Graph Contrastive Learning with Cross-view Reconstruction

Anonymous Author(s)
Affiliation
Address
email

Abstract

Although different graph self-supervised learning strategies have been proposed to tackle the supervision shortage issue in graph learning tasks, Graph contrastive learning (GCL) has been the most prevalent approach to this problem. Despite the remarkable performances those GCL methods have achieved, existing GCL methods that heavily depend on various manually designed augmentation techniques still struggle to improve model robustness without risking losing task-relevant information. Consequently, the learned representation is either brittle or unilluminating. In light of this, we introduce the GraphCV, which follows the information bottleneck principle to learn minimal yet sufficient representations from graph data. Specifically, our proposed model elicits the predictive (useful for downstream instance discrimination) and other non-predictive features separately. Except for the conventional contrastive loss which guarantees the consistency and sufficiency of the representations across different augmentation views, we introduce a cross-view reconstruction mechanism to pursue the disentanglement of the two learned representations. Besides, an adversarial global view is added as the third view of contrastive loss to avoid the learned representation from being drafted too far away from the original distribution. We empirically demonstrate that our proposed model outperforms the state-of-the-art on graph classification task over multiple benchmark datasets.

1 Introduction

Graph representation learning (GRL) has attracted significant attention due to its widespread applications in the real-world interaction systems, such as social, molecules, biological and citation networks [11]. The current state-of-the-art supervised GRL methods are mostly based on Graph Neural Networks (GNNs) [17, 35, 10, 41], which require a large amount of task-specific supervised information. Despite the remarkable performances, they are usually limited by the deficiency of label supervision in real-world graph data due to the fact that it is usually easy to collect unlabeled graph but could be very costly to obtain enough annotated label, especially in certain fields, like biochemistry. Therefore, many recent works [26, 12, 30] have studied how to fully utilize the unlabeled information on graph and thus stimulating the application of self-supervised learning (SSL) for GRL where only limited or even no label is needed.

As a prevalent and effective strategy of SSL, contrastive learning follows the mutual information maximization principle (InfoMax) [36] to maximize the agreements of the positive pairs while minimizing that of the negative pairs in embedding space. However, the graph contrastive learning paradigm guided with this InfoMax principle has been theoretically and empirically proved to be insufficient to learn the discriminative and robust representation. To be more specific, state-of-the-art GCL methods [26, 12, 44] usually implement specific perturbation $A(\cdot)$ (e.g., Subgraph Sampling, Node Dropping, Edge Removing and Attributes Masking) on the anchor graph G to generate its

Submitted to NeurIPS 2022GLFrontiers Workshop. Do not distribute.

positive pair $A(G)$ and then train the graph feature encoder to ensure the representation consistency within the positive pair, i.e., $f(G) = f(A(G))$. Consequently, such learning strategy is heavily dependent on the design of graph augmentation techniques to guarantee the robustness of the learned representation. Mild graph augmentation could push encoders to capture redundant and biased information [33], which could inadvertently suppress the important predictive features and negatively affect the downstream task via the so-called "shortcut" solution [27]. On the other hand, too aggressive augmentation may easily lead to another extreme where many predictive features are radomly dropped and the learned representations are not informative enough to identify different graphs. Although recent works [31, 43, 42, 19] try to implement learnable augmentations on the anchor graph to extract the most salient features (those are most easy to learn and resistant to augmentation) to improve representation robustness, they still suffer from the difficulty in controlling the augmentation extent to balance the representation sufficiency and robustness without the guidance of explicit domain knowledge, thus usually lead to unsatisfying performance. To mitigate this issue, a method which can disentangle the predictive and the non-predictive latent factors without sacrificing the sufficiency of the original predictive graph features is in urgent need.

In this paper, we address this challenge by proposing a novel graph contrastive learning model with cross-view reconstruction, named GraphCV. Specifically, GraphCV consists of a graph encoder followed with two decoders that are trained to extract information particular to the predictive and traivial latent factors, respectively. To achieve this optimization objective, we propose a reconstruction-based representation learning, including intra-view and inter-view reconstructions, to reconstruct the original learned representation with corresponding learned predictive relevant and irrelevant representations. Meanwhile, a adversarial graph perturbed from original view is added as the third view of the contrastive loss besides the predictive relevant representations of the two augmented graph views to ensure the predictive relevant representation maintaining the global semantics instead of the partial or even trivial features. We provide theoretical analysis to show that GraphCV is capable to learn a minimal sufficient representation with the designs above. Finally, we conduct experiments to validate the effectiveness of GraphCV, on the commonly-used graph benchmark datasets. The experiment results show that GraphCV achieves significant performance gains over different datasets and settings compared with state-of-the-art baselines.

To sum up, our main contributions of this work include three aspects: (i) We propose GraphCV to learn the disentangled and debiased representation with a cross-view reconstruction mechanism; (ii) We provide solid theoretical analysis to analysis the rationality of our designs (iii) We conduct thorough experiments to demonstrate that GraphCV can mitigate the "shortcut" solution in contrastive learning and significantly outperforms the state-of-the-art baselines over multiple graph classification benchmark datasets.

2 Preliminaries

2.1 Graph Representation Learning

In this work, we focus on the graph-level task, let $\mathcal{G} = \{G_i = (V_i, E_i)\}_{i=1}^N$ denote a graph dataset with N graphs, where V_i and E_i are the node set and edge set of graph G_i, respectively. We use $x_v \in \mathbb{R}^d$ and $x_e \in \mathbb{R}^d$ to denote the attribute vector of each node $v \in V_i$ and edge $e \in E_i$. Each graph is associated with a label, denoted as y_i, the goal the graph representation learning is to learn an encoder $f : G_i \to \mathbb{R}^d$ so that the learned representation $\mathbf{z}_i = f(G_i)$ is sufficient to identify y_i in the downstream task. We clarify sufficiency as $\mathbf{z}_i$ containing the same amount of information as G_i for label identification [1], and it is formulated as:

$$I(G_i; y_i \mid \mathbf{z}_i) = 0, \tag{1}$$

where $I(;)$ denotes the mutual information between two variables. We demonstrate the general optimization result of classical representation learning in Figure 1(a).

2.2 Contrastive Learning

Contrastive Learning (CL) is a self-supervised representation learning method which leverages instance-level identity as supervision. During the training phase, each instance x firstly goes through proper data transformation to generate two data augmentation views $t_1(x)$ and $t_2(x)$, where $t_1(\cdot)$ and

$t_2(\cdot)$ are transformation functions. Then, the CL method learns a encoder f (a backbone network plus a projection layer) which maps $t_1(x)$ and $t_2(x)$ closer in the hidden space so that the learned representations $\mathbf{z}_1$ and $\mathbf{z}_2$ maintain all the information shared by $t_1(x)$ and $t_2(x)$. The encoder is usually optimized by contrastive loss, such as NCE loss [38], InfoNCE loss [34] and NT-Xent loss [5]. We hereby provide the general optimization result of CL in Figure 1(b). In Graph Contrastive Learning (GCL), we usually use GNNs, like GCN [17] and GIN [41], as the backbone networks and the commonly-used graph data augmentation operators [44], such as node dropping, edge perturbation, subgraph sampling, and attribute masking.

All the CL-based methods are built on an assumption that augmentations do not change the information regarding to the label. Here, we follow [7] to clear up the definition of redundancy. $t_1(G)$ is redundant to $t_2(G)$ for y iff $t_1(G)$ and $t_2(G)$ share the same label-relevant information. In CL, $t_1(G)$ and $t_2(G)$ are supposed to be mutually redundant, we define the mutual redundancy as:

$$I(t_1(G); y \mid t_2(G)) = I(t_2(G); y \mid t_1(G)) = 0. \tag{2}$$

2.3 Adversarial Training

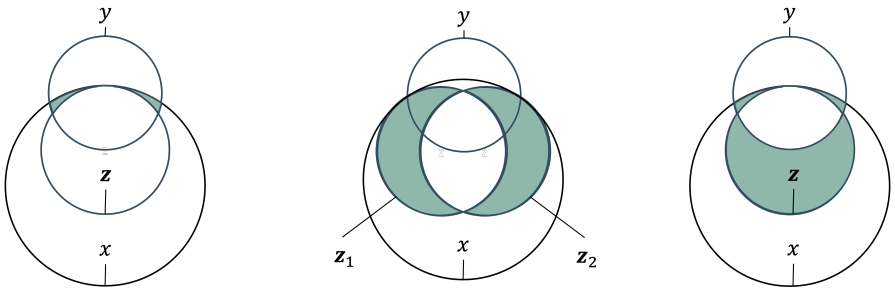
Deep neural networks have been demonstrated to be vulnerable to adversarial attacks [8]. Among all approaches proposed against adversarial attacks, Adversarial Training (AT) achieves remarkable robustness. Specifically, AT introduces a perturbation variable δ to improve the model robustness by training the network on the adversarial samples $x + \delta$. During the training phase, the model is optimized to minimize the training loss, and the δ is optimized within the radius ϵ to maximize the loss. The supervised setting of AT is defined as:

$$\min_{\theta} \mathbb{E}_{(x,y)\in D} \max_{\|\delta\|_\infty \leqslant \epsilon} \mathcal{L}(x + \delta, y; \theta), \tag{3}$$

where (x, y) are the data feature and label sampled from training set D respectively, and $\mathcal{L}$ denotes the supervised training objective, such as the cross-entropy loss. Except improving model robustness to adversarial attacks, AT is also capable of reducing overfitting and further increasing the generalization performance [39, 47]. One possible reason behind this phenomena is that AT follows the Information Bottleneck principle [32, 24], in which the optimal representations only contain minimal yet sufficient information. Depending on whether relevant to label or not, the mutual information between representation $\mathbf{z}$ and x can be decomposed into two parts:

$$I(x; \mathbf{z}) = \underbrace{I(y; \mathbf{z})}_{\text{predictive information}} + \underbrace{I(x; \mathbf{z} \mid y)}_{\text{non-predictive information}}. \tag{4}$$

Ideally, the learned representation $\mathbf{z}$ through AT is expected to keep all the predictive information from x intact while ignoring other non-predictive (traivial) information, we plot this ideal situation in Figure 1(c).



(a) Representation Learning (b) Contrastive Learning (c) Adversarial Training

Figure 1: Illustration of the relation between feature x, label y and representation $\mathbf{z}$ in terms of information entropy under the three scenarios above. (a) The ideal optimization result of classical graph representation learning, the green area becomes null when Equation 1 is satisfied. (b) The green area is optimized to null in CL, and $\mathbf{z}_1$ and $\mathbf{z}_2$ share the same intersection with y when the assumption in Equation 2 holds. (c) The green area becomes null when AT achieves its optimal optimization results, in which the second term in Equation 4 is minimized to 0.

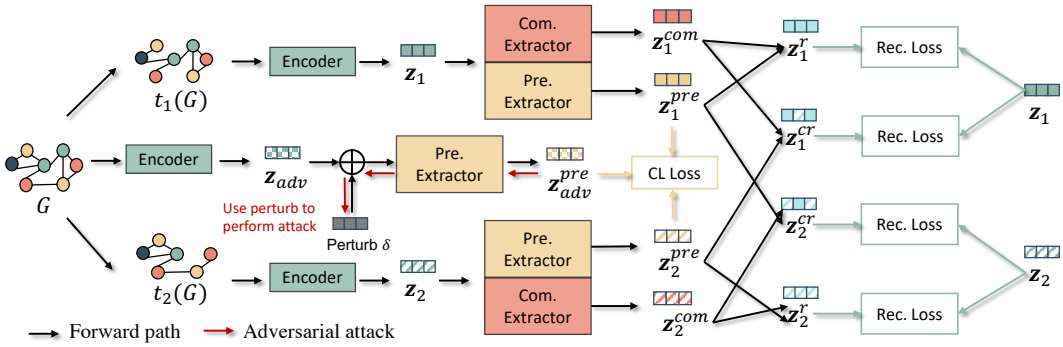


Figure 2: The illustration of proposed GraphCV. (1) Graph augmentations are applied to the input graph G to produce two augmented graphs, which are then fed into the shared graph encoder $f(\cdot)$ to generate two graph representations $\mathbf{z}_1$ and $\mathbf{z}_2$. (2) $\mathbf{z}_1$ and $\mathbf{z}_2$ are used as the inputs of the two decoder to generate two pairs of graph representations, $\mathbf{z}^{pre}$ captures the predictive factors and $\mathbf{z}^{com}$ keep other complementary non-predictive features. Then we use the two pairs of representations to reconstruct $\mathbf{z}_1$ and $\mathbf{z}_2$ in both of the intra-view and inter-view. (3) An adversarial sample generated by G will go through the same procedure to generate $\mathbf{z}_{adv}^{pre}$. We take it as the third view besides $\mathbf{z}_1^{pre}$ and $\mathbf{z}_2^{pre}$ in CL guarantee the $\mathbf{z}^{pre}$ can keep the global semantics.

3 Proposed Model

In this section, we introduce the proposed GraphCV, and its framework is shown in Figure 2. Before we dive into the details of GraphCV, we will briefly analyze the sufficiency and robustness in GCL, and provide illustration of the disentanglement hypothesis.

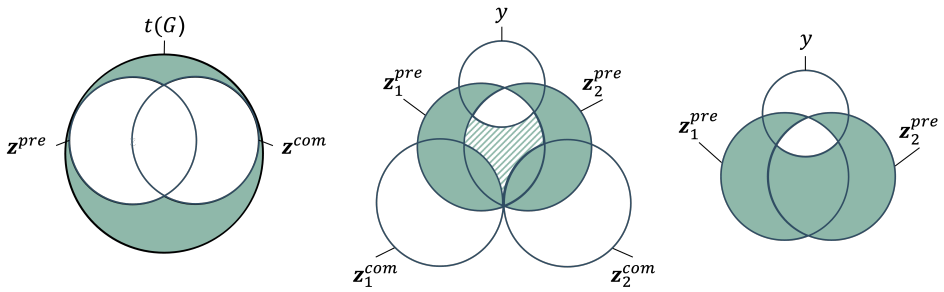
3.1 Motivation of GraphCV

To minimize the redundant information in graph representation, the principle of information bottleneck (IB) has been introduced in graph representation learning [45], and the learned model is empirically proved to be more robust to adversarial attacks. In the circumstances of graph contrastive learning (GCL), labels are not accessible to guide the optimization process, thus it is more challenging to discern the predictive information and redundant information. For each graph $G \in \mathcal{G}$, GCL methods generally pick two augmentation operators $t_1(\cdot)$ and $t_2(\cdot)$ IID sampled from the same family of augmentation $\mathcal{T}$ to generate two augmented graph $t_1(G)$ and $t_2(G)$ for contradistinction. However, as we stated in Section 1, strong augmentation will randomly drop a large part of graph features and cause overwhelming distribution shift between the two augmented views. As a result, the two augmented graph views may fail to hold the mutual redundant assumption in Section 2.2 and thereby he learned representation will not satisfy the sufficiency requirement in Section 2.1.

To address this dilemma, we propose the GraphCV with reconstruction mechanism to improve its robustness without rely on carefully balancing the representation sufficiency and robustness or training an extra augmentation module. Given an augmented graph $t(G)$, we aim to learn a air of disentangled representation $\mathbf{z} = (\mathbf{z}^{pre}, \mathbf{z}^{com})$, where $\mathbf{z}^{pred} \in \mathbb{R}^d$ is expected to be specific to the predictive information, while $\mathbf{z}^{com} \in \mathbb{R}^d$ is optimized to elicit the complementary non-predictive factors from $t(G)$, i.e., $\mathbf{z}^{pre} \sim G$. We provide the illustration of the optimal representation disentanglement in Figure 3(a) and (b): (1) $\mathbf{z}^{pre}$ is sufficient for its corresponding graph view $t(G)$ regarding to y, and the union of the $\mathbf{z}^{com}$ and $\mathbf{z}^{pre}$ cover all the information in $t(G)$; (2) $\mathbf{z}^{pre}$ and $\mathbf{z}^{com}$ are optimized toward mutually exclusion (disentangled), and $\mathbf{z}^{pre}$ is trained to capture all the shared information between the two augmentation views, thus maintains all the information for label identification (under the assumption that implemented mild augmentation does not cause predictive features loss). To approximate to this optimal point, we need to propose corresponding designs to guarantee the sufficiency and disentanglement of the representation. Next, we will introduce our framework details to explain how do we achieve the two restrictions, respectively.

3.2 Disentanglement by Cross-View Reconstruction

In GCL, we usually leverage a graph encoder to aggregate the feature of graph data as its representation. There are multiple choices of graph encoders in GCL, including GCN [17] and GIN [41], etc. In this work, we adopt GIN as the backbone network f for simplicity. Note that any other commonly-used graph encoders can also be applied to our model. Given two augmentation



(a) Sufficiency (b) Disentanglement (c) Robustness

Figure 3: Illustration of the relation between augmented graph $t(G)$, label y, predictive representation $\mathbf{z}^{pre}$ and non-predictive representation $\mathbf{z}^{com}$ in terms of information entropy under the optimal situation. The green areas in the three figure become null when the optimal situation is achieved. (a) The union of $\mathbf{z}^{pre}$ and $\mathbf{z}^{com}$ covers all the information of its corresponding augmented view $t(G)$. (b) $\mathbf{z}^{pre}$ and $\mathbf{z}^{com}$ are optimized to approximate mutually exclusion. $\mathbf{z}_1^{pre}$ and $\mathbf{z}_2^{pre}$ are optimized to be mutually redundant, including the predictive (white) and part of redundant (shadow) information. $\mathbf{z}_1^{com}$ and $\mathbf{z}_1^{com}$ is responsible to capture all the complementary non-predictive information of the two augmented views, where non-overlapping can exist in the two representations since the two randomly sampled augmentation operators can cause the distribution shift between the two views while $\mathbf{z}^{pre}$ only extract the shared part of the two views. (c) $\mathbf{z}_1^{pre}$ and $\mathbf{z}_2^{pre}$ are optimized to more robust, and the redundant information left within them is further minimized.

views $t_1(G)$ and $t_1(G)$, we firstly use the encoder $f(\cdot)$ to map the the them into a lower dimension hidden space to generate two embeddings $\mathbf{z}_1$ and $\mathbf{z}_2$. Instead of directly maximizing the agreement between the two entangled representations $\mathbf{z}_1$ and $\mathbf{z}_2$, we further feed them into a pair decoders $g = (g_{pre}, g_{com})$ (both of them are MLP-based networks) to extract the disentangled predictive and non-predictive embeddings:

$$\left[\mathbf{z}^{pre} = g_{pre}(f(t(G))),\ \mathbf{z}^{com} = g_{com}(f(t(G)))\right], \tag{5}$$

where we can generate a pair of disentangled embeddings for both $t_1(G)$ and $t_2(G)$ through the procedure above. Ideally, the mutual redundancy assumption between $(\mathbf{z}_1^{pre}, \mathbf{z}_2^{pre})$ can thus be guaranteed because $t_1(G)$ and $t_1(G)$ are augmented from the same original graph, and they naturally share the same predictive factors, unless aggressive augmentations are implemented. Here, we clarify the lower bound of the mutual information between one augmentation view and the learned predictive representation of another augmentation view in Theorem 1.

Theorem 1 *Suppose $f(\cdot)$ is a GNN encoder as powerful as 1-WL test. Let $g_{pre}(\cdot)$ elicits only the predictive information from $\mathbf{z}$ meanwhile $g_{com}(\cdot)$ extracts the non-predictive factors of G from $\mathbf{z}_1$ and $\mathbf{z}_2$. Then we have:*

$$I\left(t_1(G); \mathbf{z}_2^{pre}, \mathbf{z}_2^{com}\right) \geqslant I\left(\mathbf{z}_1^{pre}; \mathbf{z}_2^{pre}\right) \text{ where } G \in \mathcal{G} \text{ and } t_1(\cdot), t_2(\cdot) \in \mathcal{T}.$$

The detailed proof is provided in Section C of Appendix. Therefore, we can maximize the consistency between the representations of the two views by maximizing the mutual information of between $\mathbf{z}_1^{pre}$ and $\mathbf{z}_2^{pre}$. Therefore, we can derive our objective to ensure the view invariance as follow:

$$\mathcal{L}_{\text{pred}} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_{\text{CL}}(\mathbf{z}_{1,i}^{pre}, \mathbf{z}_{2,i}^{pre}), \tag{6}$$

where $\mathcal{L}_{\text{CL}}(\cdot)$ denotes the contrastive objective and we adopt InfoNCE loss in this work [34]. Meanwhile, to pursue the feature sufficiency and disentanglement as stated above, we thus propose to use the cross-view reconstruction mechanism to approximate these two objectives. To be specific, we will use the representation pair $(\mathbf{z}^{pre}, \mathbf{z}^{com})$ within and cross the augmentation views to recover the original raw data so that the two objectives can be guaranteed simultaneously. Due to the reason that graph data is a kind of non-Euclidean structured data which can not be represented in the euclidean space like the raw data in computer vision domain, we turn to infer the output of $\mathbf{z} = f(t(G))$ based on $(\mathbf{z}^{com}, \mathbf{z}^{pre})$. Firstly, we do the reconstruction within the augmentation view, namely mapping $(\mathbf{z}_w^{pre}, \mathbf{z}_w^{com})$ to $\mathbf{z}_w$, where $w \in \{1, 2\}$ representing the augmentation view. The optimal result of the this step is shown in Figure 3(a), where the joint of $\mathbf{z}_w^{pre}$ and $\mathbf{z}_w^{com}$ can cover all the information in its corresponding augmentation graph view w. Since only $\mathbf{z}^{pre}$ is involved in the contrastive

loss in Equation 6, the graph encoder f is thus not optimized to focus only on the easy-learned salient features and less powerful than 1-WL test. Then, we define the $(\mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com})$ as a cross-view representation pair and the reconstruction procedure will be repeated on it to predict $\mathbf{z}_{w'}$, aiming to ensure $\mathbf{z}^{pre}$ and $\mathbf{z}^{com}$ is optimized to approximate mutual disentanglement, where $w = 1, w' = 2$ or $w = 2, w' = 1$. The disentanglement optimization process is illustrated in Figure 3(b), where $I(\mathbf{z}^{pre}; \mathbf{z}^{com}) \to 0$ in the ideal situation. Here, we formulate the reconstruction procedures as:

$$\mathbf{z}_w^r = g_{rec}\left(\mathbf{z}_w^{pre} \odot \mathbf{z}_w^{com}\right), \ \mathbf{z}_w^{cr} = g_{rec}\left(\mathbf{z}_w^{pre} \odot \mathbf{z}_{w'}^{com}\right), \tag{7}$$

where g_{rec} is the parameterized reconstruction model and $\odot$ is the pre-defined fusion operator, like element-wise product or concatenation. The reconstruction procedures are optimized by minimizing the entropy $H\left(\mathbf{z}_w \mid \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right)$, where $w = w'$ or $w \neq w'$. Ideally, we can reach the optimal situation demonstrated in Figure 3(a) and (b) iff $H\left(\mathbf{z}_w \mid \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right) = -\mathbb{E}_{p\left(\mathbf{z}_w, \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right)}\left[\log p\left(\mathbf{z}_w \mid \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right)\right] = 0$, where $\mathbf{z}_w$ is exactly recovered given its non-predictive representation and the predictive representation of any view. Nevertheless, the condition probability $p\left(\mathbf{z}_w \mid \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right)$ is unknown for us, we hence use the variation distribution approximated by g_{rec} instead, denoted as $q\left(\mathbf{z}_w \mid \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right)$. We provide the upper bound of $H\left(\mathbf{z}_w \mid \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right)$ in Theorem 2.

Theorem 2 *Assume q is a Gaussian distribution, g_r is the parameterized reconstruction model which infer $\mathbf{z}_w$ from $(\mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com})$. Then we have:*

$$H\left(\mathbf{z}_w \mid \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right) \leqslant \left\|\mathbf{z}_w - g_r\left(\mathbf{z}_w^{pre} \odot \mathbf{z}_{w'}^{com}\right)\right\|_2^2 \text{ where } w = w' \text{ or } w \neq w'.$$

The detailed proof is demonstrated in Section C of Appendix. Since we adopt two augmentation views, the objective function constraining representation sufficiency and disentanglement can be formulated as:

$$\mathcal{L}_{\text{recon}} = \frac{1}{2N} \sum_{i=1}^{N} \sum_{w=1}^{2} \left[\left\|\mathbf{z}_{w,i} - \mathbf{z}_{w,i}^r\right\|_2^2 + \left\|\mathbf{z}_{w,i} - \mathbf{z}_{w,i}^{cr}\right\|_2^2\right]. \tag{8}$$

3.3 Adversarial Contrastive View

With the cross-view reconstruction mechanism above, the two learned representations stated above are optimized towards the disentangled manner. However, we still need to ensure $\mathbf{z}^{pre}$ is more relevant to predictive information than $\mathbf{z}^{com}$. Therefore, we extend the Equation 6 to three contrastive views and add an extra global view without dropping any features as the third views to guarantee the learned $\mathbf{z}^{pre}$ main the global semantics instead of partial or even trivial features, i.e., $\mathbf{z}_1^{com} \sim G$ and $\mathbf{z}_2^{com} \sim G$. During the experiments, we find a adversarial graph sample perturbed from origial graph view can not only play the same role of original view in our expectation, but also achieve better robustness. A possible explanation is that there is still redundant information left the shared part of the two augmentation views as illustrated in Figure 3(b), especially when the implemented augmentations are not very aggressive. Thus, an adversarial view can further eliminate the trivial information from the learned $\mathbf{z}^{pre}$ for better generalization ability. We define the adversarial objective as follows:

$$\delta^* = \underset{\|\delta\|_\infty \leqslant \epsilon}{\operatorname{argmax}} \mathcal{L}_{\text{adv}}\left(t_1(G), t_2(G), G + \delta\right), \tag{9}$$

where the adversarial sample $G + \delta$ together with the two augmentation views, i.e., $t_1(G)$ and $t_2(G)$ are employed as the positive pair. Our implementation of crafting perturbation is spurred by recent work [42] that add perturbation δ on the output of first hidden layer $\mathbf{h}^{(1)}$ because it is empirically proved to generate more challenging view than adding perturbation on the initial node feature. Therefore, the adversarial contrastive objective is defined as:

$$\mathcal{L}_{\text{adv}} = \frac{1}{N} \sum_{i=1}^{N} \max_{\delta^*} \left[\mathcal{L}_{\text{CL}}\left(\mathbf{z}_{1,i}^{com}, G + \delta^*\right) + \mathcal{L}_{\text{CL}}\left(\mathbf{z}_{2,i}^{com}, G + \delta^*\right)\right]. \tag{10}$$

where the optimized perturbation δ' is solved by projected gradient descent (PGD) [22]

3.4 The Joint Objective

We design the joint objective of GraphCV by combining all of objectives above together. Given the graph $G \in \mathbf{G}$, the graph encoder f and all the decoders g can be optimized with the objective below:

$$\min_{f,g} \mathbb{E}_{G \in \mathbf{G}} \left[\mathcal{L}_{\text{pre}} + \lambda_r \mathcal{L}_{\text{recon}} + \lambda_a \max_{\|\delta\|_\infty \leqslant \epsilon} \mathcal{L}_{\text{adv}}\right], \tag{11}$$

where λ_r and λ_a are the coefficients to balance the magnitude of each loss term. Our proposed model is able to learn optimal representation illustrated in Figure 3(c) with the joint objective. We present the training algorithm of our model in Section F of Appendix.

4 Experiments

In this section, we demonstrate the empirical evaluation results of GraphCV on public graph benchmark datasets. Ablation study and robustness analysis are conducted to evaluate the effectiveness of the designs in GraphCV. We provide the dataset statistics, training details and more analysis about hyper-parameters and representation disentanglement can be found in the Appendix.

4.1 Experimental Setups

Datasets. We evaluate our model on five graph benchmark datasets from the field of bioinformatics, including MUTAG, PTC-MR, NCI1, DD, and PROTEINS, and other five from the field of social network, which are COLLAB, IMDB-B, RDT-B, RDT-M5K, and IMDB-M, for the task of graph-level property classification. Additionally, We use ogbg-molhiv from Open Graph Benchmark Dataset [15] to demonstrate our model's advantages over large-scale dataset. More details about dataset statistics are included in Section A of Appendix.

Baselines. Under the unsupervised representation learning setting, we compare GraphCV with the seven SOTA self-supervised learning methods GraphCL [44], InfoGraph[30], MVGRL [12], AD-GCL[31], GASSL[42], InfoGCL[40] and DGCL[18], as well as four classical unsupervised representation learning methods, including node2vec [9], sub2vec [2], graph2vec [25], and GVAE[16].

Evaluation Protocol. We follow the evaluation protocols in the previous works [30, 44, 18] to verify the effectiveness of our model. The learned representation is fine-tuned by a linear SVM classifier for task-specific prediction. We report the mean test accuracy evaluated by a 10-fold cross validation with standard deviation of five random seeds as the final performance. In addition, we follow the setting of semi-supervised representation learning from GraphCL on the ogbg-molhiv dataset, with the finetune label rates as 1%, 10%, and 20%. The final performance is reported as the mean ROC-AUC of five initialization random seeds

Implementation Details. We implement our framework with PyTorch and employ the data augmentation function provided by PyGCL library [48]. We choose GIN [41] as the backbone encoder and the model is optimized by Adam optimizer. There are two specific hyper-parameters in our model, namely λ_r and λ_a, the search space of them are $\{0.0, 1.0, 3.0, 5.0, 10.0\}$ and $\{0.0, 0.5, 1.0, 1.5, 2.0\}$, respectively. More details about implementation details is provided in the Section B of Appendix. All of the experiments are conducted on Nvidia GeForce RTX 2080ti GPU.

4.2 Overall Performance Comparison

Unsupervised representation learning. The overall performance comparison is shown in Table 1. From the results, we can have three observations: (1) The GCL-based methods generally yield higher performances than classical unsupervised learning methods, indicating the effectiveness of utilizing instance-level supervision; (2) InfoGCL and GASSL achieve better performances than GraphCL, which empirically proves the conclusion that InfoMax object could suffer from the overwhelmed information and thus more challenging augmentations or perturbations are in need to produce robust representations; (3) Our proposed GraphCV and DGCL consistently outperform other baselines, proving the advantage of disentangled representation. More importantly, ADGCL achieves state-of-the-art results on most of the datasets, which further demonstrate the success of our model to learn minimal yet sufficient representations.

Semi-supervised representation learning. The semi-supervised representation learning results for ogbg-molhiv are shown in Figure 4. It is obvious that our model gains significant improvements under the three label-rate fine-tuning settings. We also notice that as the label rate increases, the amount of improvement increases as well (1%, 1.8%, and 4.4% for label rate 1%, 10%, and 20%, respectively). A possible explanation could be that as more trainable data is included in the process of fine-tuning when the label rate

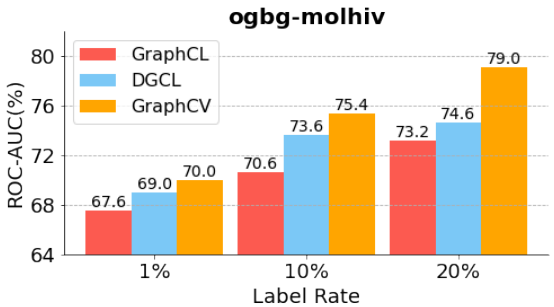


Figure 4: Performance comparison of semi-supervised learning on ogbg-molhiv.

Table 1: Overall comparison on multiple graph classification benchmarks. Results are reported as mean±std%, the best performance is bolded and runner-ups are underlined. "-" indicates the result is not reported in original papers.

	MUTAG	PTC-MR	COLLAB	NCI1	PROTEINS	IMDB-B	RDT-B	IMDB-M	RDT-M5K	DD
node2vec	72.6±10.2	58.6±8.0	-	54.9±1.6	57.5±3.6	-	-	-	-	-
sub2vec	61.1±15.8	60.0±6.4	-	52.8±1.5	53.0±5.6	55.3±1.5	71.5±0.4	36.7±0.8	36.7±0.4	-
graph2vec	83.2±9.3	60.2±6.9	-	73.2±1.8	73.3±2.1	71.1±0.5	75.8±1.0	50.4±0.9	47.9±0.3	-
InfoGraph	89.0±1.1	61.7±1.4	70.7±1.1	76.2±1.1	74.4±0.3	73.0±0.9	82.5±1.4	49.7±0.5	53.5±1.0	72.9±1.8
VGAE	87.7±0.7	61.2±1.8	-	-	-	70.7±0.7	87.1±0.1	49.3±0.4	52.8±0.2	-
MVGRL	89.7±1.1	62.5±1.7	-	-	-	74.2±0.7	84.5±0.6	51.2±0.5	-	-
GraphCL	86.8±1.3	63.6±1.8	71.4±1.2	77.9±0.4	74.4±0.5	71.1±0.4	89.5±0.8	-	56.0±0.3	78.6±0.4
InfoGCL	91.2±1.3	63.5±1.5	80.0±1.3	80.2±0.6	-	75.1±0.9	-	51.4±0.8	-	-
DGCL	92.1±0.8	65.8±1.5	**81.2±0.3**	81.9±0.2	76.4±0.5	75.9±0.7	91.8±0.2	51.9±0.4	56.1±0.2	-
AD-GCL	89.7±1.0	-	73.3±0.6	69.7±0.5	73.8±0.5	72.3±0.6	85.5±0.8	49.9±0.7	54.9±0.4	75.1±0.4
GASSL	90.9	64.6±6.1	78	80.2	-	74.2	-	51.7	-	-
GraphCV	**92.6±0.9**	**67.4±1.3**	80.5±0.5	**82.0±1.0**	**77.3±0.4**	**76.7±0.5**	**92.4±0.9**	**52.2±0.5**	**57.2±0.4**	**80.5±0.5**

increases, so does the affiliated redundant information, which as a result, deteriorate the performance even more. Therefore, removing redundant information causes a higher performance boost.

4.3 Ablation Study

To further verify the effectiveness of different modules in GraphCV, we perform ablation studies on each one of the module by creating the model variants illustrated below. The comparison results are shown in Table 2.

- **w/o Intra-view Recon.** Reconstruction is only executed within the cross view i.e., $w \neq w'$.
- **w/o Inter-view Recon.** Reconstruction is only executed within the same view i.e., $w = w'$.
- **w/o Adv. Training.** Adversarial view is discarded in the contrastive loss.

From Table 2 we can see that our model with the combination of cross-view reconstruction and adversarial training module outperforms all of the variants. Discarding any reconstruction view could cause the failure to reach the optimal situation illustrated in Figure 3. We can not guarantee the representation disentanglement assumption if we skip the inter-view reconstruction, and the sufficiency assumption may not hold if we abandon intra-view reconstruction. Either way, the predictive representations may suffer from enormous information loss during the contrastive learning and further lead to the performance deterioration. Compared with our model, the variant w/o Adv. Training may bring too much redundant information to the downstream classifier, therefore creating more confusions. The relatively larger performance deterioration for the two variants w/o Intra-view Recon and w/o Inter-view suggests the rule "better than nothing". That is, having redundant information is better than having it partially.

Table 2: Overall comparison of the model variants' performance. Results are reported as mean±std%, the best performance is bolded.

	MUTAG	PTC-MR	COLLAB	NCI1	PROTEINS	IMDB-B	RDT-B	IMDB-M	RDT-M5K	DD
w/o Intra Recon	91.5±1.2	65.8±1.3	78.4±0.7	79.6±0.7	75.6±0.5	75.4±0.8	92.0±0.4	51.5±0.4	55.8±0.6	79.3±0.7
w/o Inter Recon	91.0±0.9	64.7±1.4	78.0±0.8	78.7±1.2	74.9±0.7	75.0±0.6	91.1±0.7	50.8±0.2	55.6±0.4	79.0±0.8
w/o Adv. Training	92.1±0.6	66.8±0.5	80.3±0.5	81.2±0.9	77.0±0.3	76.4±0.6	92.2±1.0	52.0±0.4	56.8±0.5	80.1±0.6
GraphCV	**92.6±0.9**	**67.4±0.5**	**80.5±0.5**	**82.0±1.0**	**77.3±0.4**	**76.7±0.5**	**92.5±0.9**	**52.2±0.5**	**57.2±0.4**	**80.5±0.5**

4.4 Robustness Analysis

In this section, we conduct extra experiments on ogbg-molhiv dataset to evaluate the effectiveness of our design in ensuring the representation robustness under aggressive augmentation and perturbation. The results are shown in Figure 5. In the left two subplots, we plot accuracy verses edge perturbation and attribute masking strengths, respectively. Specifically, we keep the GraphCL and our proposed GraphCV under the same hyper-parameter setting and set the λ_r and λ_a of GraphCV as 5.0 and 0.5, respectively. From the results we can see that GraphCV not only consistently outperforms GraphCL but also is less affected by larger augmentation strengths. Similar observation can be find in the right two subplots, where we compare our method with GASSL under different perturbation bounds and attack steps to demonstrate its robustness against adversarial attacks. Since both our model and GASSL use GIN as the backbone network, we hereby add the performance of GIN as the compared baseline. Although aggressive adversarial attacks can largely deteriorate the performance, our proposed GraphCV still achieves more robust performance than GASSL.

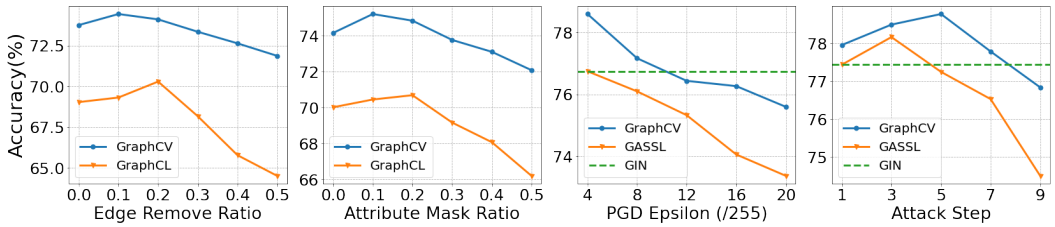


Figure 5: Performance versus augmentation strengths, perturbation bound and attack step.

5 Related Work

Graph contrastive learning. Contrastive learning is firstly proposed in the compute vision field [5] and raises a surge of interests in the area of self-supervised graph representation learning for the past few years. The principle behind contrastive learning is to utilize the instance-level identity as supervision and maximize the consistency between positive pairs in hidden space through designed contrast mode. Previous graph contrastive learning works generally rely on various graph augmentation (transformation) techniques [36, 26, 13, 44, 30] to generate positive pair from original data as similar samples. Recent works in this field try to improve the effectiveness of graph contrastive learning by finding more challenge view [31, 40, 43] or adding adversarial perturbation [42]. However, most of the existing methods contrast over entangled embeddings, where the complex intertwined information may pose obstacles to extracting useful information for downstream tasks. Our model is spared from the issue by contrasting over disentangled representations.

Disentangled representation learning on graphs. Disentangled representation learning arises from the computer vision field [14, 46] to disentangle the heterogeneous latent factors of the representations, and therefore making the representations more robust and interpretable [4]. This idea has now been widely adopted in graph representation learning. [20, 21] utilizes neighborhood routing mechanism to identify the latent factors in the node representations. Some other generative models [16, 29] utilize Variational Autoencoders to balance reconstruction and disentanglement. Recent work [18] outspreads the application of disentangled representations learning in self-supervised graph learning by contrasting the factorized representations. Although these methods gain significant benefit from the representation disentanglement, the underlined excessive information could still overload the model, thus resulting in limited capacities. Our model targets the issue by removing the redundant information that is considered irrelevant to the graph property.

Graph information bottleneck. The Information bottleneck (IB) [32] has been widely adopted as a critical principle of representation learning. A representation contains minimal yet sufficient information is considered to be in compliance with the IB priciple and many works [3, 28, 7] have empirically and theoretically proved that representation agree with IB principle is both informative and robust. Recently, IB principle is also borrowed to guide the representation learning of graph structure data. Current methods [37, 40, 31] usually propose different regularization designs to learn compressed yet informative representations in accordance with IB principle. We follow the information bottleneck to learn the expressive and robust representation from disentangled information in this work.

6 Conclusion

In this paper, we study graph representation learning in light of information bottleneck. To reach the optimum we illustrate, we propose a novel model, namely GraphCV, which is designed to disentangle the essential factors from augmented graph through a cross-view reconstruction mechanism so that the information entanglement brought by augmentations will not cause the loss of predictive information during contrastive learning. We also add an adversarial view as the third view of the contrastive learning to further remove redundant information and enhance representation robustness. In addition, we theoretically analyze the effectiveness of each component in our model and derive the objective based on the analysis. Extensive experiments on multiple graph benchmark datasets and different settings prove the ability of GraphCV to learn robust and informative graph representation. In the future, we can explore how to come up with a practical objective to further decrease the upper bound of the mutual information between the disentangled representations and try to utilize more efficient training strategy to make the proposed model more time-saving on large-scale graphs.

References

[1] Alessandro Achille and Stefano Soatto. Emergence of invariance and disentanglement in deep representations. *JMLR*, 2018.

[2] Bijaya Adhikari, Yao Zhang, Naren Ramakrishnan, and B. Aditya Prakash. Sub2Vec: Feature Learning for Subgraphs. In *KDD*, 2018.

[3] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep Variational Information Bottleneck. *ICLR*, 2017.

[4] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation Learning: A Review and New Perspectives. *TPAMI*, 2013.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *ICML*, 2020.

[6] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

[7] Marco Federici, Anjan Dutta, Patrick Forré, Nate Kushman, and Zeynep Akata. Learning robust representations via multi-view information bottleneck. In *ICLR*, 2020.

[8] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2014.

[9] Aditya Grover and Jure Leskovec. node2vec: Scalable Feature Learning for Networks. In *KDD*, 2016.

[10] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs. In *NeurIPS*, 2017.

[11] William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.

[12] Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive Multi-View Representation Learning on Graphs. In *ICML*, 2020.

[13] Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *ICML*, 2020.

[14] Jun-Ting Hsieh, Bingbin Liu, De-An Huang, Li F Fei-Fei, and Juan Carlos Niebles. Learning to Decompose and Disentangle Representations for Video Prediction. In *NeurIPS*, 2018.

[15] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open Graph Benchmark: Datasets for Machine Learning on Graphs. In *NeurIPS*, 2020.

[16] Thomas N. Kipf and Max Welling. Variational Graph Auto-Encoders. In *NeurIPS*, 2016.

[17] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*, 2017.

[18] Haoyang Li, Xin Wang, Ziwei Zhang, Zehuan Yuan, Hang Li, and Wenwu Zhu. Disentangled Contrastive Learning on Graphs. In *NeurIPS*, 2021.

[19] Sihang Li, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. Let invariant rationale discovery inspire graph contrastive learning. In *ICML*, 2022.

[20] Yanbei Liu, Xiao Wang, Shu Wu, and Zhitao Xiao. Independence promoted graph disentangled networks. In *AAAI*, 2020.

[21] Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu. Disentangled Graph Convolutional Networks. In *ICML*, 2019.

[22] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

[23] Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020.

[24] Daniel Moyer, Shuyang Gao, Rob Brekelmans, Aram Galstyan, and Greg Ver Steeg. Invariant representations without adversarial training. *NeurIPS*, 2018.

[25] A. Narayanan, Mahinthan Chandramohan, R. Venkatesan, Lihui Chen, Yang Liu, and Shantanu Jaiswal. graph2vec: Learning Distributed Representations of Graphs. *ArXiv*, 2017.

[26] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training. In *KDD*, 2020.

[27] Joshua Robinson, Li Sun, Ke Yu, Kayhan Batmanghelich, Stefanie Jegelka, and Suvrit Sra. Can contrastive learning avoid shortcut solutions? In *NeurIPS*, 2021.

[28] Ravid Shwartz-Ziv and Naftali Tishby. Opening the Black Box of Deep Neural Networks via Information. *arXiv:1703.00810 [cs]*, April 2017.

[29] Martin Simonovsky and Nikos Komodakis. GraphVAE: Towards generation of small graphs using variational autoencoders. In *ICLR*, 2018.

[30] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization. In *ICLR*, 2019.

[31] Susheel Suresh, Pan Li, Cong Hao, and Jennifer Neville. Adversarial Graph Augmentation to Improve Graph Contrastive Learning. In *NeurIPS*, 2021.

[32] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[33] Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. In *ICLR*, 2019.

[34] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv e-prints*, 2018.

[35] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *ICLR*, 2018.

[36] Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. Deep Graph Infomax. In *ICLR*, 2019.

[37] Tailin Wu, Hongyu Ren, Pan Li, and Jure Leskovec. Graph Information Bottleneck. In *NeurIPS*. Curran Associates, Inc., 2020.

[38] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised Feature Learning via Non-Parametric Instance Discrimination. In *CVPR*, 2018.

[39] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. Adversarial examples improve image recognition. In *CVPR*, 2020.

[40] Dongkuan Xu, Wei Cheng, Dongsheng Luo, Haifeng Chen, and Xiang Zhang. InfoGCL: Information-Aware Graph Contrastive Learning. In *NeurIPS*, 2021.

[41] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How Powerful are Graph Neural Networks? In *ICLR*, 2019.

[42] Longqi Yang, Liangliang Zhang, and Wenjing Yang. Graph Adversarial Self-Supervised Learning. In *NeurIPS*, 2021.

[43] Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In *ICLR*, 2021.

[44] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph Contrastive Learning with Augmentations. In *NeurIPS*, 2020.

[45] Junchi Yu, Tingyang Xu, Yu Rong, Yatao Bian, Junzhou Huang, and Ran He. Graph Information Bottleneck for Subgraph Recognition. In *ICLR*, 2020.

[46] Long Zhao, Yuxiao Wang, Jiaping Zhao, Liangzhe Yuan, Jennifer J. Sun, Florian Schroff, Hartwig Adam, Xi Peng, Dimitris Metaxas, and Ting Liu. Learning View-Disentangled Human Pose Representation by Contrastive Cross-View Mutual Information Maximization. In *CVPR*, 2021.

[47] Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. Freelb: Enhanced adversarial training for natural language understanding. In *ICLR*, 2020.

[48] Yanqiao Zhu, Yichen Xu, Qiang Liu, and Shu Wu. An Empirical Study of Graph Contrastive Learning. *arXiv.org*, 2021.

A Summary of Datasets

In this work, we use ten datasets from TU Benchmark Datasets [23] to evaluate our proposed GraphCV under unsupervised setting, where five of them are biochemical datasets and the other five belong to social network datasets. Besides, we utilize the ogng-molhiv dataset from Open Graph Benchmark (OGB) [15] to further evaluate GraphCV under semi-supervised setting. The statistics of these datasets are provided in Table 3

Dataset	#Graphs	Avg #Nodes	Avg #Edges	#Class	Metric	Category
MUTAG	188	17.93	19.79	2	Accuracy	biochemical
PTC-MR	344	14.29	14.69	2	Accuracy	biochemical
PROTEINS	1,113	39.06	72.82	2	Accuracy	biochemical
NCI1	4,110	29.87	32.30	2	Accuracy	biochemical
DD	1,178	284.32	715.66	2	Accuracy	biochemical
COLLAB	5,000	74.49	2457.78	3	Accuracy	social network
IMDB-B	1,000	19.77	96.53	2	Accuracy	social network
RDT-B	2,000	429.63	497.75	2	Accuracy	social network
IMDB-M	1,500	13.00	65.94	3	Accuracy	social network
RDT-M5K	4,999	508.52	594.87	5	Accuracy	social network
ogbg-molhiv	41,127	25.50	27.50	2	ROC-AUC	MoleculeNet

Table 3: Summary of datasets statistics.

All of the eleven datasets are public available, we attach attach their links as follow:

- TU datasets: `https://chrsmrrs.github.io/datasets/docs/datasets/`
- ogbg-molhiv dataset: `https://ogb.stanford.edu/docs/graphprop/#ogbg-mol`

B Implementation Details

All experiments are conducted with the following settings:

- Operating System: Ubuntu 18.04.5 LTS
- CPU: AMD(R) Ryzen 9 3900x
- GPU: NVIDIA GeForce RTX 2080ti
- Software: Python 3.8.5; Pytorch 1.10.1; PyTorch Geometric 2.0.4; PyGCL 0.1.2; Numpy 1.20.1; scikit-learn 0.24.1.

We choose GIN [41] as the backbone graph encoder and the model is optimized through Adam optimizer. We follow [44, 42, 18] to employ a linear SVM classifier for downstream task-specific classification. The graph augmentation operations used in our work are same as [44], including node dropping, edge perturbation, attribute masking and subgraph sampling, all of them are borrowed from the implementation of [48]. There are two specific hyper-parameters in our model, namely λ_r and λ_a, the search space of them are $\{0.0, 1.0, 3.0, 5.0, 10.0\}$ and $\{0.0, 0.5, 1.0, 1.5, 2.0\}$, respectively. For other important hyper-parameters, we find the best value of pre-training epoch from $\{20, 50, 100, 200, 300\}$, learning rate from $\{0.01, 0.005, 0.001, 0.0005, 0.0001\}$, embedding dimension from $\{32, 64, 128, 256, 512\}$, number of GNN layers from $\{2, 3, 4, 5\}$, batch size from $\{32, 64, 128, 256, 512\}$ (except for ogbg-molhiv $\{64, 128, 256, 512, 1024\}$). Besides, we fix the perturbation bound ϵ, ascent step size α and ascent step T as 0.008, 0.008 and 5 during hyper-parameter fine-tuning.

C Proof

C.1 Proof of Theorem 1

We repeat Theorem 1 as follows.

Theorem 3 *Suppose $f(\cdot)$ is a GNN encoder as powerful as 1-WL test. Let $g_{pre}(\cdot)$ elicits only the augmentation information from $\mathbf{z}$ meanwhile $g_{com}(\cdot)$ extracts the essential factors of G from $\mathbf{z}_1$ and $\mathbf{z}_2$. Then we have:*

$$I\left(t_1(G); \mathbf{z}_2^{com}, \mathbf{z}_2^{pre}\right) \geqslant I\left(\mathbf{z}_1^{pre}; \mathbf{z}_2^{pre}\right) \text{ where } G \in \mathcal{G} \text{ and } t_1(\cdot), t_2(\cdot) \in \mathcal{T}.$$

Proof. According to the assumption in Theorem 1, for any two graphs $G, G' \in \mathcal{G}$, if $G \cong G'$ then we have $\mathbf{z} = \mathbf{z}'$, where $\mathbf{z} = f(G)$ and $\mathbf{z}' = f(G')$.

Besides, $\mathbf{z}^{pre} = g_{pre}(\mathbf{z})$ is specific to the predictive factors and $\mathbf{z}^{com} = g_{com}(\mathbf{z})$ is particular to the non-predictive factors, which means $\mathbf{z}^{pre}$ and $\mathbf{z}^{com}$ are mutually excluded and $\mathbf{z}^{pre} \sim G$. So we have,

$$\begin{aligned} p\left(\mathbf{z}^{pre}, \mathbf{z}^{com}\right) &= p\left(\mathbf{z}^{pre}\right) p\left(\mathbf{z}^{com}\right) \\ p\left(\mathbf{z}^{pre}, \mathbf{z}^{com} \mid t(G)\right) &= p\left(\mathbf{z}^{pre} \mid t(G)\right) p\left(\mathbf{z}^{com} \mid t(G)\right). \end{aligned} \tag{12}$$

Then, we want to prove that given three random variables a, b and c, if they satisfy $p(b, c) = p(b)\, p(c)$ and $p(b, c \mid a) = p(b \mid a)\, p(c \mid a)$, we have $I(a, b \mid c) = I(a, b)$. According to the definition of mutual information, we have that,

$$\begin{aligned} & I(a; b \mid c) = \\ &= \sum_a \sum_b \sum_c p(a, b, c) \log \frac{p(a, b, c)\, p(c)}{p(a, c)\, p(b, c)} \\ &= \sum_a \sum_b \sum_c p(a)\, p(b, c \mid a) \log \frac{p(b, c \mid a)\, p(c)}{p(c \mid a)\, p(b)\, p(c)} \\ &= \sum_a \sum_b \sum_c p(a)\, p(b \mid a)\, p(c \mid a) \log \frac{p(b \mid a)\, p(c \mid a)}{p(c \mid a)\, p(b)} \\ &= \sum_a \sum_b p(a)\, p(b \mid a) \log \frac{p(b \mid a)}{p(b)} \\ &= \sum_a \sum_b p(a, b) \log \frac{p(b \mid a)}{p(b)} \\ &= I(a; b). \end{aligned} \tag{13}$$

After that, by applying the chain rule to $I\left(t_1(G); \mathbf{z}_2^{pre}, \mathbf{z}_2^{com}\right)$, we have,

$$\begin{aligned} I\left(t_1(G); \mathbf{z}_2^{pre}, \mathbf{z}_2^{com}\right) &= I\left(t_1(G); \mathbf{z}_2^{pre} \mid \mathbf{z}_2^{com}\right) + I\left(t_1(G); \mathbf{z}_2^{com}\right) \\ &\overset{(2)}{=} I\left(t_1(G); \mathbf{z}_2^{pre}\right) + I\left(t_1(G); \mathbf{z}_2^{com}\right) \\ &\overset{(a)}{\geqslant} I\left(t_1(G); \mathbf{z}_2^{pre}\right) \\ &\overset{(b)}{\geqslant} I\left(\mathbf{z}_1^{com}, \mathbf{z}_1^{pre}; \mathbf{z}_2^{pre}\right) \\ &\overset{(2)}{=} I\left(\mathbf{z}_1^{com}; \mathbf{z}_2^{pre}\right) + I\left(\mathbf{z}_1^{pre}; \mathbf{z}_2^{pre}\right) \\ &\overset{(a)}{\geqslant} I\left(\mathbf{z}_1^{pre}; \mathbf{z}_2^{pre}\right), \end{aligned} \tag{14}$$

where $\overset{(2)}{=}$ is derived from the conclusion we get in Equation 13, $\overset{(a)}{\geqslant}$ is based on the non-negativity of mutual information, i.e., $I(;) \geqslant 0$, and $\overset{(b)}{\geqslant}$ is because data processing inequality [6]. Finally, we reach to the lower bound of $I\left(t_1(G); \mathbf{z}_2^{pre}, \mathbf{z}_2^{com}\right)$ in Equation 13, thus we can maximize the consistency between the information we capture from the two augmentation graph views by minimizing $\mathcal{L}_{\text{pre}}$.

C.2 Proof of Theorem 2

We repeat Theorem 2 as follows.

Theorem 4 *Assume q is a Gaussian distribution, g_r is the parameterized reconstruction model which infer $\mathbf{z}_w$ from $(\mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com})$. Then we have:*

$$H\left(\mathbf{z}_w \mid \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right) \leqslant \left\|\mathbf{z}_w - g_r\left(\mathbf{z}_w^{pre} \odot \mathbf{z}_{w'}^{com}\right)\right\|_2^2 \text{ where } w = w' \text{ or } w \neq w'.$$

Proof. To reconstruct the entangled representation $\mathbf{z}_w$ from its corresponding non-predictive representation $\mathbf{z}_w^{pre}$ and the predictive representation of any augmentation view $\mathbf{z}_{w'}^{com}$ (w and w' are not necessarily equal), we need to minimize the conditional entropy:

$$H\left(\mathbf{z}_w \mid \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right) = -\mathbb{E}_{p\left(\mathbf{z}_w, \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right)}\left[\log p\left(\mathbf{z}_w \mid \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right)\right] \tag{15}$$

Since the real distribution of $p\left(\mathbf{z}_w \mid \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right)$ is unknown and intractable, we hereby introduce a variational distribution $q\left(\mathbf{z}_w \mid \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right)$ to approximate it. Therefore, we have,

$$\begin{aligned} &\mathbb{E}_{p\left(\mathbf{z}_w, \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right)}\left[\log p\left(\mathbf{z}_w \mid \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right)\right] = \\ &\qquad \mathbb{E}_{p\left(\mathbf{z}_w, \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right)}\left[\log q\left(\mathbf{z}_w \mid \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right)\right] \\ &\qquad + D_{\mathrm{KL}}\left(p\left(\mathbf{z}_w \mid \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right) \| q\left(\mathbf{z}_w \mid \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right)\right). \end{aligned} \tag{16}$$

Due to the non-negativity of KL-divergence between any two distributions, it is safe to say $-\mathbb{E}_{p\left(\mathbf{z}_w, \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right)}\left[\log q\left(\mathbf{z}_w \mid \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right)\right]$ is the upper bound of $H\left(\mathbf{z}_w \mid \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right)$. Based on the assumption of Theorem 2, let $q\left(\mathbf{z}_w \mid \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right)$ being a Gaussian distribution $\mathcal{N}\left(\mathbf{z}_w \mid g_r\left(\mathbf{z}_w^{pre} \odot \mathbf{z}_{w'}^{com}\right), \sigma^2\mathbf{I}\right)$, where $g_r(\cdot)$ is the reconstruct network that predict $\mathbf{z}_w$ from $(\mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com})$ and σ is the variance. Thus we have,

$$\begin{aligned} H\left(\mathbf{z}_w \mid \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right) &\leqslant -\mathbb{E}_{p\left(\mathbf{z}_w, \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right)}\left[\log q\left(\mathbf{z}_w \mid \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right)\right] \\ &= -\mathbb{E}_{p\left(\mathbf{z}_w, \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right)}\left[\log\left(\frac{1}{\sqrt{2\pi I}\sigma} e^{-\frac{1}{2}\frac{\left(\mathbf{z}_w - g_r\left(\mathbf{z}_w^{pre} \odot \mathbf{z}_{w'}^{com}\right)\right)^2}{(\sigma^2\mathbf{I})}}\right)\right] \\ &= -\mathbb{E}_{p\left(\mathbf{z}_w, \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right)}\left[\log\left(\frac{1}{\sqrt{2\pi I}\sigma}\right) - \frac{\left(\mathbf{z}_w - g_r\left(\mathbf{z}_w^{pre} \odot \mathbf{z}_{w'}^{com}\right)\right)^2}{2\sigma^2\mathbf{I}}\right] \end{aligned} \tag{17}$$

Hence, we get the upper bound of $H\left(\mathbf{z}_w \mid \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right)$ as Equation 17. To minimize the value of the unsolvable entropy, we can instead minimize the value of its upper bound and thereby derive the objective function as follow by neglecting the constant terms,

$$\min \mathbb{E}_{p\left(\mathbf{z}_w, \mathbf{z}_w^{pre}, \mathbf{z}_{w'}^{com}\right)} \left\|\mathbf{z}_w - g_r\left(\mathbf{z}_w^{pre} \odot \mathbf{z}_{w'}^{com}\right)\right\|_2^2. \tag{18}$$

Since we adopt two augmentation views and propose the cross-view reconstruction mechanism in our method, we can minimize the entropy by minimizing $\mathcal{L}_{\text{recon}}$ and thus guarantee the disentanglement of $\mathbf{z}^{pre}$ and $\mathbf{z}^{com}$.

D Effects of Representation Disentanglement

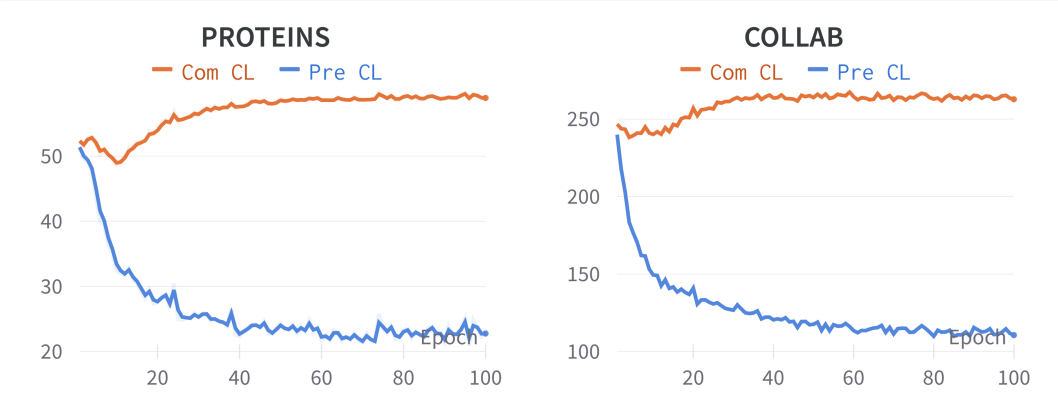


Figure 6: InfoNCE loss of the two disentangled representations between the two augmentation graph views, where orange lines are the InfoNCE loss between the two non-predictive representations and blue lines are the InfoNCE loss between the two predictive representaitons

In this section, we set experiments to further investigate the representation disentanglement of our proposed GraphCV. Specifically, we use the InfoNCE loss [34] to dynamically measure the representation difference between the two augmentation graph views based on the two disentangled representations, where blue lines indicate the InfoNCE loss between $\mathbf{z}_1^{pre}$ and $\mathbf{z}_2^{pre}$ and orange lines represent the InfoNCE loss between $\mathbf{z}_1^{com}$ and $\mathbf{z}_2^{com}$. For simplicity, we only demonstrate the first 100 pre-training epochs of PROTEINS and COLLAB in Figure 6, we can observe similar phenomena on other datasets. From the loss curves in Figure 6 we can find that contrastive loss between predictive representations gradually decreases, indicating the predictive representation is optimized to capture all the shared information between the two augmentation view. Meanwhile, we can see contrastive loss between the non-predictive representations achieve a noticeable increases, which is consistent with our expectation that the two independent sampled augmentation operators cause a distribution shift between the two augmentation views.

E Hyper-parameter Sensitivity

In this section, we study the impacts of some important hyper-parameters in our method, including reconstruction loss coefficient λ_r, adversarial loss coefficient λ_a, embedding dimension d, batch size $|\mathcal{B}|$ and number of GNN layers L. Here, we select four datasets, i.e., MUTAG, PROTEINS, RDT-B and COLLAB, to report for simplicity because the four datasets cover different domains and scales. We illustrate the impacts of these hyper-parameters in the figures below.

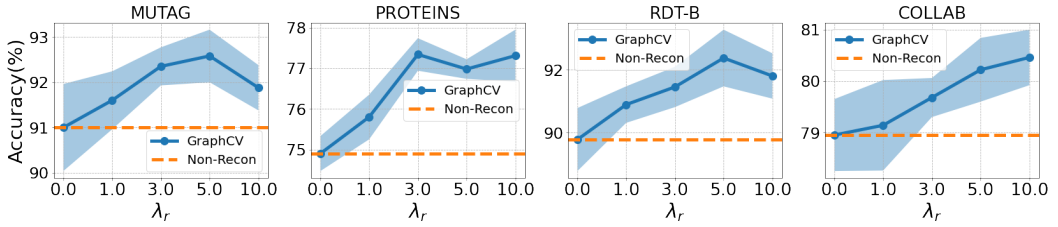


Figure 7: Impact of reconstruction loss coefficient λ_r on different datasets, we specify the non-reconstruction situation ($\lambda_r = 0$) with the dashed line for comparison.

From the result demonstrated in Figure 7, we can see the optimal reconstruction loss coefficient λ_r is different dependent on the specific dataset, but all the values in our experiment can enhance the performance compared with non-reconstruction variant, i.e., $\lambda_r = 0$, indicating the effectiveness of our proposed cross-view reconstruction mechanism.

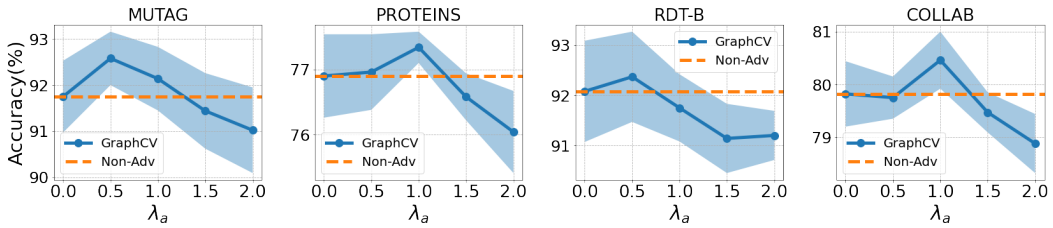


Figure 8: Impact of adversarial loss coefficient λ_a on different datasets, we specify the non-adversarial situation ($\lambda_a = 0$) with the dashed line for comparison.

The Figure 8 shows that we could further raise the model performance through the adversarial training, which proves a robust representation with less redundant information usually achieve more performance gain compared with the brittle one. During this process, we need to choose a appropriate adversarial loss coefficient λ_a, otherwise a too large λ_a may hurt the information sufficiency of the learned representation.

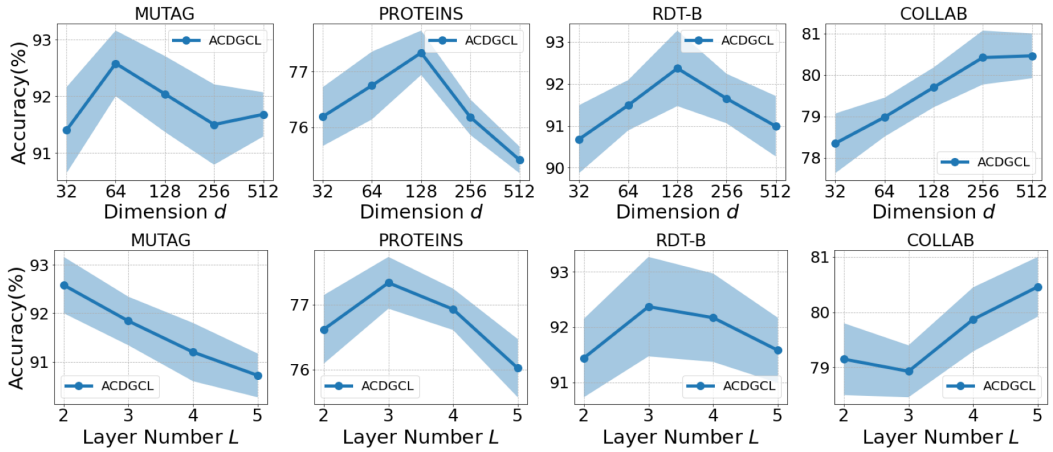


Figure 9: Impact of a embedding dimension d and GNN layer number L on different datasets.

We put the impacts of embedding dimension d and GNN layer number L together because we can find a similar observation form their experimental results. From Figure 9, we observe that the optimal values of the two hyper-parameters generally increase as the dataset scale increases. The reason behind this phenomena could be large datasets usually contain more latent factors than the small datasets, therefore a model with larger capacity is needed to fit the large datasets. However, such high-capacity message-passing model will deteriorate the performance of small dataset because it may cause the learned representation over-smoothing and hence less informative.

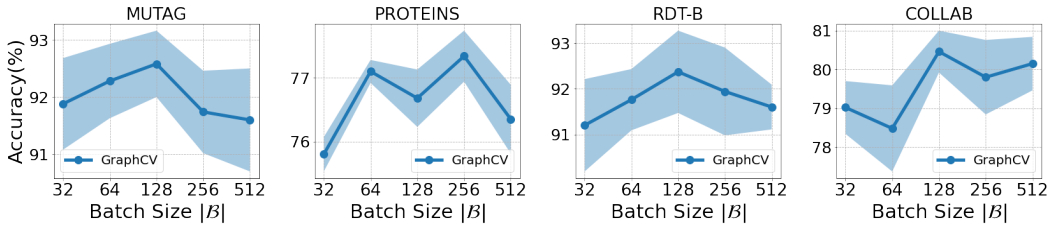


Figure 10: Impact of batch size $|\mathcal{B}|$ on different datasets.

The effect of batch size $|\mathcal{B}|$ is shown in Figure 10, we can see the performance variation of these datasets under different batch size is relatively small. Most of the datasets achieve best performance when the $|\mathcal{B}|$ is set to 128.

F Training Algorithm

In this sectionm we summarized the details of our proposed method in Algorithm 16

Algorithm 1: The training algorithm of **GraphCV**

Input: Graph dataset $\mathcal{G} = \{G_i = (V_i, E_i)\}_{i=1}^{N}$; augmentation family $\mathcal{T}$; loss coefficient λ_r, λ_a; ascernt step T; ascent step size α; perturbation bound ϵ.

Output: The disentangled predictive representations $\mathbf{Z}^{pre} = \{\mathbf{z}_i^{pre}\}_{i=1}^{N}$

for *each training epoch* **do**
  for *sampled minibatch* $\mathcal{B} = \{G_i\}_{i=1}^{|\mathcal{B}|}$ **do**
    for $G_i \in \mathcal{B}$ **do**
      $\mathbf{z}_{1,i} = f\left(t_1(G_i)\right), \mathbf{z}_{2,i} = f\left(t_2(G_i)\right)$; $\triangleright\ t_1(\cdot), t_2(\cdot) \in \mathcal{T}$
      $\mathbf{z}_{1,i}^{pre} = g_{pre}\left(\mathbf{z}_{1,i}\right), \mathbf{z}_{2,i}^{pre} = g_{pre}\left(\mathbf{z}_{1,i}\right)$;
      $\mathbf{z}_{1,i}^{com} = g_{com}\left(\mathbf{z}_{1,i}\right), \mathbf{z}_{2,i}^{com} = g_{com}\left(\mathbf{z}_{1,i}\right)$;
    Calculate $\mathcal{L}_{\text{pre}}$ according to Equation 6;
    Calculate $\mathcal{L}_{\text{recon}}$ according to Equation 8 ;
    $\mathcal{L} \leftarrow \mathcal{L}_{\text{pre}} + \lambda_r \mathcal{L}_{\text{recon}}$;
    $\delta_0 \leftarrow U(-\epsilon, \epsilon)$;
    for *each* $t = 1$ *to* T **do**
      Calculate the $\mathcal{L}_{\text{adv}}$ according to Equation 10;
      $\delta_t \leftarrow \delta_{t-1} + \alpha \nabla_\delta \mathcal{L}_{\text{adv}}$; $\triangleright$ Update perturbation to maximize $\mathcal{L}_{\text{adv}}$
      $\mathcal{L} \leftarrow \mathcal{L} + \frac{\lambda_a}{T} \mathcal{L}_{\text{adv}}$
    Update the parameter θ of f and g with the gradient $\nabla_\theta \mathcal{L}(\theta, \mathcal{B})$ over a minibatch;
return $\mathbf{Z}^{pre} = \{\mathbf{z}_i^{pre}\}_{i=1}^{N}$, where $\mathbf{z}_i^{pre} = g_{pre}\left(f(G_i)\right)$