Graph Contrastive Learning with Group Whitening

Chunhui Zhang zch16081114@163.com
North China Institute of Computing Technology, Beijing, China

Rui Miao ruimiao20@mails.jlu.edu.cn
School of Artificial Intelligence, Jilin University, Changchun, China

Editors: Berrin Yanıkoğlu and Wray Buntine

Abstract

Graph neural networks (GNNs) have demonstrated their great power in learning graph-structured data. Due to the limitations of expensive labeled data, contrastive learning has been applied in graph domain. We propose GWGCL, a graph contrastive learning method based on feature group whitening to achieve two key properties of contrastive learning: alignment and uniformity. GWGCL achieves the alignment by ensuring consistency between positive samples. There is no need for negative samples to participate, but rather to achieve the uniformity between samples through whitening. Because whitening has the effect of feature divergence, it avoids the collapse of all sample representations to a single point, which is called dimensional collapse. Moreover, GWGCL can achieve better results and higher efficiency without the need for asymmetric networks, projection layers, stopping gradients and complex loss function. Through extensive experiments, GWGCL performs competitively on node classification and graph classification tasks across ten common graph datasets. The code is in: https://github.com/MR9812/GWGCL.

Keywords: Graph neural networks, Contrastive learning, Graph representation learning

1. Introduction

Graph-structured data has emerged in a broad spectrum of research fields and application domains, such as recommender systems (He et al., 2020), biological networks (Diao et al., 2022), social networks (Perozzi et al., 2014) and knowledge graph (Wang et al., 2018). The inherent complexity(i.e., non-Euclidean space) of graph-structured data that differs from text and image has imposed unprecedented challenges to machine learning.

Recently, graph neural networks(GNNs) have exhibited prominent performance in learning representations and various downstream tasks including node classification(Kipf and Welling, 2017; Veličković et al., 2018), link prediction (Grover and Leskovec, 2016) and graph classification (Zhou et al., 2021). Although GNNs have achieved promising performance on classification tasks, most existing GNN models ignore the rich information in a large number of unlabeled data. To fully exploit the information of unlabeled data, contrastive learning as an effective self-supervised technique has been widely developed in graph data mining domain.

Contrastive learning has achieved significant success in various fields (Chen et al., 2020b; Gao et al., 2021). The contrastive learning encourages the learned representations of positive samples to be similar while pushing apart the representations of randomly sampled negative samples (Chen et al., 2020b). Researchers have proposed that contrastive learning should

satisfy the following two properties: alignment, where two samples from a positive pair should be mapped to adjacent regions in the representation space; and uniformity, where all samples should be distributed roughly evenly in the representation space (Wang and Isola, 2020). The current contrastive learning methods commonly use InfoNCE (Oord et al., 2018) loss, such as SimCLR (Chen et al., 2020b). For a center sample y, the contrastive loss function can be presented as

$$L_{cl} = -\log \frac{e^{\frac{f(x)^T f(y)}{\tau}}}{e^{\frac{f(x)^T f(y)}{\tau}} + \sum_i e^{\frac{f\left(x_i^-\right)^T f(y)}{\tau}}} \tag{1}$$

Where τ is a scalar temperature hyperparameter, $f(\cdot)$ is the encoder, $\left(\left\{x_i^-\right\}_{i=1}^M, y\right) \sim p_{\text{data}}, (x, y) \sim p_{\text{pos}}$ and negative samples are discovered by randomly sampling from the dataset. Therefore, the InfoNCE loss function brings the positive samples closer together to achieve the alignment, and satisfies the uniformity by pushing apart the negative samples, ensuring that the representations are evenly distributed. It is worth noting that the success of the InfoNCE loss is strongly influenced by the quantity and quality of negative samples. However, using a large number of negative samples incurs higher time and space costs. If only alignment is considered, there will be a phenomenon called dimension collapse (Hua et al., 2021), where different dimensions capture the same information.

Therefore, researchers have sought a more efficient approach to simultaneously satisfy the alignment and uniformity. Recently, a new research direction has emerged, where contrastive learning is performed using only positive samples. BYOL (Grill et al., 2020) utilizes an online network to update a target network by a moving average to avoid dimension collapse. Barlow Twins (Zbontar et al., 2021) avoids dimension collapse by measuring the cross-correlation matrix between two augmentation views, and making it as close to the identity matrix as possible.

In the field of graph representation learning, some methods (Zhu et al., 2020b; You et al., 2020) are similar to SimCLR, using negative samples to achieve the uniformity. Furthermore, there have been research efforts on contrastive learning methods that only use positive samples. BGRL (Thakoor et al., 2021) ensures the uniformity of all samples by utilizing an asymmetric network structure, which is achieved by adding a prediction layer and applying stop-gradient, and updating the target network by exponential moving average from the online network. CCA-SSG (Zhang et al., 2021) employs an additional loss function that enforces the cross-correlation matrix between two augmentation views to be close to the identity matrix.

Another study Ermolov et al. (2021) proposed that batch normalization(BN) (Ioffe and Szegedy, 2015) is the crucial component to avoid degenerate solutions. Based on this perspective, we introduce the normalization technique commonly used in supervised learning into the context of contrastive learning without negative samples. BN is the first method that normalizes each batch of data in a way that supports back propagation and has shown significant performance improvements in training deep neural networks. After applying the BN transformation, the feature distribution has a mean of 0 and a variance of 1, which partially alleviates dimension collapse. However, using BN alone is not sufficient, as features exhibit certain correlations, indicating the presence of redundant information. Huang et al.

(2018) proposed Decorrelated Batch Normalization (DBN), which not just centers and scales activations but whitens them. To prevent dimension collapse, it is necessary to decorrelate the feature dimensions during the contrastive learning process. Unlike CCA-SSG, which constrains the cross-correlation matrix through a loss function, we adopt group whitening to decorrelate the learned representation matrix. This approach endows the whitened representations with the following property: the elimination of correlations between dimensions and a variance of 1 for all representations. Our contributions are as follows:

- We propose GWGCL, a simple yet effective graph contrastive learning framework with group whitening, which does not rely on negative samples to achieve the uniformity.
- We conduct extensive experiments on benchmark datasets and various tasks, which can demonstrate the effectiveness of our proposed GWGCL comparing with state-of-the-art baselines.

2. Related Works

2.1. Graph neural networks

Recently, graph neural networks (GNNs) have received growing attention which utilize node features as well as the unique graph structure information to learn node representations. Existing GNNs follow the neighborhood aggregation strategy, which we iteratively update the node representation by aggregating the representations of its neighboring nodes and combining with its representations (Xu et al., 2018). Numerous variants of GNNs (Kipf and Welling, 2017; Wu et al., 2019; Veličković et al., 2018) have been proposed to achieve outstanding performances in a wide variety of graph-based tasks (Zhou et al., 2021). However, in most graph application tasks, most current GNNs ignore the rich information in a large number of unlabeled data. Therefore, we utilize the recently developed contrastive learning technique to fully explore the rich information of the unlabeled data.

2.2. Contrastive learning

Recently, contrastive learning technique has arisen a lot of research interest due to its novel ideas. The key idea of contrastive learning is to learn the representations by contrasting positive and negative samples in a self-supervised manner. The success of contrastive learning has aroused repercussions in the field of Computer Vision (Chen et al., 2020b; Grill et al., 2020; Zbontar et al., 2021) and Natural Language Processing (Gao et al., 2021). Contrastive learning on graph domain has proven to be an active and promising research area with broad potential applications (Zhu et al., 2020b; You et al., 2020; Miao et al., 2022). DGI (Veličković et al., 2019) is the first proposed graph contrastive learning method utilizing the idea of defining the mutual information between nodes and the graph representation as the contrastive metric. MVGRL (Hassani and Khasahmadi, 2020) utilizes node diffusion technique to obtain the augmented graph and explicitly models the relationship between node representation and graph summary using discriminator. GRACE (Zhu et al., 2020b) and GCA (Zhu et al., 2020a) learn discriminative representations by maximizing the agreement of nodes between different augmented views. BGRL (Thakoor et al., 2021) utilizes the asymmetric network structure, which achieved by adding a prediction layer and

applying stop-gradient, and updating the target network using the exponential moving average from the online network. CCA-SSG (Zhang et al., 2021) employs an additional loss function that enforces the cross-correlation matrix of two views' representations close to an identity matrix.

3. Approach

3.1. Model Framework

In this paper, an efficient graph contrastive learning method called GWGCL is developed, which addresses the issue of dimension collapse by applying feature whitening to remove redundancy between feature dimensions, without the use of negative samples. GWGCL consists of four main components: 1) a graph augmentation generator, 2) a encoder based on GNN (Graph Neural Network), 3) the group whitening technology, and 4) a loss function that constraints consistency among positive samples. The framework of GWGCL is illustrated in Figure 1 and the pseudo-code is in Appendix C.

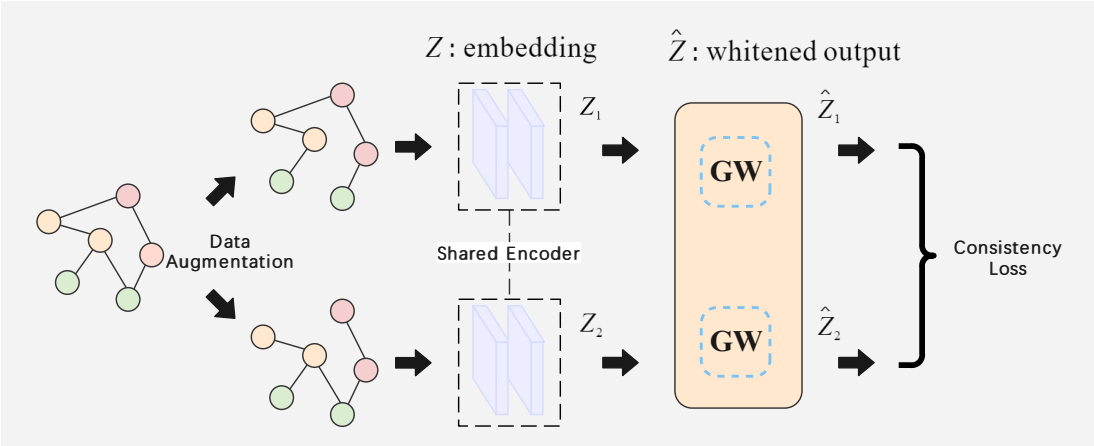


Figure 1: GWGCL first builds two augmentation views through a graph augmentation generator, then uses a shared encoder to generate node representations and performs group whitening (GW), and finally utilizes the consistency loss function to constrain the similarity between positive samples.

For a single graph $G = (A, X)$, $X \in \mathbb{R}^{N*F}$and $A \in \mathbb{R}^{N*N}$ respectively represent the feature matrix and adjacency matrix, where N is the number of nodes in the graph, and F represents the feature dimension. GWGCL first generates two views $G_1 = (A_1, X_1)$ and $G_2 = (A_2, X_2)$ of G by applying graph augmentation, and then generates node representations Z_1 and Z_2 from the augmentation graph using a GNN encoder. Among them, $Z_1, Z_2 \in \mathbb{R}^{N*D}$ and D represent the dimensions of the learned representations, and then whiten the learned node representations to achieve uniformity. Finally, the consistency loss function is used to constrain positive samples to express similarity and ensure alignment. Next, we will introduce graph data augmentation, encoder, group whitening strategy and consistency loss function in detail.

3.2. Data Augmentation

Contrastive learning learns representations by maximizing the consistency between different augmentation views of the same data. Data augmentation technology is very important in contrastive learning. We use simple graph data augmentation techniques to ensure fair comparison with existing graph contrastive learning methods. We introduce two standard methods of random graph data augmentation commonly used in previous work: edge deletion(ED) and feature mask(FM). Edge deletion refers to randomly deleting some edges from the original graph, while feature Mask refers to randomly masking some features of all nodes, setting the feature value to 0. Given an edge deletion rate β_e and a feature mask rate β_f, the graph augmentation process is described as follows:

$$\widehat{G} = ED\left(FM\left(G, \beta_f\right), \beta_e\right). \tag{2}$$

Given a graph G, two augmentation graphs $\widehat{G}_1$ and $\widehat{G}_2$ are obtained by performing edge deletion and feature mask operations on the original graph G.

3.3. Encoder

GNNs have emerged as one of the standard tools to learn graph-structured data. Mathematically, based on the unique message passing mechanism, through the k-th layer of GNNs, the learned representation vector $z_i^{(k)}$ of each node v_i is obtained by:

$$z_i^{(k)} = \text{Aggregate}(\{z_j^{(k-1)} \mid \forall j \in \mathcal{N}(i) \cup i\}; \theta), \tag{3}$$

where $\mathcal{N}(i)$ denotes the neighbor set of node i and aggregate function denotes the combination operator (e.g., sum, mean, or max) on the neighborhood representations.

3.4. Group Whitening

We perform ZCA whitening (Bell and Sejnowski, 1997) operation on the learned node representations $H = \{h_1, \ldots, h_B\} \in \mathbb{R}^{B*D}$ which is a batch of D-dimensional vector. The output obtained after ZCA whitening is vector $Z = \{z_1, \ldots, z_B\} \in \mathbb{R}^{B*D}$. The specific calculation method is as follows:

$$Z = Q\Lambda^{-1/2}Q^T\widehat{H}, \tag{4}$$

where $\widehat{H}$ is the zero-mean matrix of H, $\widehat{H}_{d,b} = H_{d,b} - \frac{1}{B}\sum_{k=1}^{B} H_{d,k}$, $\Lambda \in \mathbb{R}^{D*D}$ is the diagonal matrix of eigenvalues, $Q \in \mathbb{R}^{D*D}$ is the eigenvector, satisfying $\widehat{H}\widehat{H}^T = Q\Lambda Q^T$. ZCA whitening assumes that $\widehat{H}\widehat{H}^T \in \mathbb{R}^{D*D}$ is a full rank matrix. The output matrix Z of ZCA is a zero-mean matrix, therefore we can compute the covariance matrix by the following formula:

$$\begin{aligned} ZZ^T &= Q\Lambda^{-1/2}Q^T\widehat{H}\widehat{H}^TQ\Lambda^{-1/2}Q^T \\ &= Q\Lambda^{-1/2}Q^{\mathrm{T}}\mathrm{Q}\Lambda Q^{\mathrm{T}}\mathrm{Q}\Lambda^{-1/2}Q^{\mathrm{T}} \\ &= Q\Lambda^{-1/2}\Lambda\Lambda^{-1/2}Q^{\mathrm{T}} \\ &= QQ^{\mathrm{T}} = \mathrm{I}. \end{aligned} \tag{5}$$

It can be found that the output matrix Z obtained after ZCA whitening satisfies the property of dimensional decorrelation. In addition, in order to improve efficiency, we utilize the group whitening strategy. The grouping strategy can not only improve flexibility, but also improve computational efficiency.

$\widehat{H} \in \mathbb{R}^{B*D}$ must have at least D ranks, and since the row-mean of matrix $\widehat{H}$ is 0, its rank is constrained by $B-1$, indicating that the size of batch B is at least $D+1$. This greatly limits the flexibility of ZCA whitening, as either the number of dimensions in the representation space must be limited or the batch size must be adjusted. For group whitening, the batch size only needs to be scaled based on the group size G.

Another advantage of group whitening is that it improves efficiency. Without grouping, the computational cost of a single ZCA whitening is $O\left(BD^2\right)$, where B is the size of batch and D is the dimension of representations. Compared to the non group whitening method, the group whitening operation with the group size of G only requires $O\left(BDG\right)$ computational cost, which has significant advantages.

3.5. Consistency Loss

After obtaining the representations of augmented views, we use the Mean Squared Error(MSE) loss to constrain the similarity between positive samples. The MSE loss is obtained by calculating the squared difference between the predicted values and the ground truth. It can also measure the similarity between two vectors, where a smaller value indicates a closer relationship. The mathematical form of l_{MSE} is as follows:

$$l_{MSE}\left(z_1, z_2\right) = ||z_1 - z_2||_2^2 \,, \tag{6}$$

where z_1 and z_2 represent the embeddings obtained by encoding a pair of positive samples.

4. Experiments

In this section, extensive experiments will be conducted to evaluate the effectiveness of our proposed GWGCL by answering the following questions:

- **Q1**: Does GWGCL demonstrate superior performance in semi-supervised node classification tasks compared to current graph contrastive learning methods?
- **Q2**: Is our GWGCL better than the current graph contrastive learning methods in graph classification tasks?
- **Q3**: How do different BN layers and related hyperparameters affect the performance of GWGCL?

4.1. Node Classification

Datasets and Baselines. We conduct experiments on the commonly used citation networks (Cora, CiteSeer and PubMed) and Amazon co-purchase networks (Amazon Computer and Amazon Photo). For citation networks, following the widely used standard split proposed by Kipf and Welling (2017), we use 20 labeled nodes per class for training, 500 nodes for validation and 1000 nodes for testing. For Amazon co-purchase networks, we follow the

experimental settings of BGRL (Thakoor et al., 2021) and CCA-SSG (Zhang et al., 2021), and randomly divides these two datasets into 0.1/0.1/0.8. The statistics of datasets are presented in Table 1. GWGCL was compared with classical supervised models GCN (Kipf and Welling, 2017), GAT (Veličković et al., 2018), and the current graph contrastive learning methods, such as DGI (Veličković et al., 2019), MVGRL (Hassani and Khasahmadi, 2020), GRACE (Zhu et al., 2020b), BGRL (Thakoor et al., 2021), CCA-SSG (Zhang et al., 2021), GGD (Zheng et al., 2022), MA-GCL (Gong et al., 2023), and GREET (Liu et al., 2023).

Table 1: Statistics of datasets used in node classification task.

Datasets	Nodes	Edges	Features	Classes
Cora	2,708	5,429	1,433	7
CiteSeer	3,327	4,732	3,703	6
PubMed	19,717	44,338	500	3
Computer	13,752	245,861	767	10
Photo	7,650	119,081	745	8

Experiments Settings. GWGCL follows the widely used graph contrastive learning evaluation standard, which is proposed by DGI (Veličković et al., 2019): 1) In representation learning stage, by optimizing the objective function in GWGCL, the GCN encoder is trained in an unsupervised manner; 2) In classification evaluation stage, the parameters of the GCN encoder are fixed, the original graph is fed to the encoder to get the node representations, and finally the node representations are input into a linear classifier and trained to generate predictive labels for the nodes. In the second stage, only the nodes in the training set are used to train the Linear classifier, and then the best classifier parameters are selected by the performance of validation set. This paper implemented the representation learning stage and linear classification stage of this model using PyTorch and Adam optimizers (Kingma and Ba, 2014). All experiments were conducted on an NVIDIA TitanRTX GPU with 24GB of memory. A standard two-layer GCN (Kipf and Welling, 2017) model was used as the encoder in all datasets except CiteSeer, as experiments have shown that a single layer GCN has better performance in CiteSeer. We tune group size from $\{16, 32\}$, embedding dimension from $\{512, 1024\}$ and augmentation rate from $\{0.1, 0.2, \ldots, 0.7\}$.

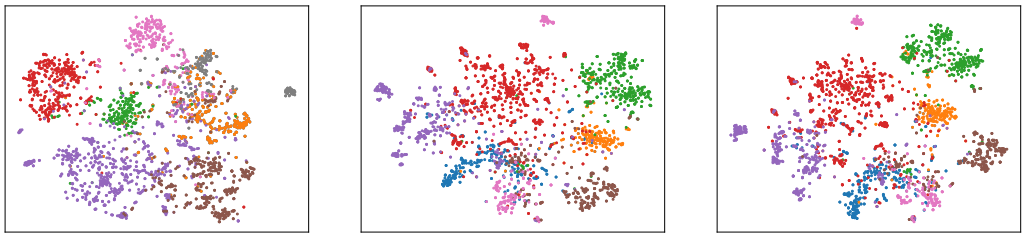
Results Analysis. The experimental results are shown in Table 2. We can make the following observations to answer research question **Q1**.

❶ *Our proposed GWGCL exhibits significantly superior performance compared with the current graph contrastive learning methods.* We can find that BGRL, CCA-SSG, and GWGCL without using negative samples have achieved good performance, indicating the feasibility of not using negative samples to improve efficiency. In addition, CCA-SSG and GWGCL exhibit better classification performance compared to BGRL, indicating that feature decorrelation can satisfy the uniformity in contrastive learning. Although the architecture of the proposed GWGCL is simple, it outperforms current graph contrastive learning methods on used datasets other than computer. And the experimental results show that the classification performance of GWGCL is better than that of BGRL with asymmetric structure and CCA-SSG with decorrelation loss function, which shows the effectiveness of

ZCA group whitening in realizing feature decorrelation. In addition, the implementation of ZCA whitening is much simpler than BGRL's asymmetric structure and CCA-SSG's decorrelation loss function.

Table 2: Test Accuracy (%) for different methods on five datasets. In addition, we show the best and runner-up results are highlighted with bold and underline, respectively. We run 10 times and report the mean ± standard deviation.

Methods	Cora	CiteSeer	Pubmed	Computer	Photo
GCN	81.7 ± 0.7	71.4 ± 0.5	79.1 ± 0.3	86.5 ± 0.5	92.4 ± 0.2
GAT	83.0 ± 0.7	72.5 ± 0.7	79.0 ± 0.3	86.9 ± 0.3	92.6 ± 0.4
DGI	82.3 ± 0.6	71.8 ± 0.7	76.8 ± 0.6	84.0 ± 0.5	91.6 ± 0.2
MVGRL	83.5 ± 0.4	73.3 ± 0.5	80.1 ± 0.7	87.5 ± 0.1	91.7 ± 0.1
GRACE	81.9 ± 0.4	71.2 ± 0.5	80.6 ± 0.4	86.3 ± 0.3	92.2 ± 0.2
CCA-SSG	84.2 ± 0.4	73.1 ± 0.3	81.6 ± 0.4	88.7 ± 0.3	93.1 ± 0.1
BGRL	82.7 ± 0.6	71.1 ± 0.8	79.6 ± 0.5	$\mathbf{90.3 \pm 0.2}$	93.2 ± 0.3
GGD	$\underline{83.9 \pm 0.4}$	73.0 ± 0.6	81.3 ± 0.8	$\underline{90.1 \pm 0.9}$	92.5 ± 0.6
MA-GCL	83.3 ± 0.4	$\underline{73.6 \pm 0.1}$	$\mathbf{83.5 \pm 0.4}$	88.8 ± 0.3	$\mathbf{93.8 \pm 0.1}$
GREET	83.8 ± 0.9	73.1 ± 0.8	80.3 ± 1.0	87.9 ± 0.4	92.9 ± 0.3
GWGCL	$\mathbf{84.5 \pm 0.8}$	$\mathbf{74.3 \pm 0.6}$	$\underline{81.9 \pm 0.5}$	88.9 ± 0.4	$\underline{93.3 \pm 0.2}$



(a) GRACE (b) CCA-SSG (c) GWGCL

Figure 2: The t-SNE embeddings of nodes in the Cora dataset.

❷ *The GWGCL learns to concentrate node embeddings of the same class within compact space.* To better understand the effect of GWGCL, we utilize t-SNE (Van der Maaten and Hinton, 2008) to visualize the node representations learned by GRACE, CCA-SSG and GWGCL in Figure 2. Our analysis focuses exclusively on the Cora dataset due to its minimal number of nodes. We can find that our GWGCL can exhibit 2D projections with more coherent shapes of clusters. Therefore, GWGCL can obtain representations that are beneficial for classification tasks.

❸ *Our GWGCL effectively relieves the problem of over-smoothing.* Over-smoothing which suggests that as the number of layers increases, the representations of the nodes in GCN are inclined to converge to a certain value and thus become indistinguishable (Chen

et al., 2020a; Gasteiger et al., 2018). We analyse the impact of model depth (number of layers) on node classification performance in Figure 3. When increasing the model depth, GWGCL performs consistently better than GCN, GRACE and CCA-SSG at each layer. This is because whitening operation constrains the divergence between nodes and decorrelates the dimensional information.

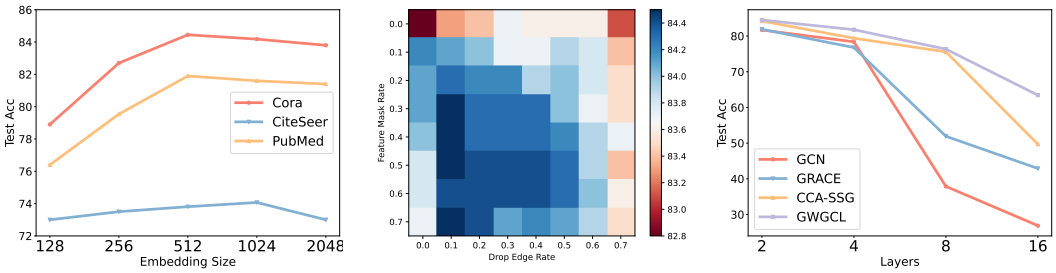


Figure 3: Left: The effect of embedding dimension on citation networks. Middle: The effect of different augmentation rates on Cora. Right: The test accuracies of GCN, GRACE, CCA-SSG and GWGCL on Cora with different layers.

4.2. Graph Classification

Datasets and Baselines. For graph classification task, we conduct experiments on TU-Dataset (Morris et al., 2020), including PROTEINS, DD, MUTAG, COLLAB and REDDIT-B. The statistics of datasets are presented in Table 3. In order to verify the effectiveness of GWGCL, we compare classical graph kernel methods, including WL (Shervashidze et al., 2011) and DGK (Yanardag and Vishwanathan, 2015), and also compare with current graph unsupervised and contrastive methods, such as graph2vec (Narayanan et al., 2017), MVGRL (Hassani and Khasahmadi, 2020), Infograph (Sun et al., 2019), GraphCL (You et al., 2020), JOAO (You et al., 2021) and SimGRACE (Xia et al., 2022).

Table 3: Statistics of datasets used in graph classification task.

Datasets	Graphs	Avg. Node	Avg. degree	Classes
PROTEINS	1,113	39.06	1.86	2
DD	1,178	284.32	715.66	2
MUTAG	188	17.93	19.79	2
COLLAB	5,000	74.49	32.99	3
RDT-B	2,000	429.63	1.15	2

Experiments Settings. For the evaluation protocol, following Sun et al. (2019), after generating graph embeddings with GIN (Xu et al., 2018) encoder and readout function, we feed graph-level representations into a SVM classifier to predict the label of graph, and report the mean 10-fold cross-validation accuracy with standard deviation after 5 runs.

Table 4: Test Accuracy (%) on five graph classification datasets.

Methods	PROTEINS	DD	MUTAG	COLLAB	RDT-B
WL	72.92 ± 0.56	–	80.72 ± 3.00	–	68.82 ± 0.41
DGK	73.30 ± 0.82	–	87.44 ± 2.72	–	78.04 ± 0.39
graph2vec	73.30 ± 2.05	–	83.15 ± 9.25	–	75.78 ± 1.03
MVGRL	–	–	75.40 ± 7.80	–	82.00 ± 1.10
InfoGraph	74.44 ± 0.31	72.85 ± 1.78	89.01 ± 1.13	70.65 ± 1.13	82.50 ± 1.42
GraphCL	74.39 ± 0.45	78.62 ± 0.40	86.60 ± 1.34	71.36 ± 1.15	89.53 ± 0.84
JOAO	74.55 ± 0.41	77.32 ± 0.54	87.35 ± 1.02	69.50 ± 0.36	85.29 ± 1.35
JOAOv2	74.07 ± 1.10	77.40 ± 1.15	87.67 ± 0.79	69.33 ± 0.34	86.42 ± 1.45
SimGRACE	$\mathbf{75.35 \pm 0.09}$	77.44 ± 1.11	89.01 ± 1.31	71.72 ± 0.82	89.51 ± 0.89
GWGCL	74.89 ± 0.54	$\mathbf{78.83 \pm 0.33}$	$\mathbf{89.68 \pm 2.27}$	$\mathbf{72.82 \pm 1.06}$	$\mathbf{90.31 \pm 0.37}$

Table 5: Summary of classification results for different representation methods (percentage accuracy).

Methods	Cora	CiteSeer	MUTAG	RDT-B
None	57.3	46.1	86.7	86.8
BN	79.4	70.9	88.1	87.3
Whitening	83.2	73.5	88.6	89.9
Group Whitening	**84.5**	**74.3**	**89.7**	**90.3**

Result Analysis. The experimental results are shown in Table 4. We show the best results in bold. We can make the following observations to answer research question **Q2**.

❹ *Our GWGCL exhibits superior performance compared with the current graph contrastive learning methods on graph classification tasks.* We report results from previous papers if available. We find that GWGCL outperforms all self-supervised baselines on four out of five datasets and has competitive results on PROTEINS. The experimental results show that our GWGCL has a wide range of applications.

4.3. Ablation Study and Efficiency Analysis

Effect of Group Whitening. ❺ *Group whitening is critical in our GWGCL framework.* In order to further validate the benefits of feature whitening, four experimental settings, namely standard BN layer, ZCA whitening, ZCA grouping whitening, and no processing, were evaluated on Cora, CiteSeer, MUTAG and RDT-B. The results were reported in Table 5. It can be observed that compared to not performing any operations, the addition of BN and whitening significantly improved the performance of both datasets, demonstrating the importance of feature divergence. And the effect of feature whitening is better than BN, which reflects the advantage of feature decorrelation. Finally, the effect of grouping whitening is stronger than feature whitening, indicating that grouping operation can improve the flexibility of feature whitening.

Table 6: Efficiency Analysis of GWGCL.

Methods	Cora		PubMed	
	Time(s)	Memory(MB)	Time(s)	Memory(MB)
GRACE	0.02	1355	0.34	12327
BGRL	0.02	1589	0.15	2279
CCA-SSG	0.02	1324	0.09	2056
GWCCL	0.02	**1263**	**0.05**	**1814**

Effect of Embedding Dimension. ❻ *Unlike supervised models, such as GCN and GAT, GWGCL requires a larger representation dimension to show superior performance.* Figure 3 shows the impact of the representation dimension in the citation networks. Through experiments, it is found that the appropriate number of dimensions in the Cora and PubMed datasets is 512, and the optimal number of dimensions in CiteSeer is 1024, which is slightly larger than the current graph contrastive learning methods. Appendix B provides more information on the performance of GCN and GAT in different dimensions, which suggests that GCN and GAT do not benefit from the larger dimensions. Through the experimental results in the Figure 3, it can be found that the classification performance of GWGCL decreases when the number of dimensions is set too large or too small.

Effect of Data Augmentation. ❼ *The augmentation ratio also plays a crucial role in graph contrastive learning.* From the results in Figure 3, it can be found that compared to using only a single graph data augmentation method, combining feature mask and edge deletion can achieve better performance, indicating the necessity of mining difficult positive samples. At the same time, it can also be found that either too large or too small augmentation rates will result in poor classification performance. It is speculated that when the augmentation rate is too small, the generated positive samples are relatively close, and bringing the positive samples closer cannot be a challenging excuse task. However, when the augmentation rate is excessively increased, it will excessively change the structure and feature attributes of the graph, thereby damaging the key information of the nodes, causing two augmentation samples of the same node to be unable to serve as positive samples of each other can also lead to a significant decrease in classification performance.

Efficiency Analysis. ❽ *GWGCL is a simple and effective graph contrastive learning method.* We analyze the efficiency of GWGCL and observe whether it has significant advantages compared to the current graph contrastive learning methods. Table 6 compares the training time (training time per epoch) and memory overhead of GWGCL with other graph contrastive learning methods on the Cora and Pubmed datasets. Overall, GWGCL has shorter training time and less memory cost, and has significant efficiency advantages.

5. Conclusion and Discussions

In this paper, we first introduce the reason why the negative samples based contrastive learning methods are successful is that the InfoNCE loss function satisfies the alignment and uniformity, pulls positive samples closer to ensure the alignment, and pushes away the

negative samples to achieve the uniformity. In order to improve the efficiency of graph contrastive learning, the current method aims to achieve uniformity without using negative samples. In order to solve the problem of dimension collapse in contrastive learning, we proposes a graph contrastive learning method GWGCL based on feature whitening, which utilizes ZCA group whitening to make the learned representations diverge and ensure the uniformity. Then, consistency loss is used to constrain the consistency between positive samples to ensure the alignment. Finally, the effectiveness of GWGCL in graph representation learning was demonstrated through experiments.

References

Anthony J Bell and Terrence J Sejnowski. The "independent components" of natural scenes are edge filters. *Vision research*, 37(23):3327–3338, 1997.

Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *International Conference on Machine Learning*, pages 1725–1735. PMLR, 2020a.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020b.

Cameron Diao, Kaixiong Zhou, Xiao Huang, and Xia Hu. Molcpt: Molecule continuous prompt tuning to generalize molecular representation learning. *arXiv preprint arXiv:2212.10614*, 2022.

Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In *International Conference on Machine Learning*, pages 3015–3024. PMLR, 2021.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.

Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997*, 2018.

Xumeng Gong, Cheng Yang, and Chuan Shi. Ma-gcl: Model augmentation tricks for graph contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4284–4292, 2023.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Aditya Grover and Jure Leskovec. Node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery*

and Data Mining, KDD '16, page 855–864, New York, NY, USA, 2016. Association for Computing Machinery.

Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*, pages 4116–4126. PMLR, 2020.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648, 2020.

Tianyu Hua, Wenxiao Wang, Zihui Xue, Sucheng Ren, Yue Wang, and Hang Zhao. On feature decorrelation in self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9598–9608, 2021.

Lei Huang, Dawei Yang, Bo Lang, and Jia Deng. Decorrelated batch normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 791–800, 2018.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

Yixin Liu, Yizhen Zheng, Daokun Zhang, Vincent CS Lee, and Shirui Pan. Beyond smoothing: Unsupervised graph representation learning with edge heterophily discriminating. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 4516–4524, 2023.

Rui Miao, Yintao Yang, Yao Ma, Xin Juan, Haotian Xue, Jiliang Tang, Ying Wang, and Xin Wang. Negative samples selecting strategy for graph contrastive learning. *Information Sciences*, 613:667–681, 2022.

Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*, 2020.

Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui Chen, Yang Liu, and Shantanu Jaiswal. graph2vec: Learning distributed representations of graphs. *arXiv preprint arXiv:1707.05005*, 2017.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(9), 2011.

Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. *arXiv preprint arXiv:1908.01000*, 2019.

Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Mehdi Azabou, Eva L Dyer, Remi Munos, Petar Veličković, and Michal Valko. Large-scale representation learning on graphs via bootstrapping. *arXiv preprint arXiv:2102.06514*, 2021.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018.

Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep Graph Infomax. In *International Conference on Learning Representations*, 2019.

Ruijie Wang, Yuchen Yan, Jialu Wang, Yuting Jia, Ye Zhang, Weinan Zhang, and Xinbing Wang. Acekg: A large-scale knowledge graph for academic data mining. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 1487–1490, 2018.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. PMLR, 2019.

Jun Xia, Lirong Wu, Jintao Chen, Bozhen Hu, and Stan Z Li. Simgrace: A simple framework for graph contrastive learning without data augmentation. In *Proceedings of the ACM Web Conference 2022*, pages 1070–1079, 2022.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Pinar Yanardag and SVN Vishwanathan. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1365–1374, 2015.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, 33:5812–5823, 2020.

Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In *International Conference on Machine Learning*, pages 12121–12132. PMLR, 2021.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

Hengrui Zhang, Qitian Wu, Junchi Yan, David Wipf, and Philip S Yu. From canonical correlation analysis to self-supervised graph neural networks. *Advances in Neural Information Processing Systems*, 34:76–89, 2021.

Yizhen Zheng, Shirui Pan, Vincent Lee, Yu Zheng, and Philip S Yu. Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination. *Advances in Neural Information Processing Systems*, 35:10809–10820, 2022.

Kaixiong Zhou, Qingquan Song, Xiao Huang, Daochen Zha, Na Zou, and Xia Hu. Multi-channel graph neural networks. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 1352–1358, 2021.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. *arXiv preprint arXiv:2010.14945*, 2020a.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. *CoRR*, abs/2006.04131, 2020b.

Appendix A. Framework of Graph Classification

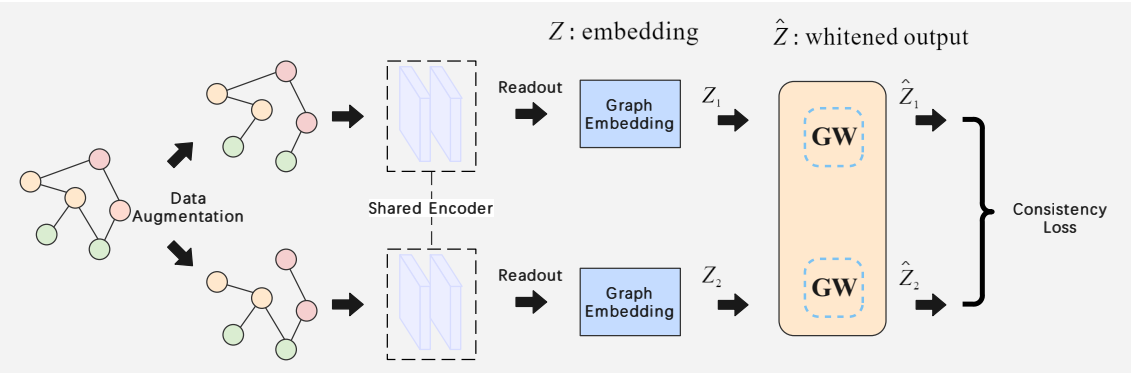


Table 7: Test Accuracy (%) for different embedding dimensions on five datasets.

Methods	Cora	CiteSeer	PubMed	Computer	Photo
GCN-64	81.7 ± 0.7	71.4 ± 0.5	79.1 ± 0.3	86.5 ± 0.5	92.4 ± 0.2
GCN-512	81.5 ± 0.5	71.7 ± 0.3	79.3 ± 0.4	86.2 ± 0.4	92.1 ± 0.2
GAT-8	83.0 ± 0.7	72.5 ± 0.7	79.0 ± 0.3	86.9 ± 0.3	92.6 ± 0.4
GAT-64	83.2 ± 0.6	72.1 ± 0.8	79.2 ± 0.4	87.2 ± 0.2	92.9 ± 0.3

Table 8: Test Accuracy (%) for different methods on Ogbn-Arxiv.

Methods	Ogbn-Arxiv
GCN	$\mathbf{71.74 \pm 0.29}$
GWGCL	71.23 ± 0.34

Appendix B. Supplementary Experiments

Appendix C. Algorithm

We provide the pseudo code for our GWGCL about node classification tasks.

Algorithm 1 Algorithm for GWGCL

Input: original graph G, encoder f_θ, number of iterations E, the edge deletion rate β_e and the feature mask rate β_f

Output: Node embeddings

for $1 \leq i \leq E$ **do**

$\widehat{G}_1 = ED\left(FM\left(G, \beta_f\right), \beta_e\right), \widehat{G}_2 = ED\left(FM\left(G, \beta_f\right), \beta_e\right)$;

/* Generate two augmentation graphs. */

$\mathbf{Z}_1 = f_\theta\left(\widehat{G}_1\right), \mathbf{Z}_2 = f_\theta\left(\widehat{G}_2\right)$;

/* Get node embeddings in augmentation graphs through the encoder */

Group whitening for node representations according to Eq 4;

Calculate the loss function $\mathcal{L}$ according to Eq 6;

$\bigtriangledown_\theta\left[\mathcal{L}\right]$;

end

Get node embeddings in the original graph, $\mathbf{Z} = f_\theta\left(G\right)$, where θ is the frozen parameters of the encoder.