Graph Contrastive Learning with Cross-view Reconstruction

Qianlong Wen[1], **Zhongyu Ouyang**[1], **Chunhui Zhang**[2], **Yiyue Qian**[1],
Yanfang Ye[1], **Chuxu Zhang**[2]
[1]University of Notre Dame, [2]Brandeis University
[1]{qwen,zouyang2,yqian5,yye7}@nd.edu;
[2]{chunhuizhang,chuxuzhang}@brandeis.edu

Abstract

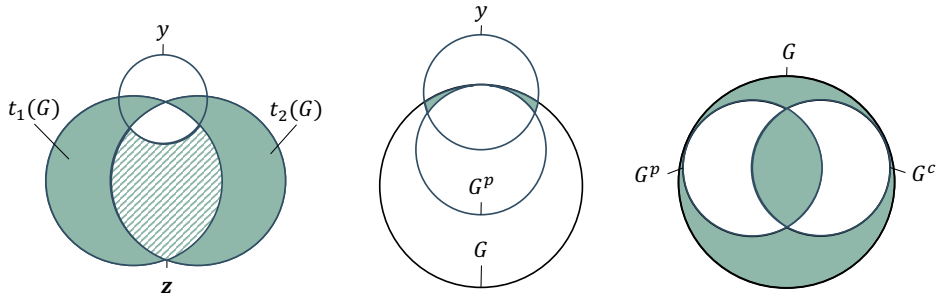
Graph self-supervised learning is commonly taken as an effective framework to tackle the supervision shortage issue in the graph learning task. Among different existing graph self-supervised learning strategies, graph contrastive learning (GCL) has been one of the most prevalent approaches to this problem. Despite the remarkable performance those GCL methods have achieved, existing GCL methods that heavily depend on various manually designed augmentation techniques still struggle to alleviate the feature suppression issue without risking losing task-relevant information. Consequently, the learned representation is either brittle or unilluminating. In light of this, we introduce the **Graph** Contrastive Learning with **C**ross-**V**iew Reconstruction (GraphCV), which follows the information bottleneck principle to learn minimal yet sufficient representation from graph data. Specifically, GraphCV aims to elicit 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 representation across different augmentation views, we introduce a cross-view reconstruction mechanism to pursue the disentanglement of the two learned representations. Besides, an adversarial view perturbed from the original view is added as the third view for the contrastive loss to guarantee the intactness of the global semantics and improve the representation robustness. 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 [13]. The current state-of-the-art supervised GRL methods are mostly based on Graph Neural Networks (GNNs) [20, 38, 12, 46], which require a large amount of task-specific supervised information. Despite the remarkable performance, 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 very costly to obtain enough annotated labels, especially in certain fields like biochemistry. Therefore, many recent works [29, 14, 33] study how to fully utilize the unlabeled information on graph and further stimulate 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) [39] to maximize the agreements of the positive pairs while minimizing that of the negative pairs in the embedding space. However, the graph contrastive learning paradigm guided by the InfoMax principle is insufficient to learn robust and transferable

36th Conference on Neural Information Processing Systems (NeurIPS 2022).



(a) Contrastive Learning (b) Sufficiency (c) Disentanglement

Figure 1: Illustration of the relation between graph G, label y, predictive feature subsets G^p and non-predictive feature subset G^c in terms of information entropy. Ideally, the green areas in the three figures is null. (a) The usual optimization result of graph contrastive learning, where the shared features of two augmentation view is extracted for the learned representation $\mathbf{z}$. Owing to the lack of supervision or domain knowledge, redundant and biased information (shadow area) is usually included in $\mathbf{z}$; (b) G^p cover the feature subset which is sufficient to make correct graph label identification ($I(y; G \mid G^p) = 0$), other features (G^c) is either useless or misguiding; (c) G^p and G^c are supposed to be mutually disentangled with each other ($I(G^p; G^c) = 0$). The union of them cover all the features of original data.

representation. State-of-the-art GCL methods [29, 14, 51] usually rely on augmentor(s) $t(\cdot)$ (e.g., Identity, Subgraph Sampling, Node Dropping, Edge Removing and Attributes Masking.) applied to the anchor graph G to generate a positive pair of graphs $t_1(G)$ and $t_2(G)$. Then, the graph feature encoder f will be trained to ensure the representation consistency within the positive pair, i.e., $\mathbf{z} = f(t_1(G)) = f(t_2(G))$. Consequently, such training strategy is heavily dependent on the choice and strength of graph augmentation techniques. To be more specific, moderate graph augmentation will push encoders to capture redundant and biased information [36], which could inadvertently suppress the space of important predictive features and negatively affect the representation transferability via the so-called "shortcut" solution [10, 26]. A more intuitively illustration is provided in the Figure 1 (a), where the shared part of the two augmentation view include both predictive information (the overlapping area with y) and non-predictive information (shadow area). Such optimization result usually yield lower contrastive loss, however, it has been empirically proved that the redundant information could lead to poor robustness [30], especially under the out-of-distribution (OOD) setting [49]. We provide a showcase example in Appendix A to illustrate the OOD scenario on graph learning task. On the other hand, overly aggressive augmentation may easily lead to another extreme where many predictive features are randomly dropped and the learned representation does not contain sufficient predictive information for downstream instance discrimination. Recent works [34, 22, 50] propose to use automated augmentations to extract the invariant rationale features [42, 40]. These methods assume the most salient sub-structure (those are resistant to graph augmentation) is sufficient to make rational and correct label identification, and thereby implement trainable augmentation operations (e.g., edge deleting, node dropping) to strictly regularize the graph topological structure. Despite that these methods can alleviate the aforementioned feature suppression issue to some extent, they still suffer from inherent limitation. The harsh regularization may force the encoders focusing on the easy-learned "shallower" features (e.g. graph size and node degree), which might be helpful under certain domains but not necessarily for others [4], thus fail to guarantee stronger robustness. Therefore, the GCL methods guided with the saliency philosophy is not flexible enough to balance the representation sufficiency and robustness without the guidance of explicit domain knowledge. To reconcile the robustness and sufficiency of the learned representation, a method which can reduce redundant and biased information without sacrificing the sufficiency of the predictive graph features is in urgent need.

Recently, the information bottleneck (IB) principle [35] has been introduced to the graph learning, which encourages extracting minimal yet sufficient information for representation learning. The core idea of IB principle is in accordance with the ultimate optimization objective to solve the feature suppression issue [30], thus shed more light on this problem. Moreover, the representation learning guided by the IB principle has been empirically proved to generate more robust and transferable representations at different domains [41]. Therefore, a graph contrastive learning framework in accordance with the IB principle is promising in balancing the representation robustness and sufficiency. Given an input graph G, we denote G^p and G^c as its predictive feature subset and the complementary non-predictive feature subset, respectively. According to the assumption of recent studies about rationale invariance discover [42, 40], the two features subsets would the satisfy $I(y; G \mid G^p) = 0$ (sufficiency condition) and disentanglement condition (i.e., $I(G^p; G^c) = 0$). We illustrate the relations among

the two feature subsets and G in Figure 1 (b) and (c). It is inevitable that the learned representation maintains some redundant information for a specific downstream task. However, a GCL framework under the guidance of the IB principle is expected to suppress the feature space of G^c as much as possible while keeping the predictive feature G^p intact simultaneously in the learned representation.

In this paper, we propose the novel **Graph** Contrastive Learning with **C**ross-**V**iew Reconstruction, named GraphCV, to pursue the optimization objective of the IB principle. Specifically, GraphCV consists of a graph encoder followed with two decoders that are trained to extract information specific to the predictive and non-predictive features, respectively. To approximate the disentanglement objective, we propose the reconstruction-based representation learning scheme, including the intra-view and inter-view reconstructions, to reconstruct the original learned representation with the two separated feature subsets. Furthermore, the encoded representation from the original view perturbed in the adversarial fashion serves as the third view when computing the contrastive loss, apart from the predictive relevant representations of the two augmentation views, to further improve the representations' robustness and prevent them from collapsing into partial or even trivial ones. We provide theoretical analysis to show that GraphCV is capable to learn minimal sufficient representations. Finally, we conduct experiments to validate the effectiveness of GraphCV over the commonly-used graph benchmark datasets. The experimental results demonstrate that GraphCV achieves significant performance gains over different datasets and settings compared with state-of-the-art baselines.

The main contributions of this work are summarized from three aspects: (i) We propose the GraphCV to alleviate the feature suppression issue with the cross-view reconstruction mechanism; (ii) We provide solid theoretical analysis on our model designs; (iii) Thorough experiments are conducted to demonstrate the robustness and transferability of the learned representations via GraphCV.

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 predict y_i related to the downstream task. We clarify the sufficiency of $\mathbf{z}_i$ as containing no less information of the label of G_i [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.

2.2 Contrastive Learning

Contrastive Learning (CL) is a self-supervised representation learning method which leverages instance-level identity for supervision. During the training phase, each graph G firstly goes through proper data augmentation to generate two data augmentation views $t_1(G)$ and $t_2(G)$, where $t_1(\cdot)$ and $t_2(\cdot)$ are two augmentation operators. Then, the CL method encourages the encoder f (a backbone network plus a projection layer) to map $t_1(x)$ 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 $t_1(G)$ and $t_2(G)$. The learning of the encoder is usually directed by a contrastive loss, such as NCE loss [44], InfoNCE loss [37] and NT-Xent loss [6]. In Graph Contrastive Learning (GCL), we usually adopt a GNN, such as GCN [20] or GIN [46], as the backbone network, and the commonly-used graph data augmentation operators [51], such as node dropping, edge perturbation, subgraph sampling, and attribute masking.

All the GCL-based methods are built on the assumption that augmentations do not break the sufficiency requirement to make correct prediction. Here, we follow [9] to clear up the definition of mutual redundancy. $t_1(G)$ is redundant to $t_2(G)$ with respect of y iff $t_1(G)$ and $t_2(G)$ share the same predictive information. Mathematically, the mutual redundancy in CL exists when:

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

Although GCL-based methods are usaully capable to extract useful information for label identification, it is unavoidable to include non-predictive features under the SSL setting owing lack of explicit domain knowledge. There exist the situation (e.g., OOD setting) that the latent space of learned representation is dominated by non-predictive features in SSL [7] and it is no more informative enough to make correct prediction. Therefore, feature suppression is not just a prevalent issue in

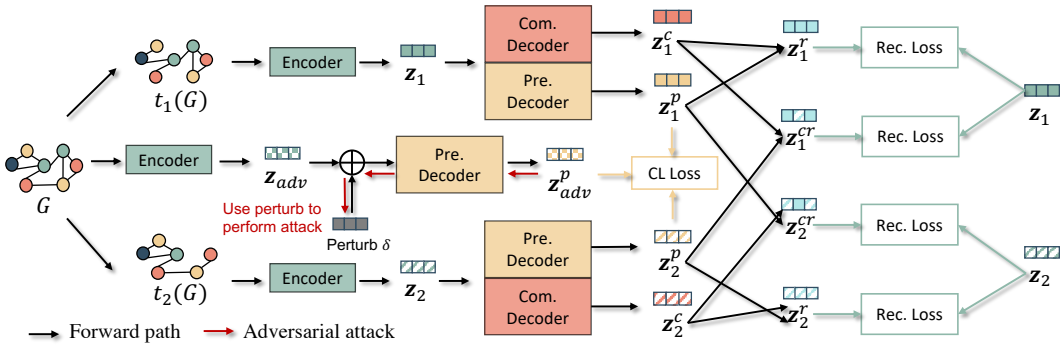


Figure 2: The illustration of the proposed GraphCV. (1) Graph augmentations are applied to the input graph G to produce two augmented graphs, which are then fed into the shared graph encoder $f(\cdot)$ to generate two graph representations $\mathbf{z}_1$ and $\mathbf{z}_2$. (2) $\mathbf{z}_1$ and $\mathbf{z}_2$ are used as the inputs of the two decoder to generate two pairs of graph representations, $\mathbf{z}^p$ captures the predictive factors and $\mathbf{z}^c$ 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 $\mathbf{z}^p_{adv}$. We take it as the third view besides $\mathbf{z}^p_1$ and $\mathbf{z}^p_2$ in CL guarantee the $\mathbf{z}^p$ can keep the global semantics.

supervised learning, but also in SSL. Due to the page limitation, we provide more discussion about the relation between feature suppression and GCL in Appendix B

3 Proposed Model

In this section, we introduce the details of our proposed GraphCV whose framework is shown in Figure 2. Corresponding theoretical analysis are provided to justify the rationality of our designs. Before diving into the details of GraphCV, we briefly introduce the overall framework of our model.

The proposed GraphCV model is designed in accordance with the IB principle to extract minimal yet sufficient representation through the designed cross-view reconstruction mechanism. Given $f(\cdot)$ as the graph encoder, we aim to map the graph representation $\mathbf{z} = f(G) \in \mathbb{R}^d$ into two different feature spaces $(\mathbf{z}^p, \mathbf{z}^c)$, where $\mathbf{z}^p \in \mathbb{R}^d$ is expected to be specific to the predictive information G^p, and $\mathbf{z}^c \in \mathbb{R}^d$ is optimized to elicit the complementary non-predictive factors G^c. Later, we reconstruct the representation $\mathbf{z}$ with the feature subsets mapped from same and different augmentation views to approximate the disentanglement objective demonstrated in Figure 1. By separating the learned representation into two sets of disentangled features and later utilizing them to reconstruct the two, we alleviate the feature suppression issue [30] at no cost of information sufficiency. We further add extra regularization to guarantee $\mathbf{z}^p$ does not collapse into shallow or partial features during the reconstruction process. We will introduce more details of GraphCV in the later contents.

3.1 Disentanglement by Cross-View Reconstruction

In GCL, we usually leverage a graph encoder, such as a GCN [20] or a GIN [46], to encode the graph data into its representation. There are multiple choices of graph encoders in GCL, including GCN [20] and GIN [46], 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 adapted to our model. Given two augmentation views $t_1(G)$ and $t_1(G)$ (where $t_1(\cdot)$ and $t_2(\cdot)$ are IID sampled from the same family of augmentation $\mathcal{T}$), we firstly use the encoder $f(\cdot)$ to map them into a lower dimension hidden space for the two embeddings, $\mathbf{z}_1$ and $\mathbf{z}_2$. Instead of directly maximizing the agreement between the two representations $\mathbf{z}_1$ and $\mathbf{z}_2$, we further feed each of them into a pair decoders (g_p, g_c) (both of them are MLP-based networks or GNN) and optimize the two decoders to map each of the presentation into the two disentangled feature sub-spaces:

$$[\mathbf{z}^p = g_p(f(t(G))),\ \mathbf{z}^c = g_c(f(t(G)))]\,, \tag{3}$$

where a pair of embeddings for both $t_1(G)$ and $t_2(G)$ are generated. Ideally, $\mathbf{z}^p_1$ and $\mathbf{z}^p_2$ suffice the mutual redundancy assumption stated in 2.2 because $t_1(G)$ and $t_1(G)$ are augmented from the same original graph, and thus naturally share the same predictive factors.

Here, we clarify the lower bound of the mutual information between one augmented view and the two mapped representations learned from the other augmented view in Theorem 1.

Theorem 1 *Suppose $f(\cdot)$ is a GNN encoder as powerful as 1-WL test. Let $\mathbf{z}_1^p$ and $\mathbf{z}_2^p$ be specific to the predictive information of G, meanwhile $\mathbf{z}_1^c$ and $\mathbf{z}_2^c$ account for the non-predictive factors of $t_1(G)$ and $t_2(G)$. Then we have:*

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

The detailed proof is provided in Appendix E. Given the lower bound, we substitute the objective by the mutual information between the two representations in the predictive view ($\mathbf{z}_1^p$ and $\mathbf{z}_2^p$) to maximize the consistency between the information of the two views. Therefore, we derive the objective function ensuring view invariance as follows:

$$\mathcal{L}_{\text{pre}} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{\text{CL}}(\mathbf{z}_{1,i}^p, \mathbf{z}_{2,i}^p), \tag{4}$$

where $\mathcal{L}_{\text{CL}}(\cdot)$ is the adopted InfoNCE loss [37]. To further pursue the feature disentanglement as illustrated in Figure 1(c), we thus propose the cross-view reconstruction mechanism. To be specific, we would like the representation pair $(\mathbf{z}^p, \mathbf{z}^c)$ within and cross the augmentation views be able to recover the raw data so that the two objectives can be approached simultaneously. Due to the fact that graphs are non-Euclidean structured data, we instead try to recover $\mathbf{z} = f(t(G))$ given ($\mathbf{z}^c$ and $\mathbf{z}^p$).

More specifically, we first perform the reconstruction within the augmentation view, namely mapping $(\mathbf{z}_w^p, \mathbf{z}_w^c)$ to $\mathbf{z}_w$, where $w \in \{1, 2\}$ representing the augmentation view. Then, we define the $(\mathbf{z}_{w'}^p, \mathbf{z}_w^c)$ as a cross-view representation pair and the reconstruction procedure is repeated on it to predict $\mathbf{z}_w$, aiming to ensure $\mathbf{z}^p$ and $\mathbf{z}^c$ is optimized to approximate mutual disentanglement, where $w = 1, w' = 2$ or $w = 2, w' = 1$. Intuitively, the reconstruction process is capable of separating the information of the shared features sets from the one resided in the unique feature sets between the two augmentation views. Since the two IID sampled augmentation operators ($t_1(\cdot)$ and $t_2(\cdot)$) are expected to preserve the predictive/rational features while varying the augmentation-related ones, we disentangle the rational features from G according to the rationale discover studies [5] to ensure the features' robustness for downstream tasks. Here, we formulate the reconstruction procedures as:

$$\mathbf{z}_w^r = g_r\left(\mathbf{z}_w^p \odot \mathbf{z}_w^c\right), \ \mathbf{z}_w^{cr} = g_r\left(\mathbf{z}_{w'}^p \odot \mathbf{z}_w^c\right), \tag{5}$$

where g_r is the parameterized reconstruction model and $\odot$ is the free-to-choose fusion operator, such as element-wise product or concatenation. The reconstruction procedures are optimized by minimizing the entropy $H\left(\mathbf{z}_w \mid \mathbf{z}_{w'}^p, \mathbf{z}_w^c\right)$, where $w = w'$ or $w \neq w'$. Ideally, we reach the optimal sufficiency and disentanglement conditions illustrated in Figure 1 (b) and (c) iff $H\left(\mathbf{z}_w \mid \mathbf{z}_{w'}^p, \mathbf{z}_w^c\right) = -\mathbb{E}_{p\left(\mathbf{z}_w, \mathbf{z}_{w'}^p, \mathbf{z}_w^c\right)}\left[\log p\left(\mathbf{z}_w \mid \mathbf{z}_{w'}^p, \mathbf{z}_w^c\right)\right] = 0$, where $\mathbf{z}_w$ is exactly recovered given its complementary representation and the predictive representation of any view. Nevertheless, the condition probability $p\left(\mathbf{z}_w \mid \mathbf{z}_{w'}^p, \mathbf{z}_w^c\right)$ is intractable, we hence use the variational distribution approximated by g_r instead, denoted as $q\left(\mathbf{z}_w \mid \mathbf{z}_{w'}^p, \mathbf{z}_w^c\right)$. We provide the upper bound of $H\left(\mathbf{z}_w \mid \mathbf{z}_{w'}^p, \mathbf{z}_w^c\right)$ in Theorem 2.

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

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

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

$$\mathcal{L}_{\text{recon}} = \frac{1}{2N} \sum_{i=1}^{N} \sum_{w=1}^{2} \left[\left\|\mathbf{z}_{w,i} - \mathbf{z}_{w,i}^r\right\|_2^2 + \left\|\mathbf{z}_{w,i} - \mathbf{z}_{w,i}^{cr}\right\|_2^2\right]. \tag{6}$$

3.2 Adversarial Contrastive View

With the cross-view reconstruction mechanism above, the two learned representations stated above are optimized towards the disentangled manner. However, it is still necessary to further prevent the learned predictive representation from focusing on the partial features, because we do not have access to the explicit domain knowledge and such small scope will increase the risk of shortcut solution. Therefore, we extend the Equation 4 to three contrastive views and add an extra global view without

topological perturbation as the third views to guarantee the learned $\mathbf{z}^p$ maintain the global semantics instead of partial or even trivial features, i.e., $\mathbf{z}_1^p \sim G$ and $\mathbf{z}_2^p \sim G$. During the experiments, we find an adversarial graph sample perturbed from original graph view can help the model achieve stronger robustness. A possible explanation is that there is still redundant information that is not predictive left in the shared information of the two $\mathbf{z}^p$'s in the two augmentation views, especially when the implemented augmentations are moderate. An adversarial view may further alleviate redundancy. We define the adversarial objective as follows:

$$\delta^* = \underset{\|\delta\|_\infty \leqslant \epsilon}{\operatorname{argmax}} \mathcal{L}_{\text{adv}} \left(t_1(G), t_2(G), G + \delta\right), \tag{7}$$

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 crafted perturbation is spurred by recent work [48] that add perturbation δ on the output of first hidden layer $\mathbf{h}^{(1)}$, since it is empirically proved to generate more challenging views than adding perturbation on the initial node feature. Therefore, the adversarial contrastive objective is defined as:

$$\mathcal{L}_{\text{adv}} = \frac{1}{N} \sum_{i=1}^{N} \max_{\delta^*} \left[\mathcal{L}_{\text{CL}} \left(\mathbf{z}_{1,i}^p, G + \delta^*\right) + \mathcal{L}_{\text{CL}} \left(\mathbf{z}_{2,i}^p, G + \delta^*\right)\right]. \tag{8}$$

where the optimized perturbation δ' is solved by projected gradient descent (PGD) [25]. Finally, we derive the joint objective of GraphCV by combining all of objectives above together. The joint objective is as follow:

$$\min_{f,g} \mathbb{E}_{G \in \mathbf{G}} \left[\mathcal{L}_{\text{pre}} + \lambda_r \mathcal{L}_{\text{recon}} + \lambda_a \max_{\|\delta\|_\infty \leqslant \epsilon} \mathcal{L}_{\text{adv}}\right], \tag{9}$$

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 1(c) with the joint objective.

4 Experiments

In this section, we demonstrate the empirical evaluation results of GraphCV on public graph benchmark datasets under different settings. Ablation study and hyper-parameter analysis are also conducted to evaluate the effectiveness of the designs in GraphCV. We further compare the robustness of GraphCV with the adversarial training-based GCL method. More content about dataset statistics, training details and other empirical analysis are provided in the Appendix.

4.1 Experimental Setups

Datasets. For unsupervised learning setting, we evaluate our model on five graph benchmark datasets from the field of bioinformatics, including MUTAG, PTC-MR, NCI1, DD, and PROTEINS, and other four from the field of social network, which are COLLAB, IMDB-B, RDT-B, and IMDB-M, for the task of graph-level property classification. For the transfer learning setting, we follow previous work [51, 47] to pretrain our model on the ZINC-2M dataset, which cotains 2 million unlabeled molecule graphs sampled from MoleculeNet [43], then evaluate its performance on eight binary classification datasets from chemistry domain, where the eight datasets are splitted according to the scaffold to simulate the out-of-distribution scenario in real-world. Additionally, We use ogbg-molhiv from Open Graph Benchmark Dataset [17] to evaluate our model over large-scale dataset under semi-supervised setting. More details about dataset statistics are included in Appendix C.

Baselines. Under the unsupervised representation learning setting, we compare GraphCV with the eight SOTA self-supervised learning methods GraphCL [51], InfoGraph[33], MVGRL [14], AD-GCL[34], GASSL[48], InfoGCL[45], RGCL [22] and DGCL[21], as well as three classical unsupervised representation learning methods, including node2vec [11], graph2vec [28], and GVAE[19]. Besides, we employ AttrMasking [18], ContextPred [18], GraphCL [51], GraphLoG [47], AD-GCL [34] and RGCL [22] as baselines to evaluate the effectiveness of our proposed GraphCV under transfer learning setting.

Evaluation Protocol. For unsupervised setting, we follow the evaluation protocols in the previous works [33, 51, 21] to verify the effectiveness of our model. The mean test accuracy score evaluated by a 10-fold cross validation with standard deviation of five random seeds as the final performance. For transfer learning setting, we follow the finetuning procedures of previous work [51, 47] and report

Table 1: Overall comparison on multiple graph classification benchmarks under unsupervised learning setting. 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	DD
node2vec	72.6±10.2	58.6±8.0	-	54.9±1.6	57.5±3.6	-	-	-	-
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	-
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	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	-
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	-	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	-
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	75.1±0.4
RGCL	87.7±1.0	-	70.9±0.7	78.1±1.1	75.0±0.4	71.9±0.8	90.3±0.6	-	78.9±0.5
GASSL	90.9±7.9	64.6±6.1	78.0±2.0	80.2±1.9	-	74.2±0.5	-	51.7±2.5	-
GraphCV	**92.3±0.7**	**67.4±1.3**	80.5±0.5	**82.0±1.0**	**76.8±0.4**	75.6±0.4	**92.4±0.9**	**52.2±0.5**	**80.5±0.5**

the mean ROC-AUC scores with standard deviation of 10 repeated runs on each downstream datasets. 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 [53]. We choose GIN [46] as the backbone graph encoder and the model is optimized through Adam optimizer. There are two specific hyperparameters 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 $\{0.0, 0.25, 0.5, 0.75, 1.0\}$, 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 learning. The overall performance comparison is shown in Table 1. From the results, we can reach to 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) RGCL, AD-GCL, and GASSL achieve better performances than GraphCL, which empirically proves the conclusion that InfoMax object could bring overwhelmed redundant information and thus suffer from feature suppression issue; (3) Our proposed GraphCV and DGCL consistently outperform other baselines, proving the advantage of disentangled representation. More importantly, GraphCV achieves state-of-the-art results on most of the datasets, demonstrating the model effectiveness.

Semi-supervised learning. The experimental results are shown in Figure 3. 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 more trainable data could bring more redundant information, thereby further deteriorate the feature suppression issue. Therefore, removing redundant information causes a higher performance boost.

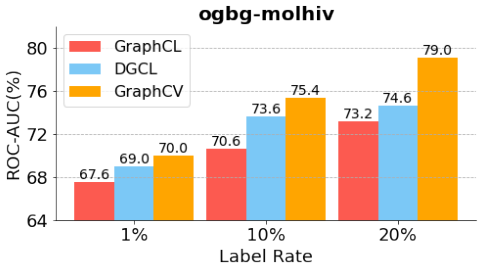


Figure 3: Performance comparison of semi-supervised learning on ogbg-molhiv.

Transfer learning. Table 2 demonstrates the experimental results under transfer learning setting. Except for the baselines mentioned above, we also include No Pre-Train as the baseline (i.e., the self-supervised pre-training on the ZINC-2M dataset is skipped and we randomly initialze the GIN backbone before finetune). It is noteworthy that some strong baselines (AttrMasking and ContextPred) are trained under the guidance of domain knowledge. Despite in lacking of such domain knowledge, our model still outperforms all the other baselines on 3 out 8 datasets and achieve highest average performance. More importantly, JOAO, RGCL and our proposed GraphCV are all developed from GraphCL, but achieve higher average performance than

Table 2: Overall comparison on multiple graph classification benchmarks under transfer learning setting. Results are reported as mean±std%, the best performance is bolded and runner-ups are underlined.

	BBBP	Tox21	ToxCast	SIDER	ClinTox	MUV	HIV	BACE	Avg
No Pre-Train	65.8±4.5	74.0±0.8	63.4 ±0.6	57.3±1.6	58.0±4.4	71.8±2.5	75.3±1.9	70.1±5.4	67.0
AttrMasking	64.3±2.8	**76.7±0.4**	**64.2±0.5**	61.0±0.7	71.8±4.1	74.7±1.4	77.2±1.1	79.3±1.6	71.1
ContextPred	68.0±2.0	75.7±0.7	63.9±0.6	60.9±0.6	65.9±3.8	75.8±1.7	77.3±1.0	79.6±1.2	70.9
GraphCL	69.5±0.5	75.4±0.9	63.8±0.4	60.8±0.7	70.1±1.9	74.5±1.3	77.6±0.9	78.2±1.2	70.8
GraphLoG	**72.5±0.8**	75.7±0.5	63.5±0.7	61.2±1.1	76.7±3.3	76.0±1.1	77.8±0.8	**83.5±1.2**	73.4
JOAO	70.2±1.0	75.0±0.3	62.9±0.5	60.0±0.8	81.3±2.5	71.7±1.4	76.7±1.2	51.5±0.4	71.9
RGCL	71.4±0.7	75.2±0.3	63.3±0.2	61.4±0.6	83.4 ±0.9	**76.7 ±1.0**	77.9±0.8	76.03±0.8	73.2
GraphCV	71.6±0.6	75.7±0.6	63.2±0.5	**62.2±0.7**	**83.6±1.5**	76.4±0.8	**77.9±1.0**	80.8±1.8	**73.9**

GraphCL. This observation further empirically prove the poisoning effect of biased information and the necessity to to suppress them.

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 two model variants: (1) **w/o CV Recon**, the cross-view reconstruction process is discarded; (2) **w/o Adv. Training**, the third adversarial view is discarded. The comparison results are shown in Table 3. We can observe from Table 3 that our model with the combination of cross-view reconstruction and adversarial training module outperforms all of the variants. Omitting the reconstruction process could cause the failure to optimize the representation in a disentangled manner illustrated in Figure 1(c), thereby the learned representation still suffer from features suppression issue. Compared with our model, the variant w/o Adv. Training may lead to representation collapse and bring extra redundant information, therefore resulting in sub-optimal performances in downstream tasks.

Table 3: 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	DD
w/o CV 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	51.7±0.6	79.0±0.8
w/o Adv. Training	92.1±0.6	66.8±0.5	76.5±0.6	81.2±0.9	76.0±0.3	75.1±0.6	92.2±1.0	50.8±0.4	80.1±0.6
GraphCV	**92.3±0.7**	**67.4±0.5**	**80.5±0.5**	**82.0±1.0**	**76.8±0.4**	**75.6±0.4**	**92.5±0.9**	**52.2±0.5**	**80.5±0.5**

4.4 Robustness and Hyper-parameter Analysis

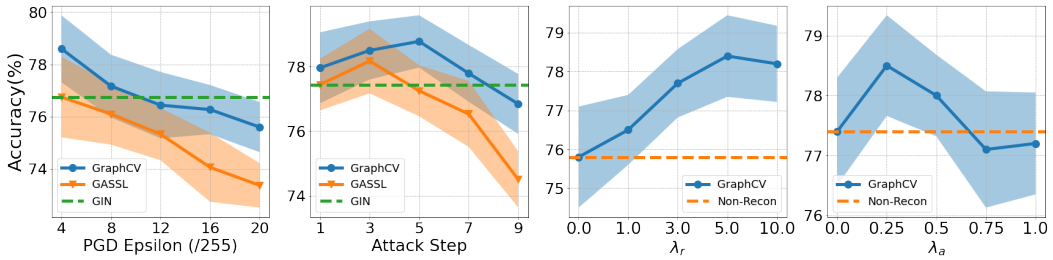


Figure 4: The model performance under different perturbation bound, attack step and analysis the sensitivity of the two important hyper-parameters (i.e., λ_r and λ_a).

In this section, we firstly conduct extra experiments on ogbg-molhiv dataset to evaluate the representation robustness under aggressive augmentation and perturbation. The results are shown in left two subplots of Figure 4, 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. In the right two subplots,

we analysis the model sensitivity of the two important hyper-parameters in our model, λ_r and λ_a. The consistent superiority of different values over the initial point (i.e., λ_r, $\lambda_a = 0$) prove the effectiveness of our design once again. We can also observe that the appropriate range of the two hyper-parameters are 5.0 to 10.0 and 0.0 to 0.5, respectively. Depend on the datasets size and attributes, the range can have some variance, we suggest to finetune the two hyper-parameters around 10.0 and 0.25 to find the appropriate values when adopting our model to a new datasets. More experiments and discussion about hyper-parameter sensitivity in provided in the Appendix G. Besides, we also conduct extra experiments to analyze the disentanglement of $\mathbf{z}^p$ and $\mathbf{z}^c$ in Appendix F.

5 Related Work

Graph contrastive learning. Contrastive learning is firstly proposed in the compute vision field [6] 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 [39, 29, 15, 51, 33] 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 [34, 45, 50] or adding adversarial perturbation [48]. 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 [16, 52] to disentangle the heterogeneous latent factors of the representations, and therefore making the representations more robust and interpretable [3]. This idea has now been widely adopted in graph representation learning. [23, 24] utilizes neighborhood routing mechanism to identify the latent factors in the node representations. Some other generative models [19, 32] utilize Variational Autoencoders to balance reconstruction and disentanglement. Recent work [21] 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) [35] 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 [2, 31, 9] 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 [41, 45, 34, 22] 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 the feature suppression problem in representation learning. To avoid the predictive features being suppressed in learned representation, we propose a novel model, namely GraphCV, which is designed in accordance with the information bottleneck principle. The cross-view reconstruction in GraphCV can disentangle those more robust and transferable features from those easily-disturbed ones. Meanwhile, we also add an adversarial view as the third view of the contrastive learning to to guarantee the global semantics and further 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 transferable 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] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep Variational Information Bottleneck. *ICLR*, 2017.

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

[4] Beatrice Bevilacqua, Yangze Zhou, and Bruno Ribeiro. Size-invariant graph representations for graph classification extrapolations. In *ICML*, 2021.

[5] Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. Invariant rationalization. In *ICML*, 2020.

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

[7] Ting Chen, Calvin Luo, and Lala Li. Intriguing properties of contrastive losses. In *NeurIPS*, 2021.

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

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

[10] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2020.

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

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

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

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

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

[16] 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.

[17] 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.

[18] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *ICLR*, 2020.

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

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

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

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

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

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

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

[26] Matthias Minderer, Olivier Bachem, Neil Houlsby, and Michael Tschannen. Automatic shortcut removal for self-supervised representation learning. In *ICML*, 2020.

[27] Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. TUDataset: A collection of benchmark datasets for learning with graphs. Technical report, arXiv, 2020.

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

[29] 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.

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

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

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

[33] 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.

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

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

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

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

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

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

[40] Qitian Wu, Hengrui Zhang, Junchi Yan, and David Wipf. Handling distribution shifts on graphs: An invariance perspective. In *ICLR*, 2022.

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

[42] Ying-Xin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat seng Chua. Discovering invariant rationales for graph neural networks. In *ICLR*, 2022.

[43] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

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

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

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

[47] Minghao Xu, Hang Wang, Bingbing Ni, Hongyu Guo, and Jian Tang. Self-supervised graph-level representation learning with local and global structure. In *ICML*, 2021.

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

[49] Haotian Ye, Chuanlong Xie, Tianle Cai, Ruichen Li, Zhenguo Li, and Liwei Wang. Towards a theoretical framework of out-of-distribution generalization. In *NeurIPS*, 2021.

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

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

[52] 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.

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

A Out-of-distribution Scenario on Graph

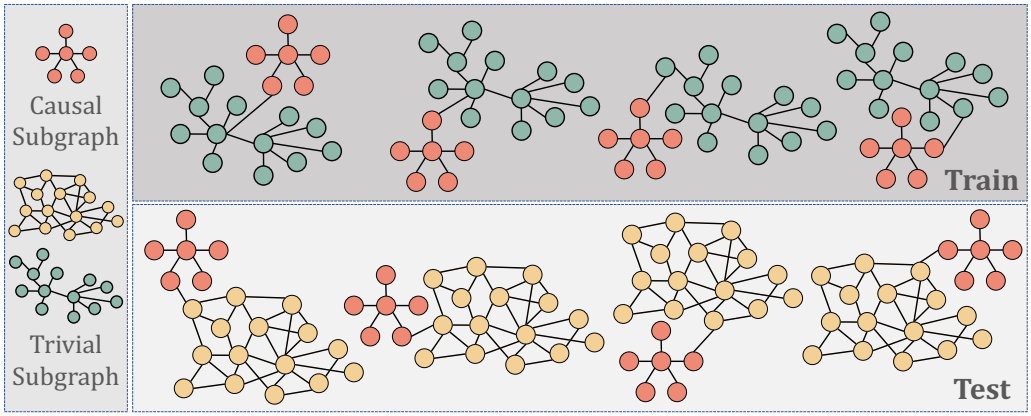


Figure 5: An out-of-distribution situation in molecule graph prediction task. The casual functional sub-structure (red) are spuriously correlated with different trivial sub-structures in training and test set. The statistical correlations can lead to poor robustness and transferability.

In this section, we will illustrate the out-of-distribution scenario in graph learning task. During molecule property study, A specific kind of property (e.g., toxicity and lipophilicity) of a molecule is usually dependent on if it has corresponding sub-structures (termed as functional group). For example, hydrophilic molecules usually have oxhydryl group ($-OH$) Therefore, a well-trained GNN model on molecule graph prediction task is capable to reflect the sub-structure information in

the graph representation. However, it is usually the case in real-world scenario that the predictive functional group is usually accompanied by some irrelevant groups in some environments, thus causing spurious correlations. This correlation usually lead to poor generalization performance when the model is evaluated on another environment with different spurious correlation. Figure 5 intuitively demonstrates this kind of scenario, where the red subgraph is the feature we can rely on to make casual prediction. But it usually show up with green subgraph that do not serve as the functional graph of the property in training set. Consequently, the model are easily misguided that the green subgraph is an important indicator of the property. When we evaluate the model on the testing set where the casual graph is correlated with another kind of group (yellow subgraph), there usually exists a huge gap between its performances on the two sets.

B Discussion on Feature Suppression

In this section, we will follow the previous works [7, 30] to present a more formal definition of the feature suppression and clarify its relation with contrastive learning.

First of all, we assume graph data G has n feature sub-spaces, $G^1, \ldots, G^n$, where each $G^i \in G$ corresponding to a distinct feature of G. To quantify the relation between G and its feature sub-spaces, we need to measure the conditional probability of G given a specific kind of feature sub-space G^i ($i \subseteq [n]$), denoted as $p(G \mid G^i)$. Finally, we define a an injective map $g : G^i \to G$ produce observation $G = g(G^i)$. Due to the reason that G^i is not explicit, so we aim to train a encoder $f : G_i \to \mathbb{R}^d$ to map input graph data G into a latent space to extract useful high-level information $\mathbf{z}^i$ corresponding to each feature sub-space G^i of input data G during cotrastive learning. Therefore, we use $p(G \mid \mathbf{z}^i)$ as the approximated value of the measurement $p(G \mid G^i)$. Then we have,

- For any feature sub-space G^i and its complementary feature sub-subspace $G^{\bar{i}}$, f suppress feature $i \subseteq [n]$ if we have $p(G \mid \mathbf{z}^i) = p(G \mid \mathbf{z}^{\bar{i}})$
- For any feature sub-space G^i and its complementary feature sub-subspace $G^{\bar{i}}$, f distinguish feature $i \subseteq [n]$ if $p(G \mid \mathbf{z}^i)$ and $p(G \mid \mathbf{z}^{\bar{i}})$ have disjoint support.

To sum up, a feature is suppressed if it does not make any difference to the instance discrimination. One of the common acknowledgements for unsuprevised learning strategy is that it can usually produce representation with uniform feature space distribution due to the lack of supervision, i.e., every feature sub-space is equally treated without feature suppression. However, it could not be the situation in contrastive learning. Taking the commonly used InfoNCE][37] as an example, it can be divided into two parts, i.e. align term and uniform term [6], as follow:

$$\tau \mathcal{L}^{\text{InfoNCE}} = \underbrace{-\frac{1}{m}\sum_{i,j}\text{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_j\right)}_{\mathcal{L}_{\text{alignment}}} + \underbrace{\frac{\tau}{m}\sum_{i}\log\sum_{k=1}^{2m}\mathbf{1}_{[k\neq i]}\exp\left(\text{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_k\right)/\tau\right)}_{\mathcal{L}_{\text{uniform}}} \tag{10}$$

Aligning the positive pair will distinguish the shared feature subspace G^i. Meanwhile, there also exits random negative samples that might own same factors in G^i, so the uniform term might suppress the feature sub-space G^i. Therefore, for any feature $i \subseteq [n]$, the optimization process can either suppress or distinguish it, but both of them can reach to lower contrastive loss. From the analysis we can derive the conclusion mentioned in Section 1 that lower contrastive loss might not yield better performance.

C Summary of Datasets

In this work, we use nine datasets from TU Benchmark Datasets [27] to evaluate our proposed GraphCV under unsupervised setting, where five of them are biochemical datasets and the other four belong to social network datasets. We also utilize the ogng-molhiv dataset from Open Graph Benchmark (OGB) [17] to further evaluate GraphCV under semi-supervised setting. Besides, the datasets sampled from MoleculeNet [43] are employed to evaluate our model under transfer learning setting. The statistics of these datasets are shown in Table 4 and 5.

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

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
ogbg-molhiv	41,127	25.50	27.50	2	ROC-AUC	MoleculeNet

Table 4: Statistics of TU-datasets and OGB dataset.

Dataset	#Graphs	Avg #Nodes	Avg Degree	#Tasks	Metric	Category
ZINC-2M	2,000,000	26.62	57.72	-	-	biochemical
BBBP	2,039	24.06	51.90	1	ROC-AUC	biochemical
Tox21	7,813	18.57	38.58	12	ROC-AUC	biochemical
ToxCast	8,576	18.78	38.62	617	ROC-AUC	biochemical
SIDER	1,427	33.64	70.71	27	ROC-AUC	biochemical
ClinTox	1,477	26.15	55.76	2	ROC-AUC	biochemical
MUV	93,087	24.23	52.55	17	ROC-AUC	biochemical
HIV	41,127	25.51	54.93	1	ROC-AUC	biochemical
BACE	1,513	34.08	73.71	1	ROC-AUC	biochemical

Table 5: Statistics of MoleculeNet datasets.

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

D 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 [46] as the backbone graph encoder and the model is optimized through Adam optimizer. We follow [51, 48, 21] to employ a linear SVM classifier for downstream task-specific classification. The graph augmentation operations used in our work are same as [51], including node dropping, edge perturbation, attribute masking and subgraph sampling, all of them are borrowed from the implementation of [53]. 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 $\{0.0, 0.25, 0.5, 0.75, 1.0\}$, respectively. For other important hyper-parameters, we find the best value of 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. As for the implementation details of transfer learning, we basically follow the pre-training setting of previous works [51, 47].

E Proof

E.1 Proof of Theorem 1

We repeat Theorem 1 as follows.

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

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

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

$$\begin{aligned} p\left(\mathbf{z}^p, \mathbf{z}^c\right) &= p\left(\mathbf{z}^p\right) p\left(\mathbf{z}^c\right) \\ p\left(\mathbf{z}^p, \mathbf{z}^c \mid t(G)\right) &= p\left(\mathbf{z}^p \mid t(G)\right) p\left(\mathbf{z}^c \mid t(G)\right). \end{aligned} \tag{11}$$

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,

$$\begin{aligned} & I(a; b \mid c) = \\ &= \sum_a \sum_b \sum_c p(a, b, c) \log \frac{p(a, b, c)\, p(c)}{p(a, c)\, p(b, c)} \\ &= \sum_a \sum_b \sum_c p(a)\, p(b, c \mid a) \log \frac{p(b, c \mid a)\, p(c)}{p(c \mid a)\, p(b)\, p(c)} \\ &= \sum_a \sum_b \sum_c p(a)\, p(b \mid a)\, p(c \mid a) \log \frac{p(b \mid a)\, p(c \mid a)}{p(c \mid a)\, p(b)} \\ &= \sum_a \sum_b p(a)\, p(b \mid a) \log \frac{p(b \mid a)}{p(b)} \\ &= \sum_a \sum_b p(a, b) \log \frac{p(b \mid a)}{p(b)} \\ &= I(a; b). \end{aligned} \tag{12}$$

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

$$\begin{aligned} I\left(t_1(G); \mathbf{z}_2^p, \mathbf{z}_2^c\right) &= I\left(t_1(G); \mathbf{z}_2^p \mid \mathbf{z}_2^c\right) + I\left(t_1(G); \mathbf{z}_2^c\right) \\ &\overset{(2)}{=} I\left(t_1(G); \mathbf{z}_2^p\right) + I\left(t_1(G); \mathbf{z}_2^c\right) \\ &\overset{(a)}{\geqslant} I\left(t_1(G); \mathbf{z}_2^p\right) \\ &\overset{(b)}{\geqslant} I\left(\mathbf{z}_1^c, \mathbf{z}_1^p; \mathbf{z}_2^p\right) \\ &\overset{(2)}{=} I\left(\mathbf{z}_1^c; \mathbf{z}_2^p\right) + I\left(\mathbf{z}_1^p; \mathbf{z}_2^p\right) \\ &\overset{(a)}{\geqslant} I\left(\mathbf{z}_1^p; \mathbf{z}_2^p\right), \end{aligned} \tag{13}$$

where $\overset{(2)}{=}$ is derived from the conclusion we get in Equation 12, $\overset{(a)}{\geqslant}$ is based on the non-negativity of mutual information, i.e., $I(;) \geqslant 0$, and $\overset{(b)}{\geqslant}$ is because data processing inequality [8]. Finally, we reach to the lower bound of $I\left(t_1(G); \mathbf{z}_2^p, \mathbf{z}_2^c\right)$ in Equation 12, thus we can maximize the consistency between the information we capture from the two augmentation graph views by minimizing $\mathcal{L}_{\text{pre}}$.

E.2 Proof of Theorem 2

We repeat Theorem 2 as follows.

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

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

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

$$H\left(\mathbf{z}_w \mid \mathbf{z}^p_{w'}, \mathbf{z}^c_w\right) = -\mathbb{E}_{p\left(\mathbf{z}_w, \mathbf{z}^p_{w'}, \mathbf{z}^c_w\right)}\left[\log p\left(\mathbf{z}_w \mid \mathbf{z}^p_{w'}, \mathbf{z}^c_w\right)\right]. \tag{14}$$

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

$$\begin{aligned}
&\mathbb{E}_{p\left(\mathbf{z}_w, \mathbf{z}^p_{w'}, \mathbf{z}^c_w\right)}\left[\log p\left(\mathbf{z}_w \mid \mathbf{z}^p_{w'}, \mathbf{z}^c_w\right)\right] = \\
&\qquad \mathbb{E}_{p\left(\mathbf{z}_w, \mathbf{z}^p_{w'}, \mathbf{z}^c_w\right)}\left[\log q\left(\mathbf{z}_w \mid \mathbf{z}^p_{w'}, \mathbf{z}^c_w\right)\right] \\
&\qquad + D_{\mathrm{KL}}\left(p\left(\mathbf{z}_w \mid \mathbf{z}^p_{w'}, \mathbf{z}^c_w\right) \| q\left(\mathbf{z}_w \mid \mathbf{z}^p_{w'}, \mathbf{z}^c_w\right)\right).
\end{aligned} \tag{15}$$

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

$$\begin{aligned}
H\left(\mathbf{z}_w \mid \mathbf{z}^p_{w'}, \mathbf{z}^c_w\right) &\leqslant -\mathbb{E}_{p\left(\mathbf{z}_w, \mathbf{z}^p_{w'}, \mathbf{z}^c_w\right)}\left[\log q\left(\mathbf{z}_w \mid \mathbf{z}^p_{w'}, \mathbf{z}^c_w\right)\right] \\
&= -\mathbb{E}_{p\left(\mathbf{z}_w, \mathbf{z}^p_{w'}, \mathbf{z}^c_w\right)}\left[\log\left(\frac{1}{\sqrt{2\pi I}\sigma} e^{-\frac{1}{2}\frac{\left(\mathbf{z}_w - g_r\left(\mathbf{z}^p_{w'} \odot \mathbf{z}^c_w\right)\right)^2}{(\sigma^2\mathbf{I})}}\right)\right] \\
&= -\mathbb{E}_{p\left(\mathbf{z}_w, \mathbf{z}^p_{w'}, \mathbf{z}^c_w\right)}\left[\log\left(\frac{1}{\sqrt{2\pi I}\sigma}\right) - \frac{\left(\mathbf{z}_w - g_r\left(\mathbf{z}^p_{w'} \odot \mathbf{z}^c_w\right)\right)^2}{2\sigma^2\mathbf{I}}\right].
\end{aligned} \tag{16}$$

Hence, we get the upper bound of $H\left(\mathbf{z}_w \mid \mathbf{z}^p_{w'}, \mathbf{z}^c_w\right)$ as Equation 16. 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,

$$\min \mathbb{E}_{p\left(\mathbf{z}_w, \mathbf{z}^p_{w'}, \mathbf{z}^c_w\right)}\left\|\mathbf{z}_w - g_r\left(\mathbf{z}^p_{w'} \odot \mathbf{z}^c_w\right)\right\|_2^2. \tag{17}$$

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 $\mathbf{z}^p$ and $\mathbf{z}^c$.

F Effects of Representation Disentanglement

In this section, we set experiments to investigate the representation disentanglement of our proposed GraphCV. Specifically, we use the InfoNCE loss [37] 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 $\mathbf{z}^p_1$ and $\mathbf{z}^p_2$ and orange lines represent the InfoNCE loss between $\mathbf{z}^c_1$ and $\mathbf{z}^c_2$. 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

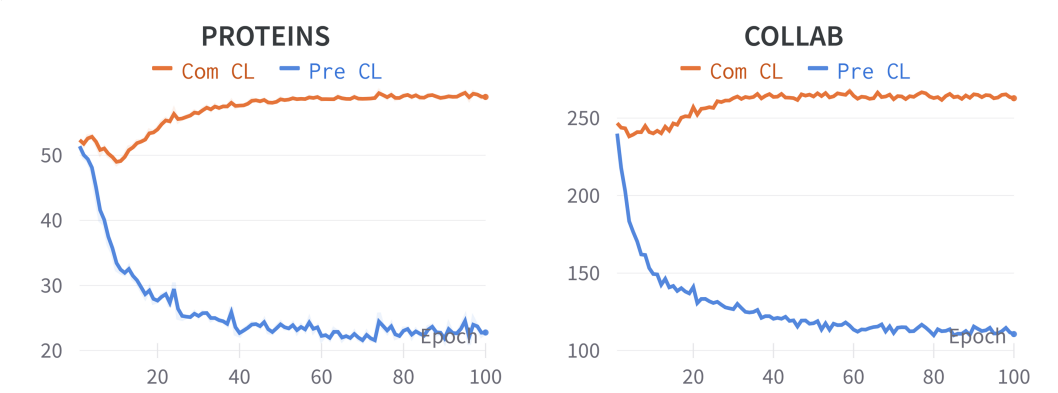


Figure 6: InfoNCE loss of the two disentangled representations between the two augmentation graph views, where orange lines are the InfoNCE loss between the two non-predictive representations and blue lines are the InfoNCE loss between the two predictive representaitons

Table 6: Performance comparison of the two learned representations. Results are reported as mean±std%, the best performance is bolded.

	MUTAG	COLLAB	NCI1	PROTEINS	IMDB-B	RDT-B	DD	ogbg-molhiv
$\mathbf{z}^c$	88.1±1.2	75.1±0.7	72.2±2.0	73.5±0.8	71.8±0.9	89.4±1.0	75.8±0.6	69.7.0±2.8
$\mathbf{z}^p$	**92.3±0.7**	**80.5±0.5**	**82.0±1.0**	**76.8±0.4**	**75.6±0.4**	**92.5±0.9**	**80.5±0.5**	**75.36±1.4**

two augmentation views. To further investigate whether the feature suppression problem is equally serious in $\mathbf{z}^p$ and $\mathbf{z}^c$, we conduct experiments to compare the performance of the two representation on downstream tasks. The comparison results are as follow:

It is easily to observe that there is a obvious performance gap between the two learned representation, indicating the different feature suppression issue between them and the features subset that are more robust to augmentation is more informative and transferable that those sensitive to augmentations. Therefore, we believe our proposed GraphCV can further alleviate the feature suppression issue with the disentanglement design.

G 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.

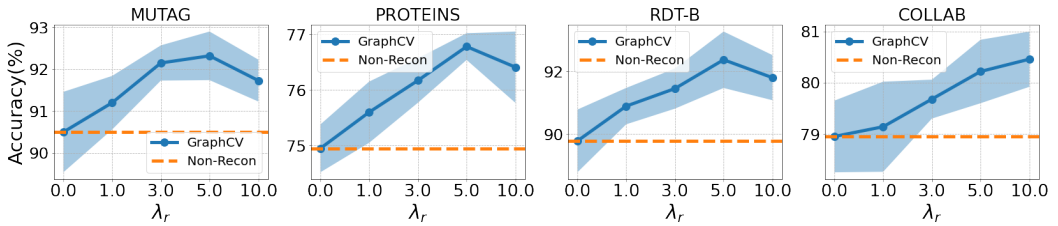


Figure 7: Impact of reconstruction loss coefficient λ_r on different datasets, we specify the non-reconstruction situation ($\lambda_r = 0$) with the dashed line for comparison.

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.

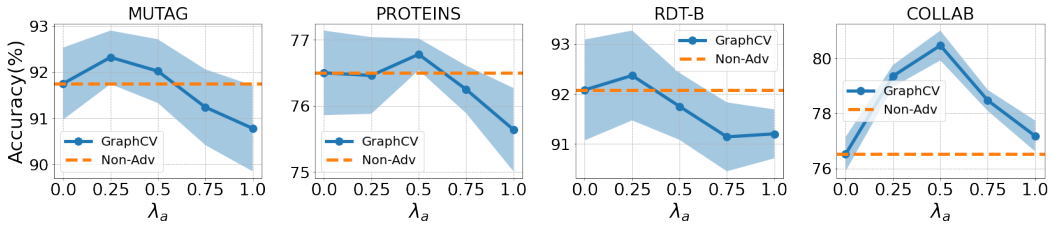


Figure 8: Impact of adversarial loss coefficient λ_a on different datasets, we specify the non-adversarial situation ($\lambda_a = 0$) with the dashed line for comparison.

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.

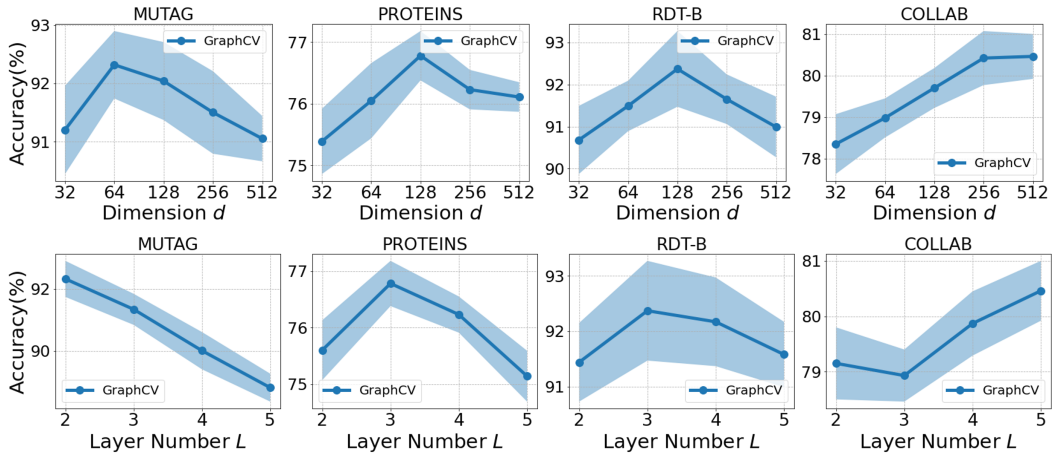


Figure 9: Impact of a embedding dimension d and GNN layer number L on different datasets.

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.

H Training Algorithm

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

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 $\mathbf{Z}^p = \{\mathbf{z}_i^p\}_{i=1}^{N}$

for *each training epoch* **do**
  for *sampled minibatch* $\mathcal{B} = \{G_i\}_{i=1}^{|\mathcal{B}|}$ **do**
    for $G_i \in \mathcal{B}$ **do**
      $\mathbf{z}_{1,i} = f\left(t_1(G_i)\right), \mathbf{z}_{2,i} = f\left(t_2(G_i)\right)$; $\triangleright\, t_1(\cdot), t_2(\cdot) \in \mathcal{T}$
      $\mathbf{z}_{1,i}^p = g_p\left(\mathbf{z}_{1,i}\right), \mathbf{z}_{2,i}^p = g_p\left(\mathbf{z}_{1,i}\right)$;
      $\mathbf{z}_{1,i}^c = g_c\left(\mathbf{z}_{1,i}\right), \mathbf{z}_{2,i}^c = g_c\left(\mathbf{z}_{1,i}\right)$;
    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 $\mathbf{Z}^p = \{\mathbf{z}_i^p\}_{i=1}^{N}$, where $\mathbf{z}_i^p = g_p\left(f(G_i)\right)$