Graph Contrastive Invariant Learning from the Causal Perspective

Yanhu Mo[1], Xiao Wang[2*], Shaohua Fan[3,4], Chuan Shi[1*]

[1]Beijing University of Posts and Telecommunications
[2] Beihang University
[3] Tsinghua University
[4] Key Laboratory of Big Data & Artificial Intelligence in Transportation, Ministry of Education(Beijing Jiaotong University)
{moyanhu, shichuan}@bupt.edu.cn, xiao_wang@buaa.edu.cn, fanshaohua@bupt.cn

Abstract

Graph contrastive learning (GCL), learning the node representation by contrasting two augmented graphs in a self-supervised way, has attracted considerable attention. GCL is usually believed to learn the invariant representation. However, *does this understanding always hold in practice?* In this paper, we first study GCL from the perspective of causality. By analyzing GCL with the structural causal model (SCM), we discover that traditional GCL may not well learn the invariant representations due to the non-causal information contained in the graph. *How can we fix it and encourage the current GCL to learn better invariant representations?* The SCM offers two requirements and motives us to propose a novel GCL method. Particularly, we introduce the spectral graph augmentation to simulate the intervention upon non-causal factors. Then we design the invariance objective and independence objective to better capture the causal factors. Specifically, (i) the invariance objective encourages the encoder to capture the invariant information contained in causal variables, and (ii) the independence objective aims to reduce the influence of confounders on the causal variables. Experimental results demonstrate the effectiveness of our approach on node classification tasks.

1 Introduction

Graph Neural Networks (GNNs) learn node representations by aggregating information from neighborhoods, which have received a great deal of attention and achieved competitive performance on various tasks over the past few years (Kipf and Welling 2017; Velickovic et al. 2018; Hamilton, Ying, and Leskovec 2017). Despite the great success, most GNNs are trained with node labels, while it is well known that manual annotations are expensive and hard to collect; therefore, self-supervised learning (SSL) has gained popularity due to its competitive performance and label-free setting (Chen et al. 2020; He et al. 2020). Graph Contrastive Learning (GCL), as one of the most successful strategies for self-supervised representation learning on graphs, has shown state-of-the-art performance on many downstream tasks (Velickovic et al. 2019; Zhu et al. 2020; Qiu et al. 2020).

The typical GCL method mainly includes three parts: graph augmentation, encoding architecture, and contrastive

loss. Most existing GCL methods learn the representations by comparing the augmentations. First, GCL generates two augmented graphs from the original graph based on some augmentation strategies, e.g., dropping edges (Velickovic et al. 2019). Then the two augmented graphs are fed into the encoding architecture (e.g., GCN (Kipf and Welling 2017)) to learn the node representations. Finally, the contrastive loss (e.g., InfoNCE (Zhu et al. 2020)) is used to train the GCL model by making the representations of positive pair in two augmented graphs similar and representations of negative pairs dissimilar. It is believed that GCL is able to learn the invariant representations by contrasting the positive and negative pairs (Zhu et al. 2021; Liu et al. 2022a). The learned invariant representations will be beneficial for the downstream tasks.

Generically, the question we want to ask is: *does GCL always possess the invariant representation ability in practice? When will it fail and how to enhance this ability?* A well-informed answer can provide a deeper understanding of the learning mechanism of GCL, identify the weakness of GCL, and motivate more powerful GCL to be proposed. The invariant representation usually represents the essential information, which can also be considered as some kind of causal variables in a graph (Arjovsky et al. 2019; Wu et al. 2022; Fan et al. 2023a). This natural and intuitive connection inspires us to explore the GCL from the causal perspective. We start with a causal analysis of GCL (more details are in Section 3) based on the structural causal model (SCM). The SCM indicates that if there are both causal and non-causal factors in a graph, only when the causal factors are the same and the non-causal factors are different in the original and two augmented graphs, GCL is able to learn the invariant causal factors. However, with the graph structures change by graph augmentation strategies, e.g., random augmentation, it is very hard to guarantee that and finally weakens the invariant representation learning ability.

Once the weakness is identified, a natural question is *how can we fix it and improve the invariant learning ability?* Still, SCM offers the following two requirements for GCL: one is on the augmentation mechanism, i.e., a better graph augmentation strategy should take the causal and non-causal factors into account. Without distinguishing the two factors, the representations obtained by GCL methods may contain both causal and non-causal information, which might weaken the performance on downstream tasks. The second is on the

---

learning mechanism, i.e., given causal and non-causal factors, how to ensure that GCL is able to learn the causal variables. This is extremely challenging because there is usually no prior knowledge of labels, etc. Moreover, even for different causal variables, they may be also statistically dependent due to the backdoor path caused by the confounders.

In this paper, we propose a novel graph contrastive learning method from the standpoint of causality, called GCIL[1] (**G**raph **C**ontrastive **I**nvariant **L**earning). Specifically, we first elaborate the structural causal model (SCM) to describe the graph generation process. According to the SCM, we generate two views from the perspective of the graph spectrum, which simulate perturbing the non-causal factors while keeping the causal contents unchanged. We assume each dimension in the representations holds the Gaussian distribution, and then we propose an invariance objective to guarantee that the representations generated from two views maintain the same mean and standard deviation for each dimension. Therefore, we can extract the causal information from the original graph. Finally, we utilize the independent module to push the causal factors mutually independent, thus none of them influence others by the backdoor path. Our contributions are summarized as follows:

- We study graph contrastive learning from the perspective of causality, and point out that existing methods may not learn the invariant representations due to the non-causal information contained in the graph.

- Based on the theory of causality, we propose a novel graph contrastive learning method that aims to learn an invariant representation via extracting the causal information contained in the original graph.

- We validate the effectiveness of GCIL compared with state-of-the-art methods on four datasets, and our method outperforms both semi-supervised and self-supervised baselines.
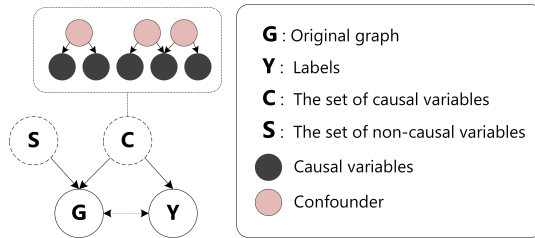

Figure 1: SCM of the graph generation process. The dashed circle and solid circle represent unobserved and observed variables, respectively.

2   Related Work

Graph Neural Networks. Graph Neural Networks (GNNs) show outstanding performance in various tasks recently. For example, GCN (Kipf and Welling 2017) averages the information of one-hop neighbors. GAT (Velickovic et al. 2018) assigns different weights to different neighbors. GraphSAGE

---

[1]https://github.com/BUPT-GAMMA/GCIL

(Hamilton, Ying, and Leskovec 2017) aggregates a subset of neighbors using various pooling methods. In the Bruna et al. (2014), fourier bias is used to decompose graph signals; ChebNet (Defferrard, Bresson, and Vandergheynst 2016) improves efficiency with the Chebyshev expansion of the graph Laplacian. Recent surveys (Wu et al. 2020; Zhou et al. 2020) provide a more complete review.

Graph Contrastive Learning. Self-supervised representation learning has attracted considerable attention in computer vision (Oord, Li, and Vinyals 2018; Chen et al. 2020; He et al. 2020). Motivated by the local-global mutual information maximization, DGI (Velickovic et al. 2019) contrasts local node embedding and global summary vector. MVGRL (Hassani and Khasahmadi 2020) employs diffusion or distance matrices to handle contrastive tasks. GRACE (Zhu et al. 2020), GraphCL (You et al. 2020), GCA (Zhu et al. 2021) obtain two views by perturbing the original graph, and then they learn representations using InfoNCE loss. ProGCL (Xia et al. 2022) use of EM algorithm to samples more appropriate hard negative samples to learn node embeddings. CCA-SSG (Zhang et al. 2021) optimizes a feature-level objective rather than discriminating positive and negative samples. Recent advances in graph contrastive learning have been summarized in several surveys (Liu et al. 2022b; Xie et al. 2022).

Causality in Graph Neural Networks. Causality studies the relationship between variables(Pearl, Glymour, and Jewell 2016; Pearl 2009), which has shown many benefits in deep learning. With the help of causality, many methods have achieved great success in various computer vision tasks (Zhang et al. 2020; Mitrovic et al. 2020). There is some research related to the graph. For example, DIR-GNN (Wu et al. 2022) conducts the intervention on the non-causal part to generate the representations to discover the rationale on the graph. DisC (Fan et al. 2022a) disentangles the graph into causal and bias subgraphs, thus alleviating the bias in the datasets. RGCL (Li et al. 2022) introduces an invariance look into self-supervised learning and proposes a method to preserve stable semantic information. Fan et al. (2022b) explores agnostic label selection bias in GNNs. CIGA (Chen et al. 2022) guarantees OOD generalization across distribution shifts by capturing the invariance of graphs. GraphNOTEARS (Fan et al. 2023b) studies the associated mechanism for generating node features in dynamic graph data. Different from them, we exploit the self-supervised node classification task from the standpoint of causality and propose a new graph contrastive learning loss based on the theory of causality.

3   Causal Analysis on GCL

Notations and Framework

Notations. Let $G = (V, E)$ denote a graph, where V is the set of nodes with $|V| = N$ and E is the set of edges. Each graph G has an adjacency matrix $A^{N \times N}$, where A_{ij} denotes the relation between two nodes, i.e., $A_{ij} = 1$ iff there is an edge between v_i and v_j. Graph G is often assigned with a node feature matrix $X = [x_1, x_2, ..., x_N]^T \in R^{N \times F}$, where x_i is the feature vector of node v_i. The goal of GCL is to learn an optimal GNN encoder to acquire node representations without requiring any label information.

Graph Contrastive Learning (GCL). Given graph G, the pipeline of GCL is to generate two graph views V_A, V_B with graph augmentation generators τ_A, τ_B as follows:

$$V_A = \tau_A(G, X), V_B = \tau_B(G, X). \qquad (1)$$

Then, the two graph views are fed into a shared GNN encoder $g(\cdot)$ to generate the node representations Z^A, Z^B, respectively:

$$Z^A = g(V_A), Z^B = g(V_B), \qquad (2)$$

where $Z^A = [z_1^A, z_2^A, ..., z_N^A]^T \in R^{N \times d}$ and $Z^B = [z_1^B, z_2^B, ..., z_N^B]^T \in R^{N \times d}$. The d denotes the dimension of representation. We employ a contrastive loss (Velickovic et al. 2019; Chen et al. 2020) to optimize the encoder $g(\cdot)$, pushing it to construct the invariant representations. Once the encoder $g(\cdot)$ is well trained, we can finally obtain the node representations $Z = g(G, X)$.

Causal Interpretation

Before analyzing graph contrastive learning with a causal view, we first elaborate on the structural causal model (SCM) to describe the graph generation process based on the following assumptions: (1) The original graph G can be decomposed into the set of causal variables C and the set of non-causal variables S. (2) Only C causally influences both the input graph G and downstream tasks, and S does not provide any information about the downstream tasks. (3) There is no causal relationship between S and C., i.e., the generation process of C is independent of S. Based on these assumptions, the SCM for the node classification task can be depicted in Figure 1, where Y represents the node labels. The dashed circle in SCM means the unobserved latent variable and the solid circle represents the observed variable. Please note that the causal variables in C may also be statistically dependent, so when we estimate the causal effect of a causal variable (marked by gray nodes), other causal variables are considered confounders (marked by red nodes). More details between variables are described as follows:

- $C \to G \leftarrow S$. The observed node data is generated by two unobserved latent variables: causal variables C and non-causal variables S.
- $C \longrightarrow Y$. This link means the causal part C contains all the necessary information for the node classification task.
- $G \dashleftarrow\dashrightarrow Y$. This dashed link indicates statistical dependence between G and Y. Since the raw graph contains two kinds of information, we will construct the statistical dependence relationship between G and Y if we do not distinguish the latent variables C and S.
- Common causes between variables are called confounders. In the SCM, the confounder provides a backdoor path to the causal variables, contributing to the correlation between different causal variables.

Now we analyze GCL based on the above SCM. In GCL, the augmented graphs V_A will contain the causal variables C_A and non-causal variables S_A, also, V_B will contain C_B and S_B. Ideally, only when $C_A = C_B = C$ and $S_A \neq S_B$, GCL is able to capture the invariant information C well.

However, let us take the widely used graph augmentation strategy (i.e., random augmentation (Zhu et al. 2020)) as an example. Because the random augmentation does not distinguish C and S, it is very hard to guarantee that $C_A = C_B = C$ and $S_A \neq S_B$ still hold after graph augmentation. As a result, the learned final representation Z will contain both C and S, and the predicted label Y' will be made as follows:

$$h(Z) = h(C, S) = Y', \qquad (3)$$

where h denotes a classification head. Apparently, because the representation contains non-causal information, the prediction might change when the non-causal part shifts, i.e.:

$$h(C, S) \neq h(C, S'). \qquad (4)$$

It is unreasonable because the two representations contain the same causal information, yet their predictions differ.

Therefore, we should perturb the non-causal factors S while keeping the causal factors C unchanged when generating augmentations, which can be regarded as conducting intervention on S, and meanwhile, ensure the consistency of causal information between different augmentations. Thus, the model should satisfy the following equation:

$$P^{do(S=s_i)}(Y|C) = P^{do(S=s_j)}(Y|C), \qquad (5)$$

where $do\,(S = s)$ represents the intervention on non-causal factors S. This formulation encourages the model to extract information only contained in causal factors C and discard trivial information.

Moreover, as shown in Figure 1, the causal variables contained in the C might be influenced by confounders (i.e., $c_1 \leftarrow Confounder \to c_2$, where c_1, c_2 denote two causal variables, respectively). The confounders lead to the correlation between variables, indicating that different causal variables will encode similar information. To obtain a more informative representation, it is essential to eliminate the influence of the confounders, which refers to the mutually independent between causal variables, i.e.,

$$\forall c_i, c_j \in C, c_i \perp\!\!\!\perp c_j, \qquad (6)$$

where c_i and c_j denote two different causal variables. This formulation implies that different variables in C are mutually independent, effectively mitigating the influence of confounders.

4 The Proposed Model: GCIL

In this section, we illustrate our proposed graph contrastive learning algorithm inspired by causality. The overall framework is shown in Figure 2. We initially generate two views through spectral graph augmentation and random augmentation, which can be seen as causal interventions. The two views are subsequently fed into a shared GNN encoder to generate node representations. Finally, we propose two dimension-level contrastive objectives. The invariance objective pushes the encoder to capture the invariant information contained in causal variables, while the independence objective encourages the different dimensions of the representations to be independent. Thus, we can obtain the invariant representations from the well-trained model.
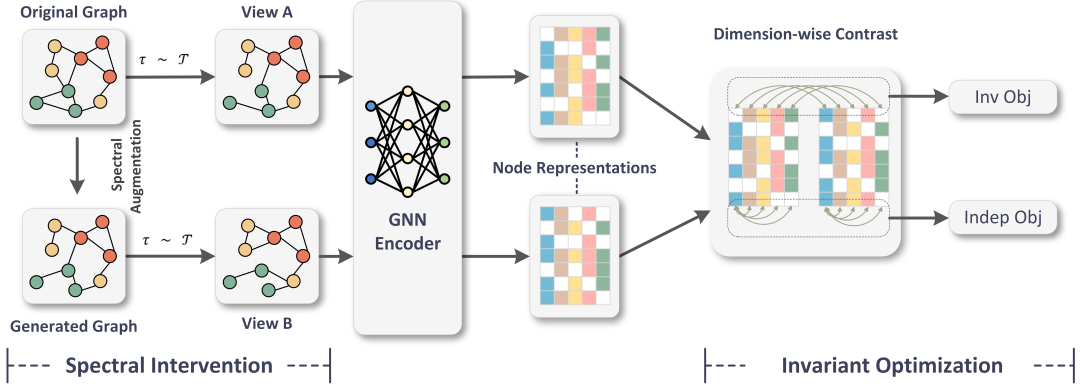
Figure 2: Overview of the GCIL framework. Given an original graph G, we first generate two views by spectral and random augmentation. The two views are subsequently fed into a shared GNN encoder to generate representations. At last, we optimize the invariance objective and the independence objective to render the model to learn the invariant representations.

Causal Intervention

Based on the above analysis, we need to generate graph augmentations that meet the following conditions: perturbing the non-causal information and maintaining causal information. However, it is very challenging that graph G does not provide any prior knowledge about C and S. Following (Liu et al. 2022a), perturbing the structure of the initial graph will change the strength of frequencies in the graph spectrum. The general graph augmentation should satisfy that: in two contrasted augmentations, the difference in high-frequency amplitudes should be greater than in low-frequency amplitudes. In other words, the lowest-frequency information can be approximately regarded as an invariant pattern between the two views.

We consider the low-frequency information in the graph as causal content and the high-frequency information as non-causal content. Therefore, we conduct the intervention on S by disrupting the high-frequency information while leaving the low-frequency information unchanged. In particular, given an original adjacency matrix A, our goal is to obtain an adjacency matrix A' by intervention, i.e., $A' = A + \Delta_{A+} - \Delta_{A-}$, where Δ_{A+} and Δ_{A-} indicate which edge is added and deleted, respectively. Δ_{A+} is obtained by maximizing the following objective function:

$$J = \langle \Theta L, \Delta_{A+} \rangle^2 + \epsilon H(\Delta_{A+}) + \langle f, \Delta_{A+} \mathbf{1}_n - a \rangle + \langle g, \Delta_{A+}^\top \mathbf{1}_n - b \rangle, \quad (7)$$

where Θ is a parameter updated in training and L is the laplacian matrix of G. ϵ is the weight parameter, f and g are Lagrange multipliers, and a, b are the node degree distributions. $H(\cdot)$ represents the entropy regularization, i.e., $H(P) = -\sum_{i,j} P_{i,j}(\log(P_{i,j}) - 1)$. The calculation of Δ_{A-} can be referred to (Liu et al. 2022a). We then corrupt two graphs with random data augmentation as follows:

$$V_A = \tau_A(A, X), V_B = \tau_B(A', X), \tau_A, \tau_B \in T, \quad (8)$$

where T represents the whole augmentation function space.

Invariance Objective

Now, based on the analysis in Section 3, we need to require that the relationship between V_A and V_B in Eq. 8 satisfies Eq. 5, so that the model can learn invariant representation. However, GCL does not provide any information about labels Y. To achieve the goal in Eq. 5, the equation can be reformulated as follows:

$$CE(C, S = s_i) = CE(C, S = s_j), \quad (9)$$

where CE represents the causal effect of variables. That is, we need to capture the consistency between the node representations $Z^A, Z^B \in R^{N \times d}$ obtained by Eq. 2. Then, we assume that each dimension in the representation holds the Gaussian distribution, and we propose an invariance objective that encourages the representations to remain unchanged dimension-wise, i.e., by enforcing the consistency of the mean and standard deviations for each dimension of the representations. Formally, the learning objectives can be formulated as follows:

$$\min_g \sum_i \left\| Z_i^A - Z_i^B \right\|_2^2, \quad s.t. Std(Z_i^A) = Std(Z_i^B) = \lambda, \quad (10)$$

where Z_i^A, Z_i^B denote the i-th dimension of two embedding matrices, respectively, and Std represents the standard deviation. The first term encourages the mean of two embedding matrices to be equal in the same dimension, and the second term pushes the standard deviation close to λ, where λ is the hyper-parameter.

Independence Objective

According to the SCM, different causal variables might be correlated due to the confounders, leading to less informative representations generated by the model. To mitigate this issue, we propose an independence objective to satisfy Eq. 6. The independence objective seeks to force the different causal variables to be mutually independent, thereby eliminating the correlation between causal variables. Specifically, we use the Hilbert-Schmidt Independence Criterion (HSIC) to measure the independence between factors. A value of 0

for HSIC indicates that the two variables are independent. Minimizing the following equation different dimensions in the representation Z to be independent:

$$\sum_{i \neq j} HSIC(Z_i, Z_j) = \sum_{i \neq j} \frac{1}{(N-1)^2} \operatorname{Tr}(\boldsymbol{K}_i \boldsymbol{H} \boldsymbol{K}_j \boldsymbol{H}), \quad (11)$$

where Z_i is the i-th dimension of embedding matrix, $\boldsymbol{H}$ represents the centering matrix $\boldsymbol{I} - \boldsymbol{1}\boldsymbol{1}^T \frac{1}{N}$, $\boldsymbol{K}_i$ and $\boldsymbol{K}_j$ are the two kernel matrices of Z_i, Z_j respectively.

Here, we introduce the notion of a kernel matrix $\boldsymbol{K_i} \in R^{N \times N}$, which calculates the kernel function value between each sample of the i-th variable. For example, $K_i^{a,b} = \kappa(Z_{i,a}, Z_{i,b})$, where κ denotes the kernel function, $K_i^{a,b}$ represents the number in the a-th row and b-th column of the matrix K_i, and $Z_{i,a}$ denotes the value in the i-th column and a-th row in the representation matrix Z. Using a complex kernel (e.g., Gaussian kernel) in HSIC to measure independence between dimensions can result in high space complexity, making it challenging to implement in scenarios with large sample sizes and dimensions. Inspired by lemma 1 in (Mialon, Balestriero, and LeCun 2022), minimizing the HSIC of different dimensions can be considered as minimizing the sum of the off-diagonal elements of the covariance matrix. The proof for this assertion is as follows:

Let kernel function $\kappa(Z_{i,a}, Z_{i,b}) = g(Z_{i,a})g(Z_{i,b})^T$, where $g : R \to R^L$ is an elementwise projector. We denote the mapping of such projectors on Z as $Q = g(Z) = [g(Z_1), \ldots, g(Z_d)] \in R^{N \times DL}$. According to the lemma 1 in Mialon, Balestriero, and LeCun (2022), we have:

$$
\begin{aligned}
&HSIC(Z_i, Z_j) \\
&= \frac{1}{(N-1)^2} \operatorname{Tr}(g(Z_i)g(Z_i)^T \boldsymbol{H} g(Z_j)g(Z_j)^T \boldsymbol{H}) \\
&= \frac{1}{(N-1)^2} \left\| g(Z_i)^T \boldsymbol{H} g(Z_j) \right\|_F^2 \qquad (12) \\
&= \left\| \operatorname{Cov}(g(Z_i), g(Z_j)) \right\|_F^2 \\
&= \left\| \operatorname{Cov}(\boldsymbol{Q})_{(i-1)L:iL, (j-1)L:jL} \right\|_F^2 .
\end{aligned}
$$

where Tr represents the trace of matrix, Cov represents the covariance of two variables. We let $g(X) = X$, i.e. linear kernel, in this case $Z = Q$, thus we have

$$\sum_{i \neq j} HSIC(Z_i, Z_j) = \sum_{i \neq j} \operatorname{Cov}(\mathrm{Q})_{i,j}^2 = \sum_{i \neq j} \operatorname{Cov}(\mathrm{Z})_{i,j}^2. \quad (13)$$

The calculation of HSIC values for different dimensions is converted to the calculation of covariance. Minimizing the eq. 13 ensuring independence between different dimensions.

Optimization Objective

We further normalize the embedding matrix dimension-wisely and $\tilde{\mathbf{Z}}$ represents the node embeddings after normalization. Note that $\left\| \tilde{Z}_i \right\|^2 = 1$, thus $\min_g \sum_i \left\| Z_i^A - Z_i^B \right\|_2^2$ can be replaced by maximizing the inner product of $\tilde{Z}_i^A$ and $\tilde{Z}_i^B$, e.g. $\max_g \sum_i \tilde{Z}_i^A \cdot \tilde{Z}_i^B$, where $\cdot$ represents inner product. The s_i denotes the standard deviation of the i-th dimension before

Dataset	#Nodes	#Edges	#Classes	#Features
Cora	2,708	10,556	7	1,433
Citeseer	3,327	9,228	6	3,703
Pubmed	19,717	88,651	3	500
Wiki-CS	11,701	432,246	10	300
Flickr	7,575	479,476	9	12,047

Table 1: Statistics of benchmark datasets

normalization. Minimizing $\sqrt{\|s_i - \lambda\|_2^2}$ pushes the standard deviation close to λ.

The overall optimization objective of our proposed GCIL is summarized as follows:

$$
\begin{aligned}
L = &-\alpha \sum_i \tilde{Z}_i^A \cdot \tilde{Z}_i^B + \beta \sum_i \{ \sqrt{\|s_i^A - \lambda\|_2^2} + \sqrt{\|s_i^B - \lambda\|_2^2} \} \\
&+ \gamma \sum_{i \neq j} \operatorname{Cov}(\tilde{\mathrm{Z}}^A)_{i,j}^2 + \operatorname{Cov}(\tilde{\mathrm{Z}}^B)_{i,j}^2,
\end{aligned}
\tag{14}
$$

where α, β, and γ are hyper-parameters controlling the importance of each term in the loss. The λ represents the desired standard deviation of the dimensions.

5 Experiements

Experimental Setup

Dataset. To evaluate our method, we consider five commonly used node classification benchmark datasets from the previous works (Velickovic et al. 2019; Mernyei and Cangea 2020; Liu et al. 2022a), including Cora, Citeseer, Pubmed, Wiki-CS, and Flickr. The statistic of these datasets is summarized in Table 1. In the above datasets, three benchmark citation datasets contain sparse one-hot features, while Wiki-CS has dense numerical features. We adopt the public splits for Cora, Citeseer, Pubmed, and Flickr, where the training set contains 20 nodes per class, 500 nodes for validation, and 1000 for testing. For the Wiki-CS dataset, we evaluate the models on the public splits provided in (Mernyei and Cangea 2020). More details of datasets are presented in Appendix A.

Baseline. We choose two kinds of methods as benchmarks: semi-supervised methods and self-supervised methods. **(1) Semi-supervised mehtods**: GCN (Kipf and Welling 2017) and GAT (Velickovic et al. 2018). **(2) Self-supervised methods**: DGI (Velickovic et al. 2019), MVGRL (Hassani and Khasahmadi 2020), GRACE (Zhu et al. 2020), GCA (Zhu et al. 2021), GraphCL (You et al. 2020), COSTA (Zhang et al. 2022), CCA-SSG (Zhang et al. 2021), ProGCL(Xia et al. 2022).

Parameter Settings. We set the dimensions of all datasets to 512, with a learning rate of 0.0002 for the Flickr dataset and 0.001 for all other datasets. The weight decay for all datasets is 0.0001. Additional details of parameter settings are presented in Appendix C.

Dataset	Cora		Citeseer		PubMed		Wiki-CS		Flickr	
Metrics	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1
GCN	80.6±0.7	81.5±0.6	68.1±0.5	70.9±0.5	78.5±0.3	78.9±0.3	73.2±0.8	77.5±0.4	48.9±1.6	50.2±1.2
GAT	81.3±0.3	82.3±0.2	67.5±0.2	72.0±0.9	77.4±0.2	77.8±0.2	75.5±0.4	78.3±0.4	35.0±0.8	37.1±0.3
DGI	80.4±0.7	82.0±0.5	67.7±0.9	71.7±0.8	76.8±0.9	76.7±0.9	70.6±0.1	75.6±0.1	31.2±1.6	33.0±1.6
MVGRL	81.5±0.5	82.8±0.4	66.8±0.7	72.5±0.5	79.8±0.4	79.7±0.3	74.9±0.1	78.1±0.1	31.2±2.9	33.4±3.0
GRACE	79.2±1.0	80.0±1.0	65.1±1.2	68.7±1.1	80.0±0.7	79.9±0.7	74.8±0.2	78.2±0.1	35.7±1.3	37.3±1.0
GCA	79.9±1.1	81.1±1.0	62.8±1.3	65.9±1.0	80.8±0.6	81.4±0.6	74.9±0.0	78.3±0.0	41.2±0.5	42.2±0.6
GraphCL	80.7±0.9	82.3±0.9	67.8±1.0	71.9±0.9	77.0±0.4	76.8±0.5	-	-	32.1±1.1	34.5±0.9
CCA-SSG	82.9±0.8	83.6±0.9	67.9±1.0	73.1±0.7	80.7±0.6	81.0±0.6	73.7±0.2	78.2±0.1	37.0±1.1	39.3±0.9
COSTA	81.2±0.4	82.4±0.7	62.9±3.6	66.4±4.3	79.8±0.6	80.4±0.9	72.2±5.1	76.2±2.3	**42.0±2.6**	42.8±2.5
ProGCL	81.0±0.4	81.9±0.6	62.9±0.9	65.9±0.9	80.5±2.3	80.5±2.3	75.3±0.1	**78.7±0.1**	37.8±1.1	38.5±1.0
Ours	**83.8±0.5**	**84.4±0.7**	**69.1±0.4**	**73.7±0.5**	**81.5±0.5**	**81.6±0.7**	**75.6±0.1**	78.6±0.1	40.0±0.8	**43.0±0.4**

Table 2: Quantitative results on node classification, where both the mean accuracy and the standard deviation are shown in the table. The best performance of self-supervised is bolded in table. The '-' indicates Out-of-Memory on a 24GB GPU.

Ablation	Cora		Wiki-CS	
	Ma-F1	Mi-F1	Ma-F1	Mi-F1
w/o indep	57.5	57.8	34.3	51.4
w/o inv	79.2	80.5	57.5	63.6
w/o Aug	83.3	84.1	75.1	78.0
GCIL	**83.8**	**84.4**	**75.6**	**78.6**

Table 3: Ablation Results on Cora and Wiki-CS

Node Classification

In this section, we evaluate the proposed GCIL on node classification. We conducted experiments on five datasets and the results are shown in Table 2. The best results of all self-supervised learning methods are bolded in the table. Please note that GraphCL has an out-of-memory issue on the Wiki-CS dataset. As we can see, our method GCIL achieves excellent performance on all five datasets compared to self-supervised methods. Except for the Flickr dataset, our method performs even better than semi-supervised learning methods. Specifically, we achieve the SOTA performance on Cora, Citeseer, and Pubmed datasets. In addition, we achieve the best result in Macro-F1 for the Wiki-CS dataset and only lower than the ProGCL method on Micro-F1. In the Flickr dataset, we can see that the semi-supervised method GCN achieves the best performance in two metrics. Our method outperforms all self-supervised methods in the Micro-F1 metric.

We empirically find that a two-layer GCN encoder yields the best accuracy on the Cora, Pubmed, Wiki-CS, and Flickr datasets, while a one-layer GCN encoder achieves the best performance on the Citeseer dataset. We note that some GCL methods (Zhu et al. 2020; Zhang et al. 2022) also utilize two layers of GCN on the Citeseer dataset. This indicates that our approach achieves strong performance with fewer computations compared to the aforementioned methods on certain datasets.

Ablation Studies

In this section, we investigate the impact of Causal Intervention (Aug.), Invariance Objective (Inv.) and Independence Objective (Indep.) in GCIL. We design three variants of GCIL. (1) **GCIL_w/o_Inv**: GCIL without Invariance Objective (inv.) module, i.e., hyper-parameter α and β are set to zero, and γ corresponds with GCIL. (2) **GCIL_w/o_Indep**: GCIL without Independence Objective (indep.) module, i.e., hyper-parameter γ is set to zero while α and β corresponds with GCIL. (3) **GCIL_w/o_Aug**: The spectral-based augmentation is removed while maintaining the optimization objective.

The results of three variants on four datasets are reported in Table 3. From the results, we can find that the overall performance order is as follows: GCIL > GCIL_w/o_Aug > GCIL_w/o_Indep > GCIL_w/o_Inv. The invariance objective has the greatest impact on the performance, which illustrates capturing the consistency between the node representations greatly encourages the model to encode task-related information. The performance of GCIL_w/o_Indep is worse than GCIL, which suggests that promoting independence between dimensions encourages the model to learn more informative node representations. Additionally, the result of GCIL_w/o_Aug indicates that our augmentation encourages the model to learn the invariant representations better than random augmentation. Finally, GCIL obtained the best performance, which indicates the effectiveness of considering three components together.

Hyper-parameter Sensitivity

In this section, we investigate the effect of α, β, γ, and embedding dimension. The results on four hyper-parameters are respectively reported in Figure 3.

Analysis of α. The α controls the importance of the term that guarantees a constant mean of the same dimension in two representations. We fix the hyper-parameters β and γ while varying α from 0.5 to 1.4 and report the F1-Macro and F1-Micro performance of GCIL. As shown in Figure 3(a) and 3(b), the performance of GCIL first increases and then
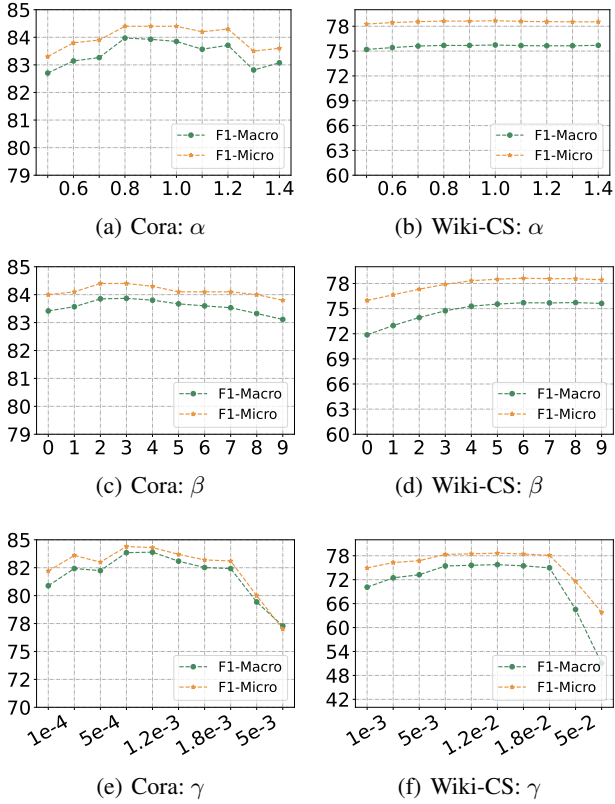
(a) Cora: α (b) Wiki-CS: α

(c) Cora: β (d) Wiki-CS: β

(e) Cora: γ (f) Wiki-CS: γ

Figure 3: The hyper-parameter sensitivity of GCIL with varying α, β and γ on Cora and Wiki-CS datasets.

decreases with the growth of α, while the performance fluctuation is very slight. It indicates that GCIL is not sensitive to α.

Analysis of β. The β controls the importance of the term that guarantees constant standard deviation of the same dimension in two representations. Similar to the above process, we fix the hyper-parameters α and γ while varying β from 0 to 9. In Figure 3(c) and 3(d), we can see that with a suitable range, the performance of our method benefits from the increase of β, and the performance will drop slightly with the large value of β.

Analysis of γ. The parameter γ controls the contribution of the independence objective. We varied γ from 1e-4 to 1e-2 for Cora and from 1e-3 to 0.1 for Wiki-CS to test its impact. Increasing attention on the independent term improved performance, demonstrating the objective's effectiveness in encoding relevant information. However, excessively large values of γ sharply decreased performance, indicating poor optimization of other loss terms. We can choose the appropriate γ by observing the degree of optimization of the invariance objective.

Visualization

To better understand the correlated relationships between the dimensions of the representation, we visualize the correlation matrix of the representation matrix Z for the Wiki-CS

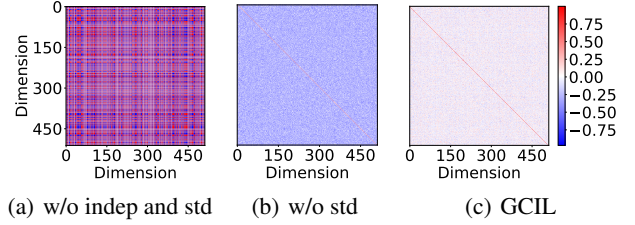(a) w/o indep and std   (b) w/o std   (c) GCIL

Figure 4: Correlation matrix of the representations on Wiki-CS.

dataset. In Figure 4, each row and column corresponds to a representation dimension, and the color indicates the Pearson correlation coefficient (Cohen et al. 2009). In Figure 4(a), we set the hyper-parameter β and γ to 0, while keeping the other hyper-parameters unchanged. We observe that strong correlations exist among different dimensions of the representation. This suggests that these dimensions may encode similar information. In Figure 4(b), we set the hyper-parameter β to 0 and discover that the correlations between dimensions are reduced compared to the previous case. However, different dimensions still correlate with each other. In contrast, the result of GCIL, as shown in Figure 4(c), the correlation matrix of the representation demonstrates that almost all off-diagonal values converge to 0. This indicates that different dimensions contain orthogonal and distinct information, which highlights the effectiveness of our method in learning informative representations. Overall, these visualizations confirm that our proposed method successfully captures and utilizes orthogonal information in the representation, leading to improved performance in capturing meaningful features.

6   Conclusion

In this paper, we study graph contrastive learning from the causal perspective and find out that previous methods may discard the causal information contained in the original graph, which prevents the model from learning the invariant representations. To learn the invariant representations, we propose a novel GCL method from the causal view. We first simulate conducting intervention on non-causal factors with spectral graph augmentation. Then, we design the invariance objective and the independence objective to encourage the model to extract causal information contained in the graph. Experimental results demonstrate that our proposed GCIL obtains the best performance across baselines on four node classification datasets.

References

Arjovsky, M.; Bottou, L.; Gulrajani, I.; and Lopez-Paz, D. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

Bruna, J.; Zaremba, W.; Szlam, A.; and Lecun, Y. 2014. Spectral networks and locally connected networks on graphs. In *ICLR*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *ICML*, 1597–1607.

Chen, Y.; Zhang, Y.; Bian, Y.; Yang, H.; Kaili, M.; Xie, B.; Liu, T.; Han, B.; and Cheng, J. 2022. Learning causally invariant representations for out-of-distribution generalization on graphs. *Advances in Neural Information Processing Systems*, 35: 22131–22148.

Cohen, I.; Huang, Y.; Chen, J.; Benesty, J.; Benesty, J.; Chen, J.; Huang, Y.; and Cohen, I. 2009. Pearson correlation coefficient. *Noise reduction in speech processing*, 1–4.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *NeurIPS*.

Fan, S.; Wang, X.; Mo, Y.; Shi, C.; and Tang, J. 2022a. Debiasing Graph Neural Networks via Learning Disentangled Causal Substructure. In *NeurIPS*.

Fan, S.; Wang, X.; Shi, C.; Cui, P.; and Wang, B. 2023a. Generalizing graph neural networks on out-of-distribution graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Fan, S.; Wang, X.; Shi, C.; Kuang, K.; Liu, N.; and Wang, B. 2022b. Debiased graph neural networks with agnostic label selection bias. *IEEE transactions on neural networks and learning systems*.

Fan, S.; Zhang, S.; Wang, X.; and Shi, C. 2023b. Directed acyclic graph structure learning from dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 7512–7521.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *NeurIPS*.

Hassani, K.; and Khasahmadi, A. H. 2020. Contrastive multi-view representation learning on graphs. In *ICML*, 4116–4126.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 9729–9738.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

Li, S.; Wang, X.; Zhang, A.; Wu, Y.; He, X.; and Chua, T.-S. 2022. Let invariant rationale discovery inspire graph contrastive learning. In *ICML*, 13052–13065.

Liu, N.; Wang, X.; Bo, D.; Shi, C.; and Pei, J. 2022a. Revisiting Graph Contrastive Learning from the Perspective of Graph Spectrum. In *NeurIPS*.

Liu, Y.; Jin, M.; Pan, S.; Zhou, C.; Zheng, Y.; Xia, F.; and Yu, P. 2022b. Graph self-supervised learning: A survey. *IEEE Transactions on Knowledge and Data Engineering*.

Mernyei, P.; and Cangea, C. 2020. Wiki-cs: A wikipedia-based benchmark for graph neural networks. *arXiv preprint arXiv:2007.02901*.

Mialon, G.; Balestriero, R.; and LeCun, Y. 2022. Variance covariance regularization enforces pairwise independence in self-supervised representations. *arXiv preprint arXiv:2209.14905*.

Mitrovic, J.; McWilliams, B.; Walker, J. C.; Buesing, L. H.; and Blundell, C. 2020. Representation Learning via Invariant Causal Mechanisms. In *ICLR*.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Pearl, J. 2009. *Causality*. Cambridge university press.

Pearl, J.; Glymour, M.; and Jewell, N. P. 2016. *Causal inference in statistics: A primer*. John Wiley & Sons.

Qiu, J.; Chen, Q.; Dong, Y.; Zhang, J.; Yang, H.; Ding, M.; Wang, K.; and Tang, J. 2020. Gcc: Graph contrastive coding for graph neural network pre-training. In *KDD*, 1150–1160.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *ICLR*.

Velickovic, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2019. Deep Graph Infomax. *ICLR*.

Wu, Y.; Wang, X.; Zhang, A.; He, X.; and Chua, T. 2022. Discovering Invariant Rationales for Graph Neural Networks. In *ICLR*.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Philip, S. Y. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 4–24.

Xia, J.; Wu, L.; Wang, G.; Chen, J.; and Li, S. Z. 2022. Progcl: Rethinking hard negative mining in graph contrastive learning. In *ICML*, 24332–24346. PMLR.

Xie, Y.; Xu, Z.; Zhang, J.; Wang, Z.; and Ji, S. 2022. Self-supervised learning of graph neural networks: A unified review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020. Graph contrastive learning with augmentations. *NeurIPS*, 5812–5823.

Zhang, D.; Zhang, H.; Tang, J.; Hua, X.-S.; and Sun, Q. 2020. Causal intervention for weakly-supervised semantic segmentation. *NeurIPS*, 655–666.

Zhang, H.; Wu, Q.; Yan, J.; Wipf, D.; and Yu, P. S. 2021. From canonical correlation analysis to self-supervised graph neural networks. *NeurIPS*, 76–89.

Zhang, Y.; Zhu, H.; Song, Z.; Koniusz, P.; and King, I. 2022. COSTA: Covariance-Preserving Feature Augmentation for Graph Contrastive Learning. In *KDD*, 2524–2534.

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

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2020. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*.

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2021. Graph contrastive learning with adaptive augmentation. In *TheWebConf*, 2069–2080.