in \mathbb{R}^+$ is the temperature for the concrete distribution. After concrete relaxation, the binary entries $a_{u,v}$ from a Bernoulli distribution are transformed into a deterministic function of $\pi_{u,v}$ and ϵ.

The graph structure after the concrete relaxation is a weighted fully connected graph, which is computationally expensive. We hence extract a symmetric sparse adjacency matrix by masking off those elements which are smaller than a non-negative threshold a_0.

Step-2: Learn Distribution of IB-Graph Representation. For the compression term $I(Z_{\mathrm{IB}}; G)$ in Eq. (8), we consider a parametric Gaussian distribution as prior $r(Z_{\mathrm{IB}})$ and $p(Z_{\mathrm{IB}}|G)$ to allow an analytic computation of Kullback Leibler (KL) divergence (Hershey and Olsen 2007):

$$\begin{aligned} r\left(Z_{\mathrm{IB}}\right) &= \mathcal{N}\left(\mu_0, \Sigma_0\right), \\ p\left(Z_{\mathrm{IB}}|G\right) &= \mathcal{N}\left(f_\phi^\mu\left(G_{\mathrm{IB}}\right), f_\phi^\Sigma\left(G_{\mathrm{IB}}\right)\right), \end{aligned} \tag{14}$$

where $\mu \in \mathbb{R}^K$ and $\Sigma \in \mathbb{R}^{K\times K}$ is the mean vector and the diagonal co-variance matrix of Z_{IB} encoded by $f_\phi(G_{\mathrm{IB}})$. The dimensionality of Z_{IB} is denoted as K, which specifies the bottleneck size. We model the $f_\phi(G_{\mathrm{IB}})$ as a graph neural network (GNN) with weights ϕ, where $f_\phi^\mu(G_{\mathrm{IB}})$ and $f_\phi^\Sigma(G_{\mathrm{IB}})$ are the $2K$-dimensional output value of the GNN:

$$\begin{aligned} \forall u \in V, Z_{\mathrm{IB}}(u) &= \mathbf{GNN}\left(X_{\mathrm{IB}}, A_{\mathrm{IB}}\right), \\ \left(f_\phi^\mu\left(G_{\mathrm{IB}}\right), f_\phi^\Sigma\left(G_{\mathrm{IB}}\right)\right) &= \mathbf{Pooling}\left(\left\{Z_{\mathrm{IB}}(u), \forall u \in V\right\}\right), \end{aligned} \tag{15}$$

where the first K-dimension outputs encode μ and the remaining K-dimension outputs encode Σ (we use a softplus transform for $f_\phi^\Sigma(G_{\mathrm{IB}})$ to ensure the non-negativity). We treat $r(Z_{\mathrm{IB}})$ as a fixed d-dimensional spherical Gaussian $r(Z_{\mathrm{IB}}) = \mathcal{N}(Z_{\mathrm{IB}}|0, \mathrm{I})$ as in (Alemi et al. 2016).

Step-3: Sample IB-Graph Representation. To obtain Z_{IB}, we can use the reparameterization trick (Kingma and Welling 2013) for gradients estimation:

$$Z_{\mathrm{IB}} = f_\phi^\mu(G_{\mathrm{IB}}) + f_\phi^\Sigma(G_{\mathrm{IB}}) \odot \varepsilon, \tag{16}$$

where $\varepsilon \in \mathcal{N}(0, \mathrm{I})$ is an independent Gaussian noise and $\odot$ denotes the element-wise product. By using the reparameterization trick, randomness is transferred to ε, which does not affect the back-propagation. For the first term $I(Z_{\mathrm{IB}}, Y)$ in Eq. (8), $q_\theta(Y|Z_{\mathrm{IB}})$ outputs the label distribution of learned graph G_{IB} and we model it as a multi-layer perceptron classifier with parameters θ. The multi-layer perceptron classifier takes Z_{IB} as input and outputs the predicted label.

Training Objective. We can efficiently compute the upper bounds in Eq. (8) on the training data samples using the gradient descent based backpropagation techniques, as illustrated in Algorithm 1. The overall loss is:

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}}(Z_{\mathrm{IB}}, Y) + \beta\mathcal{D}_{\mathrm{KL}}\left(p\left(Z_{\mathrm{IB}}|G\right)||r\left(Z_{\mathrm{IB}}\right)\right), \tag{17}$$

where $\mathcal{L}_{\mathrm{CE}}$ is the cross-entropy loss and $\mathcal{D}_{\mathrm{KL}}(\cdot||\cdot)$ is the KL divergence. The variational approximation proposed above

Algorithm 1: The overall process of VIB-GSL

Input: Graph $G = (X, A)$ with label Y; Number of training epochs E;

Output: IB-graph G_{IB}, predicted label $\hat{Y}$

Parameter initialization;
for $e = 1, 2, \cdots, E$ **do**
  // Learn IB-Graph
  $X_{\mathrm{IB}} \leftarrow \{X_i \odot M, i \in |V|\}$;
  $A_{\mathrm{IB}} \leftarrow \bigcup_{u,v \in V} \{a_{u,v} \sim \mathrm{Ber}(\pi_{u,v})\}$;
  $G_{\mathrm{IB}} \leftarrow (X_{\mathrm{IB}}, A_{\mathrm{IB}})$;
  // Learn distribution
  Encode $(f_\phi^\mu(G_{\mathrm{IB}}), f_\phi^\Sigma(G_{\mathrm{IB}}))$ by a GNN;
  // Sample graph representation
  Reparameterize $Z_{\mathrm{IB}} = f_\phi^\mu(G_{\mathrm{IB}}) + f_\phi^\Sigma(G_{\mathrm{IB}}) \odot \varepsilon$;
  // Optimize
  $\mathcal{L} = \mathcal{L}_{\mathrm{CE}}(Z_{\mathrm{IB}}, Y) + \beta \mathcal{D}_{\mathrm{KL}}\left(p\left(Z_{\mathrm{IB}}|G\right) || r\left(Z_{\mathrm{IB}}\right)\right)$;
  Update model parameters to minimize $\mathcal{L}$.
end

facilitates the training stability effectively, as shown in Section 4.2. We also analyze the impact of compression coefficient β on performance and learned structure in Section 4.2.

Property of VIB-GSL Different with traditional GNNs and graph structure learning methods (e.g., IDGL (Chen, Wu, and Zaki 2020), NeuralSparse (Zheng et al. 2020)), VIB-GSL is independent of the original graph structure since it learns a new graph structure. This property renders VIB-GSL extremely robust to noisy information and structure perturbations, which is verified in Section 4.2.

3.3 Comparison with multiple related methods.

In this subsection, we discuss the relationship between the proposed VIB-GSL and two related works using the IB principle for graph representation learning, i.e., GIB (Wu et al. 2020) and SIB (Yu et al. 2020). Remark that VIB-GSL follows the Markov Chain $< (Y, G_n) \rightarrow G \rightarrow G_{\mathrm{IB}} >$.

VIB-GSL vs. GIB GIB (Wu et al. 2020) aims to learn robust node representations Z by the IB principle following the Markov Chain $< (Y, G_n) \rightarrow G \rightarrow Z >$. Specifically, GIB regularizes and controls the structure and feature information in the computation flow of latent representations layer by layer. Our VIB-GSL differs in that we aim to learn an optimal graph explicitly, which is more interpretable than denoising in the latent space. Besides, our VIB-GSL focuses on graph-level tasks while GIB focuses on node-level ones.

VIB-GSL vs. SIB SIB (Yu et al. 2020) aims to recognise the critical subgraph G_{sub} for input graph following the Markov Chain $< (Y, G_n) \rightarrow G \rightarrow G_{sub} >$. Our VIB-GSL aims to learn a new graph structure and can be applied for non-graph structured data. Moreover, SIB directly estimates the mutual information between subgraph and graph by MINE (Belghazi et al. 2018) and uses a bi-level optimization scheme for the IB objective, leading to an unstable and inefficient training process. Our VIB-GSL is more stable to train with the tractable variational approximation, which is demonstrated by experiments in Figure 5.

4 Experiments

We evaluate VIB-GSL[1] on two tasks: graph classification and graph denoising, to verify whether VIB-GSL can improve the effectiveness and robustness of graph representation learning. Then we analyze the impact of information compression quantitatively and qualitatively.

4.1 Experimental Setups

Datasets. We empirically perform experiments on VIB-GSL on four widely-used social datasets including IMDB-B, IMDB-M, REDDIT-B, and COLLAB (Rossi and Ahmed 2015). We choose the social datasets for evaluation because much noisy information may exist in social interactions.

Baselines. We compare the proposed VIB-GSL with a number of graph-level structure learning baselines, including NeuralSparse (Zheng et al. 2020), SIB (Yu et al. 2020) and IDGL (Chen, Wu, and Zaki 2020), to demonstrate the effectiveness and robustness of VIB-GSL. We do not include GIB in our baselines since it focuses on node-level representation learning. Similar with SIB (Yu et al. 2020), we plug various GNN backbones[2] into VIB-GSL including GCN (Kipf and Welling 2016), GAT (Veličković et al. 2017), GIN (Xu et al. 2019) to see whether the VIB-GSL can boost the performance of graph classification or not. For a fair comparison, we use the mean pooling operation to obtain the graph representation and use a 2-layer perceptron as the graph classifier for all baselines.

Parameter Settings. We set both the information bottleneck size K and the embedding dimension of baseline methods as 16. For VIB-GSL, we set $t = 0.1$ in Eq. (13), $a_0 = 0.1$ and perform hyperparameter search of $\beta \in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$ for each dataset.

4.2 Results and Analysis

Graph Classification. We first examine VIB-GSL's capability of improving graph classification. We perform 10-fold cross-validation and report the average accuracy and the standard deviation across the 10 folds in Table 1, where Δ denotes the performance improvement for specific backbone and "–" indicates that there is no performance improvement for backbones without structure learner. The best results in each backbone group are underlined and the best results of each dataset are shown in bold. As shown in Table 1, the proposed VIB-GSL consistently outperforms all baselines on all datasets by a large margin. Generally, the graph sparsification models (i.e., NeuralSparse and SIB) show only a small improvement in accuracy and even have a negative impact on performance (e.g., on COLLAB), which is because they are constrained by the observed structures without mining underlying relations. The performance superiority of VIB-GSL over different GNN backbones implies that

[1] Code is available at https://github.com/VIB-GSL/VIB-GSL.

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

Table 1: Summary of graph classification results: "average accuracy ± standard deviation" and "improvements" (%). Underlined: best performance of specific backbones, **bold**: best results of each dataset.

Structure Learner	Backbone	IMDB-B Accuracy	IMDB-B Δ	IMDB-M Accuracy	IMDB-M Δ	REDDIT-B Accuracy	REDDIT-B Δ	COLLAB Accuracy	COLLAB Δ
N/A	GCN	70.7±3.7	-	49.7±2.1	-	73.6±4.5	-	77.6±2.6	-
	GAT	71.3±3.5	-	50.9±2.7	-	73.1±2.6	-	75.4±2.4	-
	GIN	72.1±3.8	-	49.7±0.4	-	85.4±3.0	-	78.8±1.4	-
NeuralSparse	GCN	72.0±2.6	↑1.3	50.1±3.1	↑0.4	72.1±5.2	↓1.5	76.0±2.0	↓1.6
	GAT	73.4±2.2	↑2.1	53.7±3.1	↑2.8	74.3±3.1	↑1.2	75.4±5.8	0.0
	GIN	73.8±1.6	↑1.7	54.2±5.4	↑4.5	86.2±2.7	↑0.8	76.6±2.1	↓2.2
SIB	GCN	72.2±3.9	↑1.5	51.8±3.9	↑2.1	76.7±3.0	↑3.1	76.3±2.3	↓1.3
	GAT	72.9±4.6	↑1.6	51.3±2.4	↑0.4	75.3±4.7	↑2.2	77.3±1.9	↑1.9
	GIN	73.7±7.0	↑1.6	51.6±4.8	↑1.9	85.7±3.5	↑0.3	77.2±2.3	↓1.6
IDGL	GCN	72.2±4.2	↑1.5	52.1±2.4	↑2.4	75.1±1.4	↑1.5	78.1±2.1	↑0.5
	GAT	71.5±4.6	↑0.2	51.8±2.4	↑0.9	76.2±2.5	↑3.1	76.8±4.4	↑1.4
	GIN	74.1±3.2	↑2.0	51.1±2.1	↑1.4	85.7±3.5	↑0.3	76.7±3.8	↓2.1
VIB-GSL	GCN	74.1±3.3	↑3.4	54.3±1.7	↑4.6	77.5±2.4	↑3.9	78.3±1.4	↑0.7
	GAT	75.2±2.7	↑3.9	54.1±2.7	↑3.2	78.1±2.5	↑5.0	79.1±1.2	↑3.7
	GIN	**77.1±1.4**	**↑5.0**	**55.6±2.0**	**↑5.9**	**88.5±1.8**	**↑3.1**	**79.3±2.1**	**↑0.5**

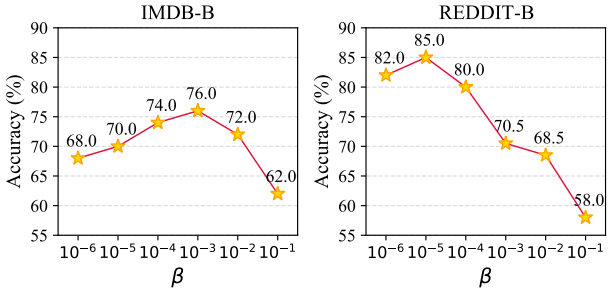


Figure 2: Impact of β on IMDB-B and REDDIT-B.

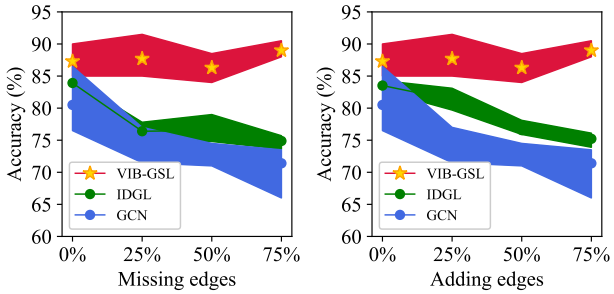


Figure 3: Test accuracy (± standard deviation) in percent for the edge attack scenarios on REDDIT-B.

VIB-GSL can learn better graph structure to improve the representation quality.

Graph Denoise. To evaluate the robustness of VIB-GSL, we generate a synthetics dataset by deleting or adding edges on REDDIT-B. Specifically, for each graph in the dataset, we randomly remove (if edges exist) or add (if no such edges) 25%, 50%, 75% edges. The reported results are the mean accuracy (solid lines) and standard deviation (shaded region) over 5 runs. As shown in Figure 3, the classification accuracy of GCN dropped by 5% with 25% missing edges and dropped by 10% with 25% noisy edges, indicating that GNNs are indeed sensitive to structure noise. Since the proposed VIB-GSL does not depend on the original graph structure, it achieves better results without performance degradation. IDGL is still sensitive to structure noise since it iteratively updates graph structure based on node embeddings, which is tightly dependent on the observed structure.

Parameter Sensitivity: Trade Off between Prediction and Compression. We explore the influence of the Lagrangian multiplier β trading off prediction and compression in Eq. (3) and Eq. (8). Note that there is a relationship between increasing β and decreasing K (Shamir, Sabato, and Tishby 2010), and the following analysis is with $K = 16$. Figure 2 depicts the changing trend of graph classification accuracy on IMDB-B and REDDIT-B. Based on the results, we make the following observations: (1) Remarkably, the graph classification accuracies of VIB-GSL variation across different β collapsed onto a hunchback shape on both datasets. The accuracy first increases with the increase of β, indicating that removing irrelevant information indeed enhances the graph representation learning. Then the accuracy progressively decreases and reaches very low values, indicating that excessive information compression will lose effective information. (2) Appropriate value of β can greatly increase the model's performance. VIB-GSL achieves the best balance of prediction and compression with $\beta = 10^{-3}$ and $\beta = 10^{-5}$ on IMDB-B and REDDIT-B, respectively. This indicates that different dataset consists of different percent of task-irrelevant information and hence needs a different degree of information compression.

Graph Visualization. To examine the graph structure changes brought by VIB-GSL intuitively, we present two samples from the IMDB-B dataset and visualize the origi-

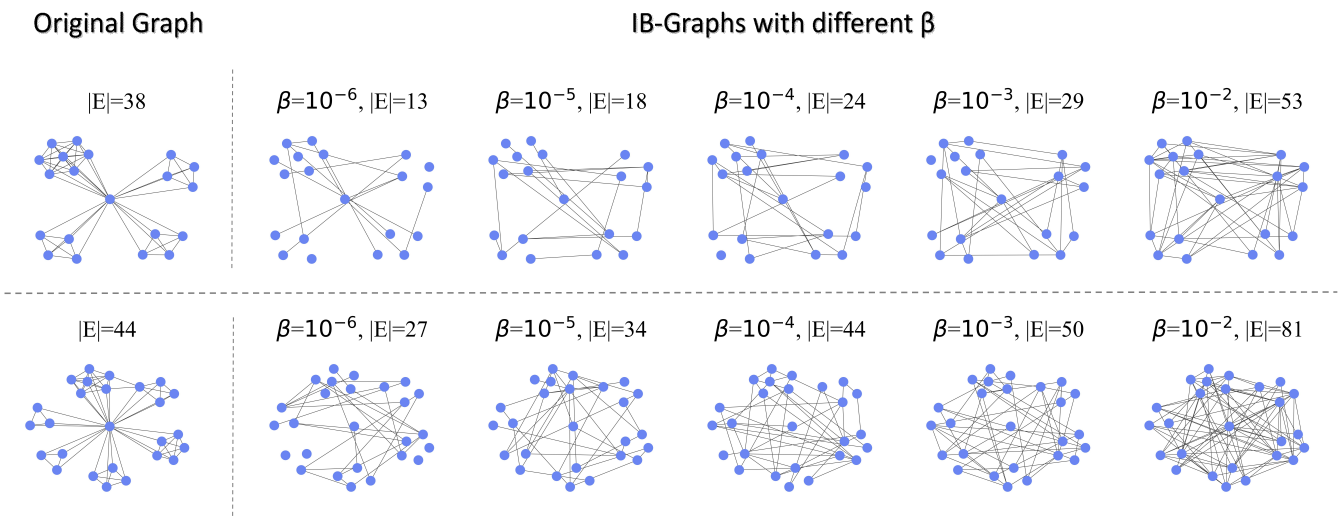
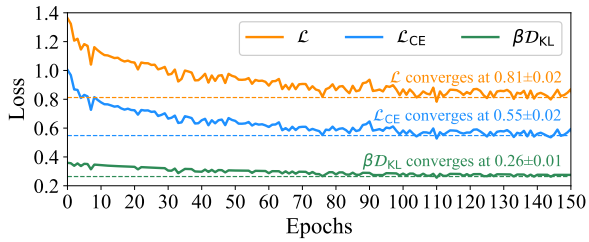


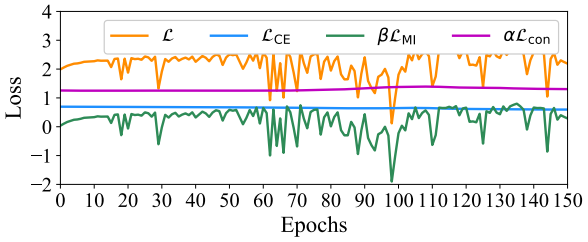
Figure 4: Original graph and IB-Graphs with different β when VIB-GSL achieves the same testing performance.

nal graph and IB-Graphs learned by VIB-GSL in Figure 4, where $|E|$ indicates the number of edges. To further analyze the impact of information compression degree, we visualize the learned IB-Graph with different β when VIB-GSL achieves the same testing performance. Note that VIB-GSL does not set sparsity constraint as in most structure learning methods. As shown in Figure 4, we make the following observations: (1)VIB-GSL tends to generate edges that connect nodes playing the same structure roles, which is consistent with the homophily assumption. (2)When achieving the same testing performance, VIB-GSL with larger β will generate a more dense graph structure. It is because with the degree of information compression increasing, the nodes need more neighbors to obtain enough information.

Training Stability. As mentioned in Section 3.3, VIB-GSL deduces a tractable variational approximation for the IB objective, which facilitates the training stability. In this subsection, we analyze the convergence of VIB-GSL and SIB (Yu et al. 2020) on REDDIT-B with a learning rate of 0.001. The IB objective in (Yu et al. 2020) is $\mathcal{L} = \mathcal{L}_{\text{CE}} + \beta\mathcal{L}_{\text{MI}} + \alpha\mathcal{L}_{con}$, where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss, $\mathcal{L}_{\text{MI}}$ is the MINE loss of estimating mutual information between original graph and learned subgraph and $\mathcal{L}_{con}$ is a connectivity regularizer. Figure 5(a) depicts the losses of VIB-GSL (i.e., overall loss $\mathcal{L}$, cross-entropy loss $\mathcal{L}_{\text{CE}}$ for classification, and the KL-divergence loss $\mathcal{D}_{\text{KL}}$) with $\beta = 10^{-3}$, where the dash lines indicates the mean value in the last 10 epochs when VIB-GSL converges. As mentioned in Section 3.3, SIB adopted a bi-level optimization scheme for IB objective. Figure 5(b) depicts the losses of SIB (i.e., overall loss $\mathcal{L}$, classification loss $\mathcal{L}_{\text{CE}}$, the MI estimation loss $\mathcal{L}_{\text{MI}}$, and the connectivity loss $\mathcal{L}_{con}$) with $\beta = 0.2$ and $\alpha = 5$ as suggested in its source code. As shown in Figure 5(a), VIB-GSL converge steadily, showing the effectiveness of the variational approximation. As shown in Figure 5(b), the MI estimation loss $\mathcal{L}_{\text{MI}}$ is very unstable because of the bi-level optimization scheme, making SIB is very difficult to converge.



(a) VIB-GSL.



(b) SIB.

Figure 5: Training dynamics of VIB-GSL and SIB.

5 Conclusion

In this paper, we advance the Information Bottleneck principle for graph structure learning and propose a framework named VIB-GSL, which jointly optimizes the graph structure and graph representations. VIB-GSL deduces a variational approximation to form a tractable IB objective function that facilitates training stability and efficiency. We evaluate the proposed VIB-GSL in graph classification and graph denoising. Experimental results verify the superior effectiveness and robustness of VIB-GSL.

Technical Appendix

A. Proofs

Proof of Lemma 1

We first provide the proof of Lemma 1 in Section 3.1.

Proof. We prove the Lemma 1 following the same strategy of Proposition 3.1 in (Achille and Soatto 2018). Suppose G is defined by Y and G_n, and G_{IB} depends on G_n only through G. We can define the Markov Chain $< (Y, G_n) \rightarrow G \rightarrow G_{\text{IB}} >$. According to the data processing inequality (DPI), we have:

$$\begin{aligned} I(G_{\text{IB}}; G) &\geq I(G_{\text{IB}}; Y, G_n) \\ &= I(G_{\text{IB}}; G_n) + I(G_{\text{IB}}; Y|G_n) \\ &= I(G_{\text{IB}}; G_n) + H(Y|G_n) - H(Y|G_n; G_{\text{IB}}). \end{aligned} \tag{18}$$

Since G_n is be a task-irrelevant nuisance, it is independent with Y, we have $H(Y|G_n) = H(Y)$ and $H(Y|G_n; G_{\text{IB}}) \leq H(Y|G_{\text{IB}})$. Then

$$\begin{aligned} I(G_{\text{IB}}; G) &\geq I(G_{\text{IB}}; G_n) + H(Y|G_n) - H(Y|G_n; G_{\text{IB}}) \\ &\geq I(G_{\text{IB}}; G_n) + H(Y) - H(Y|G_{\text{IB}}) \\ &= I(G_{\text{IB}}; G_n) + I(G_{\text{IB}}; Y). \end{aligned} \tag{19}$$

Thus we obtain $I(G_{\text{IB}}; G_n) \leq I(G_{\text{IB}}; G) - I(G_{\text{IB}}; Y)$. □

Proof of Proposition 1

Then we provide the proof of Proposition 1 in Section 3.1.

Proof. According to the definition of mutual information,

$$\begin{aligned} -I(Y, G_{\text{IB}}) &= -\iint p(Y, G_{\text{IB}}) \log \frac{p(Y, G_{\text{IB}})}{p(Y)p(G_{\text{IB}})} dY\, dG_{\text{IB}} \\ &= -\iint p(Y, G_{\text{IB}}) \log \frac{p(Y|G_{\text{IB}})}{p(Y)} dY\, dG_{\text{IB}}, \end{aligned} \tag{20}$$

where $p(Y|G_{\text{IB}})$ can be fully defined by the Markov Chain $< (Y, G_n) \rightarrow G \rightarrow G_{\text{IB}} >$ as $p(Y|G_{\text{IB}}) = \int p(Y|G_{\text{IB}}) p(G_{\text{IB}}|G) dG$. Since $p(Y|G_{\text{IB}})$ is intractable, let $q_\theta(Y|G_{\text{IB}})$ be the variational approximation of the true posterior $p(Y|G_{\text{IB}})$. According to the non-negativity of Kullback Leiber divergence:

$$\begin{aligned} &\mathcal{D}_{\text{KL}}(p(Y|G_{\text{IB}})||q_\theta(Y|G_{\text{IB}})) \geq 0 \Longrightarrow \\ &\int p(Y|G_{\text{IB}}) \log p(Y|G_{\text{IB}}) dY \\ &\geq \int p(Y|G_{\text{IB}}) \log q_\theta(Y|G_{\text{IB}}) dY. \end{aligned} \tag{21}$$

Plug Eq. (20) into Eq. (21), then we have

$$\begin{aligned} -I(Y, G_{\text{IB}}) &\leq -\iint p(Y, G_{\text{IB}}) \log \frac{q_\theta(Y|G_{\text{IB}})}{p(Y)} dY\, dG_{\text{IB}} \\ &= -\iint p(Y, G_{\text{IB}}) \log q_\theta(Y|G_{\text{IB}}) dY\, dG_{\text{IB}} \\ &\quad + H(Y), \end{aligned} \tag{22}$$

where $H(Y)$ is the entropy of label Y, which can be ignored in optimization procedure. □

Proof of Proposition 2

We next provide the proof of Proposition 2 in Section 3.1.

Proof. According to the definition of mutual information,

$$I(G_{\text{IB}}, G) = \iint p(G_{\text{IB}}, G) \log \frac{p(G_{\text{IB}}|G)}{p(G_{\text{IB}})} dG_{\text{IB}} dG. \tag{23}$$

In general, computing the distribution $p(G_{\text{IB}}) = \int p(G_{\text{IB}}|G) p(G) dG$ is very difficult, so we use $r(G_{\text{IB}})$ as the variational approximation to $p(G_{\text{IB}})$. Since the Kullback Leiber divergence $\mathcal{D}_{\text{KL}}(p(Z)||r(Z)) \geq 0$,

$$\begin{aligned} &\mathcal{D}_{\text{KL}}(p(Z)||r(Z)) \geq 0 \Longrightarrow \\ &\int p(z) \log p(z) dz \geq \int p(z) \log r(z) dz. \end{aligned} \tag{24}$$

Plug Eq. (23) into Eq. (24), then we have

$$I(G_{\text{IB}}, G) \leq \iint p(G_{\text{IB}}, G) \log \frac{p(G_{\text{IB}}|G)}{r(G_{\text{IB}})} dG_{\text{IB}} dG. \tag{25}$$

□

B. Training Efficiency

For VIB-GSL, the cost of learning an IB-Graph is $\mathcal{O}(nd + n^2 d)$ for a graph with n nodes in $\mathbf{R}^d$, while computing graph representation costs $\mathcal{O}(n^2 d + ndK)$, where d is the node feature dimension and K is the bottleneck size. If we assume that $d \approx K$ and $d \ll n$, the overall time complexity is $\mathcal{O}(Kn^2)$.

We compare the training efficiency of VIB-GSL with other baselines and show the mean training time of one epoch in seconds (10 runs) in Figure 6. For Subgraph-IB, we set the inner loop iterations as 10. For IDGL, we set the maximal number of iterations in the dynamic stopping strategy to 10 as suggested in its source code. As shown in Figure 6, VIB-GSL shows comparable efficiency with other methods when achieving the best performance.

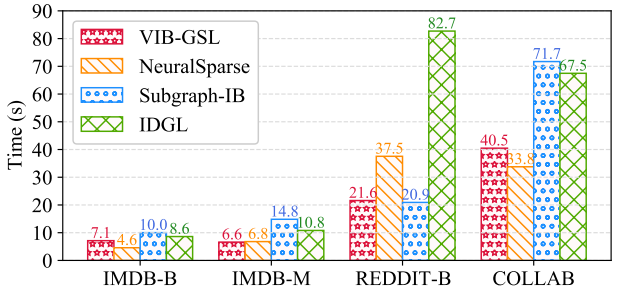


Figure 6: Training time of one epoch on various datasets.

Acknowledgments

The corresponding author is Jianxin Li. The authors of this paper are supported by the (No.U20B2053 and 61872022), State Key Laboratory of Software Development Environment (SKLSDE-2020ZX-12), Outstanding Research Project of Shen Yuan Honors College, BUAA, (230121208), the ARC DECRA Project (No. DE200100964), and in part by NSF under grants III-1763325, III-1909323, III-2106758, and SaTC-1930941.

References

Achille, A.; and Soatto, S. 2018. Emergence of invariance and disentanglement in deep representations. *The Journal of Machine Learning Research*, 19(1): 1947–1980.

Alemi, A. A.; Fischer, I.; Dillon, J. V.; and Murphy, K. 2016. Deep variational information bottleneck. In *ICLR*.

Bao, F. 2021. Disentangled Variational Information Bottleneck for Multiview Representation Learning. *arXiv preprint arXiv:2105.07599*.

Belghazi, M. I.; Baratin, A.; Rajeshwar, S.; Ozair, S.; Bengio, Y.; Courville, A.; and Hjelm, D. 2018. Mutual information neural estimation. In *ICML*, 531–540.

Chen, Y.; Wu, L.; and Zaki, M. 2020. Iterative deep graph learning for graph neural networks: Better and robust node embeddings. In *NeurIPS*.

Cover, T. M. 1999. *Elements of information theory*. John Wiley & Sons.

Dubois, Y.; Kiela, D.; Schwab, D. J.; and Vedantam, R. 2020. Learning optimal representations with the decodable information bottleneck. In *NeurIPS*.

Franceschi, L.; Niepert, M.; Pontil, M.; and He, X. 2019. Learning discrete structures for graph neural networks. In *ICML*, 1972–1982.

Gao, C.; Chen, J.; Liu, S.; Wang, L.; Zhang, Q.; and Wu, Q. 2021. Room-and-object aware knowledge reasoning for remote embodied referring expression. In *CVPR*, 3064–3073.

Hershey, J. R.; and Olsen, P. A. 2007. Approximating the Kullback Leibler divergence between Gaussian mixture models. In *ICASSP*, IV–317.

Jang, E.; Gu, S.; and Poole, B. 2017. Categorical reparameterization with gumbel-softmax. In *ICLR*.

Jeon, I.; Lee, W.; Pyeon, M.; and Kim, G. 2021. IB-GAN: Disengangled Representation Learning with Information Bottleneck Generative Adversarial Networks. In *AAAI*, 7926–7934.

Kim, J.; Kim, M.; Woo, D.; and Kim, G. 2021. Drop-Bottleneck: Learning Discrete Compressed Representation for Noise-Robust Exploration. In *ICLR*.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. In *ICLR*.

Kipf, T. N.; and Welling, M. 2016. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

Li, Q.; Peng, H.; Li, J.; Xia, C.; Yang, R.; Sun, L.; Yu, P. S.; and He, L. 2020. A survey on text classification: From shallow to deep learning. *ACM Transactions on Intelligent Systems and Technology*.

Mahabadi, R. K.; Belinkov, Y.; and Henderson, J. 2021. Variational Information Bottleneck for Effective Low-Resource Fine-Tuning. In *ICLR*.

Pan, Z.; Niu, L.; Zhang, J.; and Zhang, L. 2020. Disentangled Information Bottleneck. *arXiv preprint arXiv:2012.07372*.

Peng, H.; Zhang, R.; Dou, Y.; Yang, R.; Zhang, J.; and Yu, P. S. 2021. Reinforced Neighborhood Selection Guided Multi-Relational Graph Neural Networks. *ACM Transactions on Information Systems*.

Rossi, R.; and Ahmed, N. 2015. The network data repository with interactive graph analytics and visualization. In *AAAI*, 4292–4293.

Saxe, A. M.; Bansal, Y.; Dapello, J.; Advani, M.; Kolchinsky, A.; Tracey, B. D.; and Cox, D. D. 2019. On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12): 124020.

Shamir, O.; Sabato, S.; and Tishby, N. 2010. Learning and generalization with the information bottleneck. *Theoretical Computer Science*, 411(29-30): 2696–2711.

Shapiro, A. 2003. Monte Carlo sampling methods. *Handbooks in operations research and management science*, 10: 353–425.

Sinha, S.; Bharadhwaj, H.; Goyal, A.; Larochelle, H.; Garg, A.; and Shkurti, F. 2020. Diversity inducing Information Bottleneck in Model Ensembles. In *AAAI*, 9666–9674.

Sun, L.; Dou, Y.; Yang, C.; Wang, J.; Yu, P. S.; He, L.; and Li, B. 2018. Adversarial attack and defense on graph data: A survey. *arXiv preprint arXiv:1812.10528*.

Sun, Q.; Li, J.; Peng, H.; Wu, J.; Ning, Y.; Yu, P. S.; and He, L. 2021. SUGAR: Subgraph neural network with reinforcement pooling and self-supervised mutual information mechanism. In *Web Conference*, 2081–2091.

Tian, Y.; Sun, C.; Poole, B.; Krishnan, D.; Schmid, C.; and Isola, P. 2020. What makes for good views for contrastive learning? In *NeurIPS*.

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

Tong, Z.; Liang, Y.; Ding, H.; Dai, Y.; Li, X.; and Wang, C. 2021. Directed Graph Contrastive Learning. *Advances in Neural Information Processing Systems*, 34.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph Attention Networks. In *ICLR*.

Wu, T.; Ren, H.; Li, P.; and Leskovec, J. 2020. Graph information bottleneck. In *NeurIPS*.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *ICLR*.

Yang, L.; Wu, F.; Zheng, Z.; Niu, B.; Gu, J.; Wang, C.; Cao, X.; and Guo, Y. 2021. Heterogeneous Graph Information Bottleneck. In *IJCAI*, 1638–1645.

Yu, J.; Xu, T.; Rong, Y.; Bian, Y.; Huang, J.; and He, R. 2020. Graph Information Bottleneck for Subgraph Recognition. In *ICLR*.

Yu, J.; Xu, T.; Rong, Y.; Bian, Y.; Huang, J.; and He, R. 2021. Recognizing Predictive Substructures with Subgraph Information Bottleneck. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhang, D.; Yin, J.; Zhu, X.; and Zhang, C. 2018. Network representation learning: A survey. *IEEE transactions on Big Data*, 6(1): 3–28.

Zheng, C.; Zong, B.; Cheng, W.; Song, D.; Ni, J.; Yu, W.; Chen, H.; and Wang, W. 2020. Robust graph representation learning via neural sparsification. In *ICML*, 11458–11468.

Zhmoginov, A.; Fischer, I.; and Sandler, M. 2020. Information-bottleneck approach to salient region discovery. In *ECML/PKDD*, 531–546.

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

Zhu, Y.; Xu, W.; Zhang, J.; Liu, Q.; Wu, S.; and Wang, L. 2021. Deep Graph Structure Learning for Robust Representations: A Survey. *arXiv preprint arXiv:2103.03036*.

Zügner, D.; Akbarnejad, A.; and Günnemann, S. 2018. Adversarial attacks on neural networks for graph data. In *ACM SIGKDD*, 2847–2856.